CS512 – Mining user reviews to recommend popular dishes

Sonia Mohanlal, mohanla2
Namrata Prabhu, nprabhu2
Xiaocheng Yuan, xyuan12
Motivation

Problem

• Information overload, due to high volume of reviews
• Need to extract/recommend relevant items to serve information need
• No existing applications address this specific need

Idea

• Create application that automatically mines user reviews to recommend popular dishes
• Use data-mining and natural language processing techniques to extract and rank dishes by popularity
Related Work: Yelp

Keywords highlighted are not helpful.
Related Work: DishTip

Extract dishes but ranks them completely incorrectly

Does not contain any data for many restaurants
Problem Definition – Inputs & Outputs

Input: Corpus of user reviews

- Dish mentions in reviews
- Review Ratings
- Dish Sentiment Analysis

Ranked list of popular dishes
Experimental dataset

The Challenge Dataset:

- 2.2M reviews and 591K tips by 552K users for 77K businesses
- 566K business attributes, e.g., hours, parking availability, ambience.
- Social network of 552K users for a total of 3.5M social edges.
- Aggregated check-ins over time for each of the 77K businesses
- 200,000 pictures from the included businesses

Get the Data
Preprocessing data
- Collect and parse champaign-urbana reviews for restaurants with at least 15 reviews

Entity Extraction
- Obtain relevant entities from user reviews through TopMine

Entity Categorization
- Use NLP techniques to fetch entities related to food

Ranking
- Assign each dish relevance score based on various measures that include but are not limited to, user ratings, sentiment, document frequency etc.
## Entity Extraction

<table>
<thead>
<tr>
<th>Collocation with pointwise mutual information</th>
<th>TopMine</th>
<th>SegPhrase</th>
</tr>
</thead>
</table>
| • Adv: Obtains words that occur more frequently than expected.  
  • Disadv: Lower recall and precision compared to other techniques | • Adv: Fast unsupervised algorithm that generates phrases, has high recall.  
  • Disadv: Results contain noise | • Adv: Supervised algorithm that generates high quality phrases, has high precision.  
  • Disadv: Lower recall compared to TopMine |
Other techniques tried

• TopMine results post-processed with The New York Public Library’s menu data from 1940’s to the present. Menu data not exhaustive.

• ClusType with distant supervision. Results unreliable as no type information found for food in Freebase due to API deprecation. Other strategies yielded poorer results than TopMine.

  We ultimately picked TopMine since it has higher recall as our priority was to extract as many dishes as possible, since we could filter those out later and re-rank according to our own ranking function to compute its popularity score.
## Entity Extraction - TopMine

<table>
<thead>
<tr>
<th>Rank</th>
<th>Entity</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>beer battered bacon</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>cheese curds</td>
<td>22</td>
</tr>
<tr>
<td>3</td>
<td>pretty good</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>good food</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>food is good</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>Great food</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>waiting minutes</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>bison burger</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>downtown Champaign</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>deviled eggs</td>
<td>12</td>
</tr>
</tbody>
</table>

Initial results after running TopMine with `min_supp` of 4 and threshold 2. Results contain dishes but also has significant noise.
Three types of errors and fixes

- **highly recommend wait to go back**
  - Solution: Use POS tag to eliminate phrases which do not end in a noun or a noun-phrase.

- **downtown champaign Chambana**
  - Solution: Use existing NER tool to identify locations.

- **pretty good good food food is good**
  - Solution: Use document frequency to filter out common phrases.
List of keywords with post-processing

1. beer battered bacon  23
2. cheese curds       22
3. bison burger       13
4. deviled eggs       12
5. beer battered asparagus  12
6. beer selection     11
7. stout ice cream    11
8. chocolate cake     11
9. fried egg          11
10. fish and chips    10
Ranking - Features

- True Document frequency
- User Rating
- Sentiment Analysis
- Recency
True Document Frequency

• We view each piece of review text as a document for the business

• Here, we particularly look at true document frequency of each phrase, especially if one of the extracted phrases is a substring of another.

• For ex: suppose our list of keywords contains “sweet potato fries” and “sweet potato”, we calculate the true document frequency of “sweet potato” by counting only those occurrences where it is not followed by fries.
Rating & Sentiment Analysis

• Higher true document frequency of a dish usually indicates higher popularity

• But user ratings and the sentiment of the dish within the review is also important.

• For example, a review may have an overall negative sentiment but a particular dish may be mentioned positively.

\[
\text{RatingScore} : f = \begin{cases} 
0, & \text{if userRating} \leq 3 \\
1, & \text{otherwise.}
\end{cases} \quad \text{SentimentScore} : f = \begin{cases} 
1, & \text{if sentiment of dish is positive} \\
0, & \text{if sentiment of dish is negative}
\end{cases}
\]
Example of positive review and negative sentiment

Here the review rating has 4 stars, but the dish pizza (highlighted in bold) has a negative sentiment.
Example of negative review and positive sentiment

Here the review rating has 2 stars but the dish pizza (highlighted in bold) has positive sentiment.

In my opinion a good brewpub has great beer with good food to accompany the beer. Maybe it was the fact that we checked it out 4 days after it opened, but of the 5 beer styles we got, 4 of them were “off”. One was clearly oxidized (badly), one was infested to lacto and tasted very funny and the other two mildly oxidized. We hoped the manager would come visit with us, but no luck. The bartender indicated that depending on the time of day, the beer tastes different from the yeast. First time I have heard of that. Food was decent, but I have had better pub grub. Maybe the beer just ruined my taste buds.

(Please) The pizza is huge. Enough for two to share or one very hungry person.
Recency

• For recency, we want to penalize dish-mentions in older reviews since they may not be relevant in the present.

• The recency of document can be represented by the following formula (time difference is in months):

\[ \Delta t = t_{current} - t_{document} \]

• We model the recency as an exponential decay:

\[ e^{-\lambda (\Delta t)} \]

• Here, lambda is a parameter that the user can control, the higher the lambda the more dishes mentioned in older reviews are penalized and vice-versa.
Sample Results

<table>
<thead>
<tr>
<th>For Black Dog:</th>
<th>For Destihl</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Burnt ends</td>
<td>1 Beer-Battered Asparagus</td>
</tr>
<tr>
<td>2 Pulled pork</td>
<td>2 Beer-Battered Bacon</td>
</tr>
<tr>
<td>3 Sweet potato fries</td>
<td>3 Cheese curds</td>
</tr>
<tr>
<td>4 Georgia Peach</td>
<td>4 Stuffed Poblano Peppers</td>
</tr>
<tr>
<td>5 Beef Brisket</td>
<td>5 Bread Pudding</td>
</tr>
</tbody>
</table>
Future Directions

- Sub-categorize dishes. For ex: ranked list of drinks, appetizers etc.
- Experiment with more techniques and criteria to improve ranking and information retrieval, for ex: user query
- Obtain dish popularity trends with temporal mining
Questions?