Yelp Restaurant Subcategory Prediction

Gabriel Appleby

September 2019

1 Proposal

Yelp is a popular website that allows users to review businesses. While some amount of categorical data is collected about the business, such as whether or not it is a restaurant, most of the data comes in the form of raw text. The yelp data set is quite large, so we will restrict ourselves to restaurants within the United States. We propose to take textual reviews left by users and train a deep neural network to predict the subcategory of the restaurant. Most restaurants already have a subcategory, such as Italian food, or American food.

We formalize the problem as a multi-class supervised learning problem, where we have some features and want to train a model to predict a known category. Let R be the set of all yelp restaurants in the United States that contain one or more review, C be the set of all yelp reviews, and A be the set of every restaurant's other attributes. Each restaurant $r \in R$, has reviews other / comments $\mathbf{c}_r \in C$, along with a vector of other attributes $\mathbf{a}_r \in A$. An individual review/comment c_{ru} is written for a specific restaurant by a specific user $u \in U$, where U is the set of all users.

When predicting the category of a restaurant r the input X_r will contain $|\mathbf{c}_r| + 1$ rows, and $|c_{ru}|$ columns. The extra row being where the attributes $\mathbf{a}_r$ will be placed. The length of $|c_{ru}|$ will depend upon the embedding, and the $|\mathbf{a}_r|$ will simply be the number of useful features available for all restaurants. The output will be a logit for each class / subcategory.

$$X_{cr} = \begin{bmatrix} \mathbf{a}_r \\ C_u \end{bmatrix} \tag{1}$$

The main sources of difficulty with this approach is that we want to treat each input review the same, and also want to allow for a variable number of reviews per user. We accomplish these goals by simply using the same weight vector for all review rows (all rows but attributes), and zero padding any inputs that have fewer than the largest number of reviews. A nice way to think of this approach is as a particularly weird convolutional neural network. There is only one filter at each layer that is the same size as a single row. It passes over each row just as a normal filter with stride 1. This approach is of course needlessly computationally expensive since we have to do a bunch of zero padding, but should yield interesting results.

We propose to take yelp restaurant review data and extract the restaurant subcategory from it. This will allow us to fill in any the subcategory for restaurants missing the tag.