Classifying Food and Beverage Clients

Gaurav Chawla, Jason Kuo, Nanshan Li and Andres Potapczynski
Neoway delivers customer insights based on firmographic data publicly available on the web

We know Consumer Goods

Today there are hundreds of thousands of independent food/beverages operators and retailers in the USA.

Using industry specific data, Neoway helps uncover this landscape, whether you are looking to discover new sales opportunities, optimize your go-to-market approach, or maximize sales with existing customers.

SEE HOW IT WORKS
They want to uncover certain food/beverages categories from a client’s webpages.

- **Restaurant**
  - Type of Cuisine (American, Mexican, Chinese, etc.)
- **Supermarket / Grocery**
  - Organic / Non-Organic
- **Bar**
  - Predominant type of alcohol served (beer, wine, cocktail)
- **Liquor Store**
  - Predominant type of alcohol sold (beer, wine, cocktail)
For this, they have gathered close to 1M HTMLs from ~200K clients

- Capturing the HTMLs for different clients
- Storing the HTMLs indexed by the initial URL in the cloud
- Adding metadata (for a subset)
- Running Ad-hoc methods

Gathered the HTML contents via a web crawling bot
Stored the HTMLs indexed by the initial URL in the cloud
Augmented the database with information from Google, Yelp and others
Labelling using keywords via RegEx
We designed a pipeline to help them improve the previous process:

1. **Tokenization**: Extracting useful information from the HTMLs.
2. **Anomaly Detection**: Removing non-informative HTMLs.
3. **Feature Extraction**: Creating useful features.
4. **Classification**: Train different classifiers for each task.
We designed a pipeline to help them improve the previous process.
Store the tokenized data in a single DB to accelerate the model prototyping and shareability

**Approach**
- Extracted HTML body via XML parser
- Employed NLTK package for lemmatizing, handling punctuation and capitalization, etc
- POS tagging removed sentence filler words
- Created a parallelizable implementation for speed

**Tokenized HTML**

- Windy, city, grille, home, gift, certificate, coupon, restaurant, overview, letter, history, menu, main, dish, side, challenge, photo, meal, social, medium, facebook, twitter, foursquare, contact, application, windy, city, grille, privacy, policy, gyro, menu, strip, cooked, lamb, beef, tomato, onion, cucumber, sauce, pita, fold, chicken, gyro, menu, grilled, chicken, tomato, onion, cucumber

**Recommendations / Next Steps**
- Generate a consolidated DB that contains all the tokens for each client
  - Use a NoSQL DB such as SQLite Dict or MongoDB
We designed a pipeline to help them improve the previous process.
Isolation Forests\(^1\) gives an anomaly score to each observation

**Approach**

- Construct a series of meta-features (length, # of unique tokens, etc.)
- Define weak thresholds on each feature
- Use Isolation Forests to determine the outliers

**Recommendations / Next Steps**

- Find the relationship of meta-feature via classification
- Understand the procedure’s impact on accuracy

---

We designed a pipeline to help them improve the previous process:

1. **Tokenization**
   - Extracting useful information from the HTMLs

2. **Anomaly Detection**
   - Removing non informative HTMLs

3. **Creating useful features**

4. **Classification**
   - Train different classifiers for each task

5. **Feature Extraction**
Use Topic Modeling and Doc2Vec to get a dense feature representation

**Feature construction process**

- **TF-IDF Filtering**
  - Reduces the vocabulary from 16K to ~ 2K
  - Decreases training complexity downstream

- **Topic Modeling**
  - Uncovers the general topics present in the data

- **Doc2Vec**
  - Generates a semantic dense representation

**75x training time reduction**
Understand the topics present in the data and create features that preserve its semantics

LDA\(^1\) and NMF\(^2\) helped uncover topics in the data

**LDA Topic Examples with top words**
- Topic 44: pizza, cheese, chicken, sauce, mozzarella, tomato, onion, Italian, garlic
- Topic 46: margarita, tacos, taco, Mexican, menu, location, specials, happy hour
- Topic 71: twitter, facebook, Instagram, google, email, skip, press, online, menu, location
- Topic 76: chicken, rice, shrimp, sauce, beef, pork, fried, vegetable, onion, spicy

**NMF Topic Examples with top words**
- Topic 1: cheese, bacon, onion, tomato, lettuce, salad, cheddar, chicken, choice, potato
- Topic 4: pizza, slice, topping, crust, pepperoni, cheese, large, pasta, order, phone
- Topic 6: Mexican, authentic, margarita, family, salsa, tacos, good, recipe, nachos
- Topic 83: good, great place, time, friendly staff, family, atmosphere, town, friend

**Recommendations / Next Steps**
- Augment client categorization based on the topics uncovered
- Use a smaller and dense set of features to save time
- Try state-of-the-art word embeddings like ELMo or Bert

---

Now, we designed a pipeline to help them improve the previous process:

1. **Tokenization**
   - Extracting useful information from the HTMLs

2. **Anomaly Detection**
   - Removing non informative HTMLs

3. **Creating useful features**
   - Feature Extraction

4. **Classification**
   - Train different classifiers for each task
Overall Results and models used for each classification task

**Establishment Type**
- Classification
- Multi-class
- Naive Bayes
- F1 score: 90%

**Restaurant Cuisine Classification**
- Multi-label Classification
- Binary Relevance with Logistic Regression (one-hot)
  - F1 score: 70%
  - Jaccard Similarity: 66%
  - MicroAvg - Pre: 86%
  - MicroAvg - Re: 60%

**Supermarket / Grocery Organic vs Not organic Classification**
- Binary Classification
- NN
  - (64ReLu-128ReLu-64ReLu-2Sigmoid
- Accuracy: 92%
- Precision: 43%
- Recall: 75%

**Bar Type of Alcohol Classification**
- Categorical Regression
- Ridge Regression (per category)
- RMSE Beer: 0.08
- RMSE Cocktail: 0.07
- RMSE Wine: 0.02

**Liquor Store Type of Alcohol Classification**
- Categorical Regression
- Ridge Regression (per category)
Overall Results and models used for each classification task

### Restaurant Cuisine Classification
- **Problem**
- **Model**
- **Performance**
  - Multi-label Classification
  - Binary Relevance with Logistic Regression (one-hot)
  - F1 score: 70%
  - Jaccard Similarity: 66%
  - MicroAvg - Pre: 86%
  - MicroAvg - Re: 60%

### Supermarket / Grocery Organic vs Not organic Classification
- **Problem**
- **Model**
- **Performance**
  - Binary Classification
  - NN
  - (64ReLu-128ReLu-64ReLu-2Sigmoid)
  - Accuracy: 92%
  - Precision: 43%
  - Recall: 75%

### Bar Type of Alcohol Classification
- **Problem**
- **Model**
- **Performance**
  - Categorical Regression
  - Ridge Regression (per category)
  - RMSE Beer: 0.08
  - RMSE Cocktail: 0.07
  - RMSE Wine: 0.02

### Liquor Store Type of Alcohol Classification
- **Problem**
- **Model**
- **Performance**
  - Categorical Regression
  - Ridge Regression (per category)
Cuisine classification carried considerable difficulties

Task: Multi-label Classification

- 99 non-mutually exclusive cuisines
- Implied hierarchy not present
- Multi-label output required
- Highly imbalanced data

Metrics

- Hamming loss
- Jaccard Similarity Score
- Micro-average Precision & Recall
- F1 Score

Three main approaches:

- Binary Relevance
  - Assumes independence between cuisines
- Classification Chains
  - Output is added to input of the next classifier: n! permutations
- Multi-label KNN
  - Lazy KNN method
Cuisine classification carried considerable difficulties

Results on 66 Regional Cuisines

- Binary Relevance approach:
  - Decision Tree
  - KNN
  - Logistic Regression
  - Multi-layer Perceptron
  - Random Forest

- Comparing results from different feature representations on Logistic Regression BR:
  - LDA
  - LDA + Doc2Vec
  - NMF
  - Doc2Vec
  - One hot encoding

Recommendations

- Clean up existing cuisine labels
- Develop potential hierarchy and orthogonal label framework:
  - Regional cuisines
  - Food
  - Restaurant format
  - Dietary restrictions
- Framework / spatial understanding of cuisines can be used to improve approach taken in this task
Overall Results and models used for each classification task

Restaurant Cuisine Classification
- Multi-label Classification
- Binary Relevance with Logistic Regression (one-hot)
  - F1 score: 70%
  - Jaccard Similarity: 66%
  - MicroAvg - Pre: 86%
  - MicroAvg - Re: 60%

Supermarket / Grocery Organic vs Not organic Classification
- Binary Classification
  - NN
  - (64ReLu-128ReLu-64ReLu-2Sigmoid
  - Accuracy: 92%
  - Precision: 43%
  - Recall: 75%

Bar Type of Alcohol Classification
- Categorical Regression
  - Ridge Regression (per category)
  - RMSE Beer: 0.08
  - RMSE Cocktail: 0.07
  - RMSE Wine: 0.02

Liquor Store Type of Alcohol Classification
- Categorical Regression
  - Ridge Regression (per category)

Establishment Type Classification
- Multi-class
- Naive Bayes
- F1 score: 90%

Problem
- Model
- Performance
The Ridge Regression coefficients correctly capture the relevant words for each class.

**Words & coefficients - Cocktail**

**Words & coefficients - Wine**

**Words & coefficients - Beer**

**Approach**
- Run a regression model for each type of drink
- Test different regularization schemes like Lasso, Ridge and Elastic Net

**Recommendations**
- Use Gridsearch CV to determine the amount of regularization
- Try non-linear models that output a class score (like NN with sigmoid)
Our models handle cases where the labels are incorrect

**Given**
- (Beer, Cocktail, Wine)
  - (0.6, 0.3, 0.1)

**Prediction**
- (Beer, Cocktail, Wine)
  - (0.3, 0.5, 0.1)

**Given**
- (Beer, Cocktail, Wine)
  - (0.2, 0.6, 0.2)

**Prediction**
- (Beer, Cocktail, Wine)
  - (0.5, 0.2, 0.2)
Our pipeline has addressed all the previous limitations but still has elements to improve

### Ad-hoc limitations
- Heavily dependent on domain knowledge
- Not adaptable to new labels
- RegEx language hard to debug and error-prone

### Pipeline results
- **Derives domain knowledge** based on labeled data
- **Scalable** to all types of problems
- **Robust** to mistakes and noise

### Next Steps
- Run ensemble algorithms like CatBoost
- Add **weightings** to deal with the minority class
- Create a **small test set** to validate performance while avoiding noisy labels
Q & A
Back-up
Isolation Forest assigns an anomaly score for each observation.

At each partition, it selects at random a feature and split.

Results of Isolation Forest in two clusters.
Use TF-IDF to construct a manageable vocabulary

As expected, the words exhibit a long-tailed distribution

TF-IDF Discussion

- Adjusting the document frequency bounds
  - Setting an upper bound removes uninformative common stop words such as food, order or menu
  - Setting a lower bound removes rare words such as foreign language terms that may overfit model

- Incorporate bigrams in the feature list
  - Captures additional semantics such as pizza sauce or cheese burger that may be lost just looking at single word counts

\[
w_{i,j} = tf_{i,j} \times \log \left( \frac{N}{df_i} \right)
\]

- \( tf_{i,j} \) = number of occurrences of \( i \) in \( j \)
- \( df_i \) = number of documents containing \( i \)
- \( N \) = total number of documents
LDA uncovers the topics present in the data

Example of LDA in a given corpus

LDA Graphical Model
NMF also finds topics in the data

Matrix Factorization

\[ \begin{align*}
W &\times H &\approx V \\
\end{align*} \]

NMF creates “additive” elements

\[ \begin{align*}
\text{NMF} &\times \text{Original} = \end{align*} \]
Doc2Vec generates a dense vector representation that preserves the semantics

**Doc2Vec Problem**

Classifier

Average/Concatenate

Paragraph Matrix

- Center Word
- Context Word

c=0  The cute cat jumps over the lazy dog.
c=1  The cute cat jumps over the lazy dog.
c=2  The cute cat jumps over the lazy dog.

**Word2Vec Country – Capital Example**

Country and Capital Vectors Projected by PCA
## Multi-label classification metrics

### Hamming Loss

\[
\frac{1}{|N| \cdot |L|} \sum_{i=1}^{N} \sum_{j=1}^{L} \text{XOR}(y_{i,j}, z_{i,j})
\]

where \( y_{i,j} \) is the target and \( z_{i,j} \) is the prediction.

### Jaccard Similarity Score

\[
J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}.
\]

### Micro-averaged Precision and Recall

- Microaveraging Precision \( Pr^{\text{micro}}(D) = \frac{\sum_{c \in C} TP_s(c)}{\sum_{c \in C} TP_s(c) + FP_s(c)} \)
- Microaveraging Recall \( Re^{\text{micro}}(D) = \frac{\sum_{c \in C} TP_s(c)}{\sum_{c \in C} TP_s(c) + FN_s(c)} \)

### F1 score

\[
F1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]
Before, Neoway relied on ad-hoc methods which carried some limitations:

- Heavily dependent on domain knowledge
- Not adaptable to new labels
- Difficult to handle various forms of the same lemma
- RegEx language hard to debug and error-prone
- Unwieldy supporting large vocabulary

### Defining a set of rules
- Determined the relevant keywords to look for:
  - Organic
  - Draft Beer
  - ...
  - Tortilla
- Construct a set of RegExs to find the keywords
  - /\w\._%+-]+\[/
- Run the search on the database

### Inspecting the HTML
```html
<html>
  <body>
    <h1>
      Welcome to Whole Foods
    </h1>
    ...
    <p>
      Find the best deals on organic apples
    </p>
  </body>
</html>
```