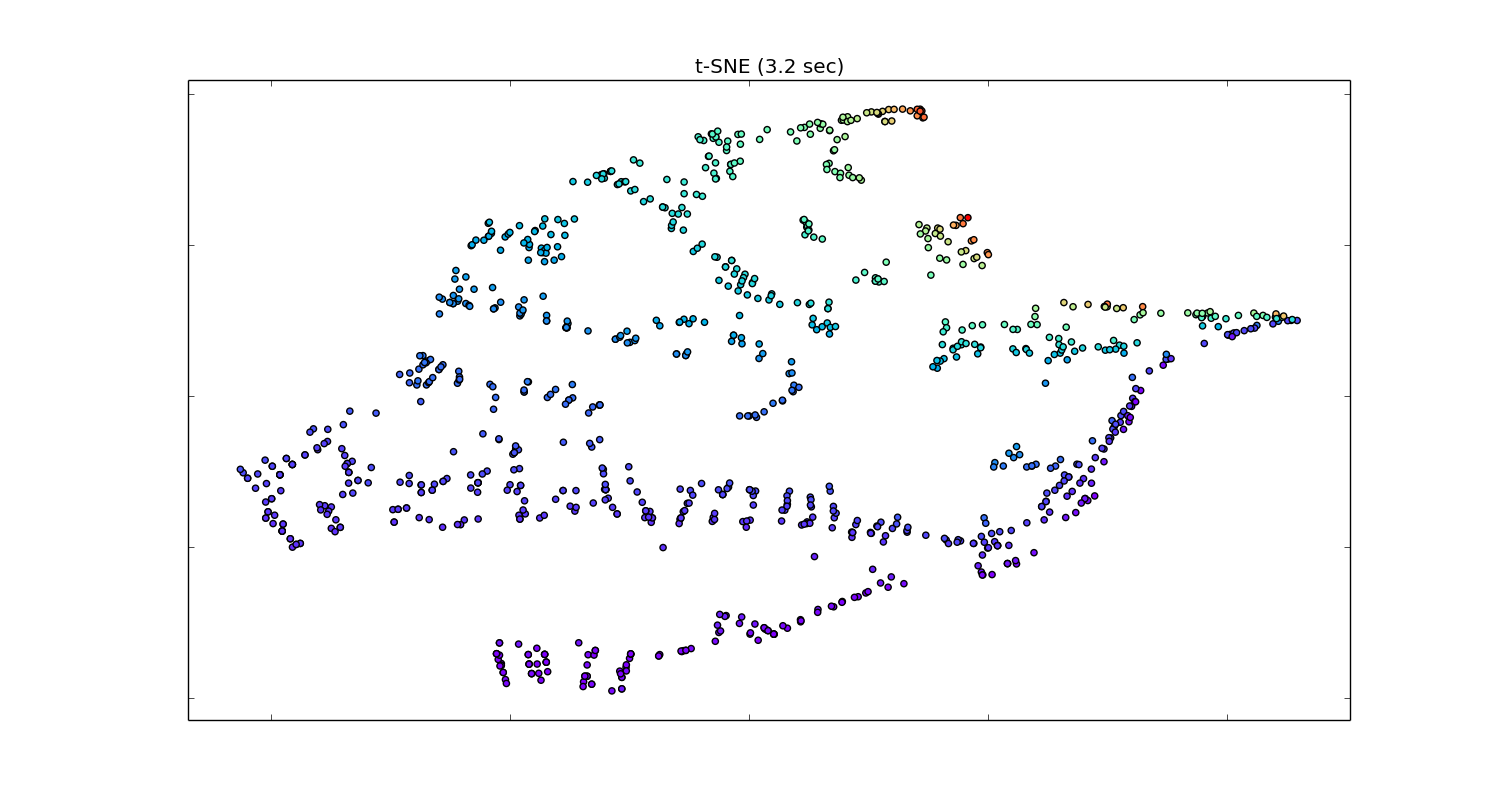
self assigned features+tsne on yelp restaurant dataset



The features are stars, review\_count, latitude, longitude, categories. Since each restaurant has multiple category labels, I assign each label a number and take an average of the labels one restaurant has. Features are then normalized, and plotted using tsne. Color is mapped to the last feature (indicating category). The majority of restaurants are french and italian. They are the ones in dark blue and purple color.

We can see that category is the dominant feature here so the clusters are mostly based on categories.

features:

data\_list=[] # stars, review\_count, latitude, longitude, categories

categories={}

attr\_num = 0 # for giving category a number attribute

for i in data[0:1000]:

#print i.get('categories')

count=0

sum=0

for j in i.get('categories'):

if(categories.has\_key(j)==False):

categories.update({j:attr\_num})

attr\_num += 1

sum += categories.get(j)

count += 1

#print categories

data\_list.append([i.get('stars'), i.get('review\_count'), i.get('latitude'), i.get('longitude'), sum\*1.0/count])

print data\_list

print categories

import numpy as np

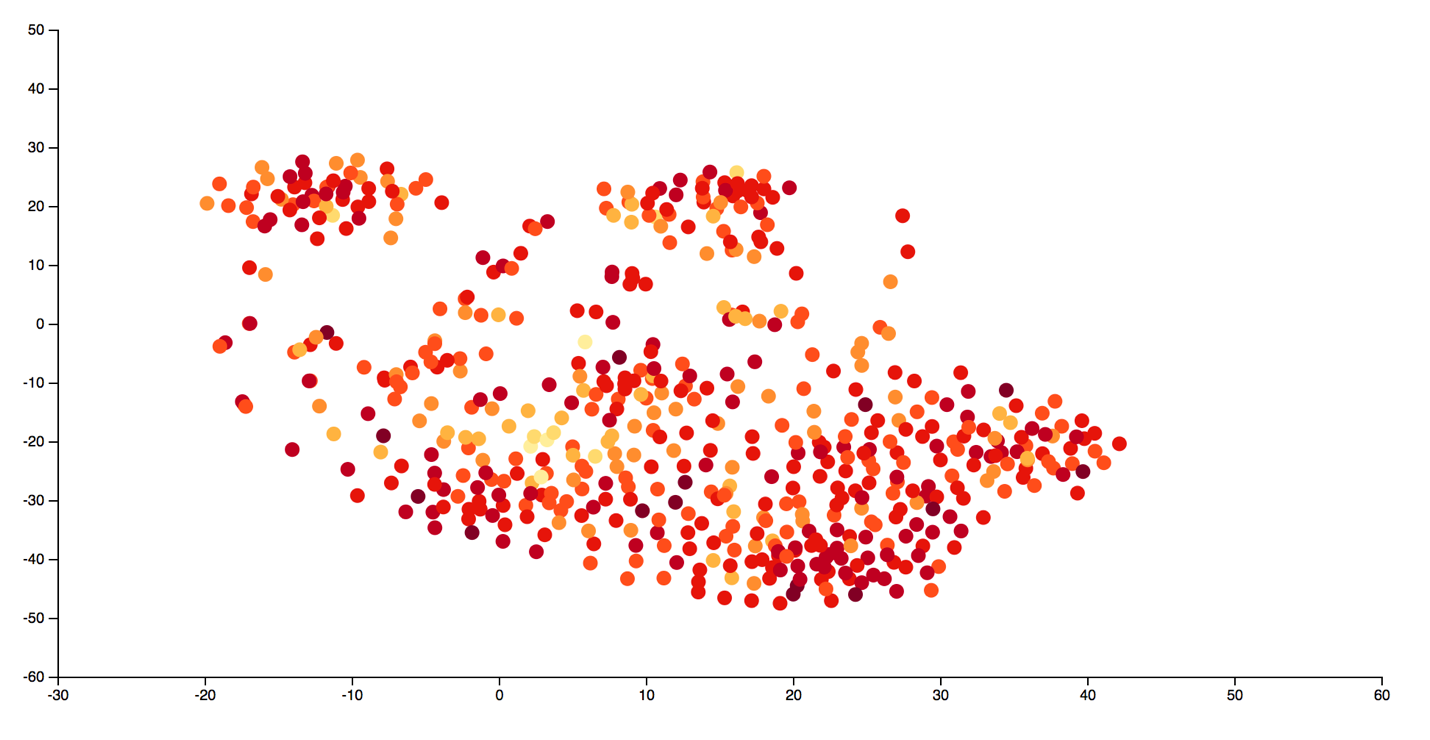
X=np.array(data\_list)

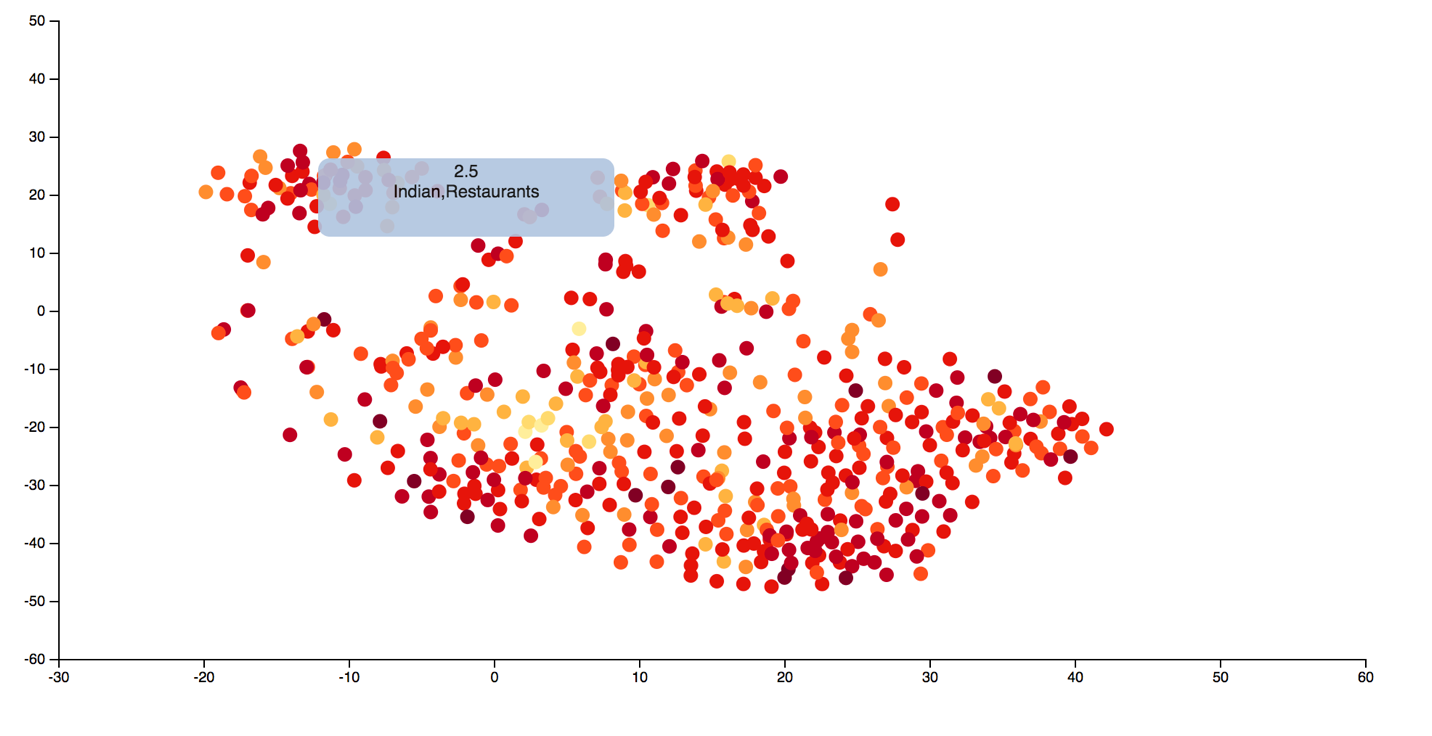
from sklearn import preprocessing

X\_normalized = preprocessing.normalize(X, norm='l2')

Assign each catogery a number and sum/count as the last feature. Color mapped to this feature

tf-idf + tsne on yelp dataset





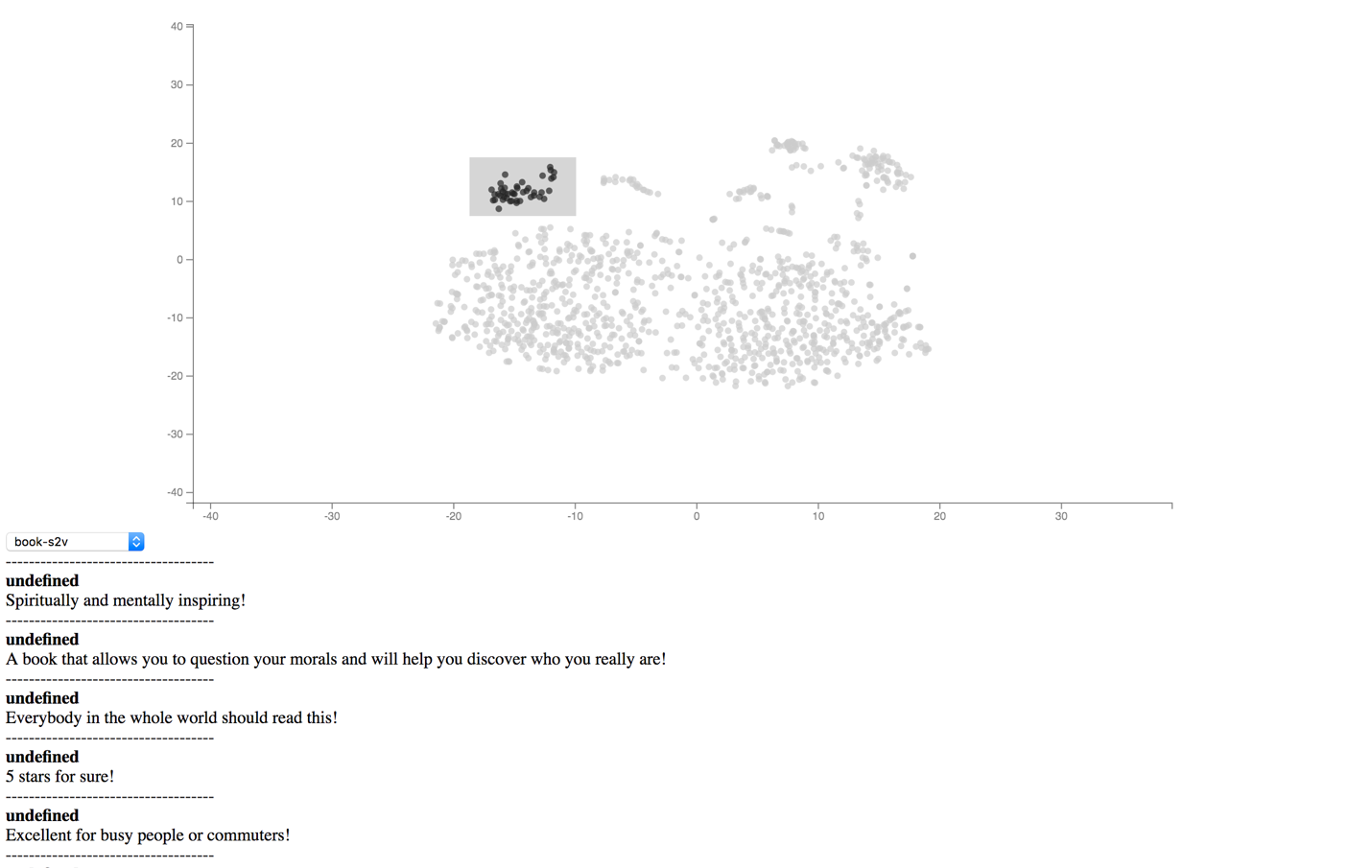
Using tf-idf, the most frequently used words yet unique to one certain restaurant give the most influence in the features.

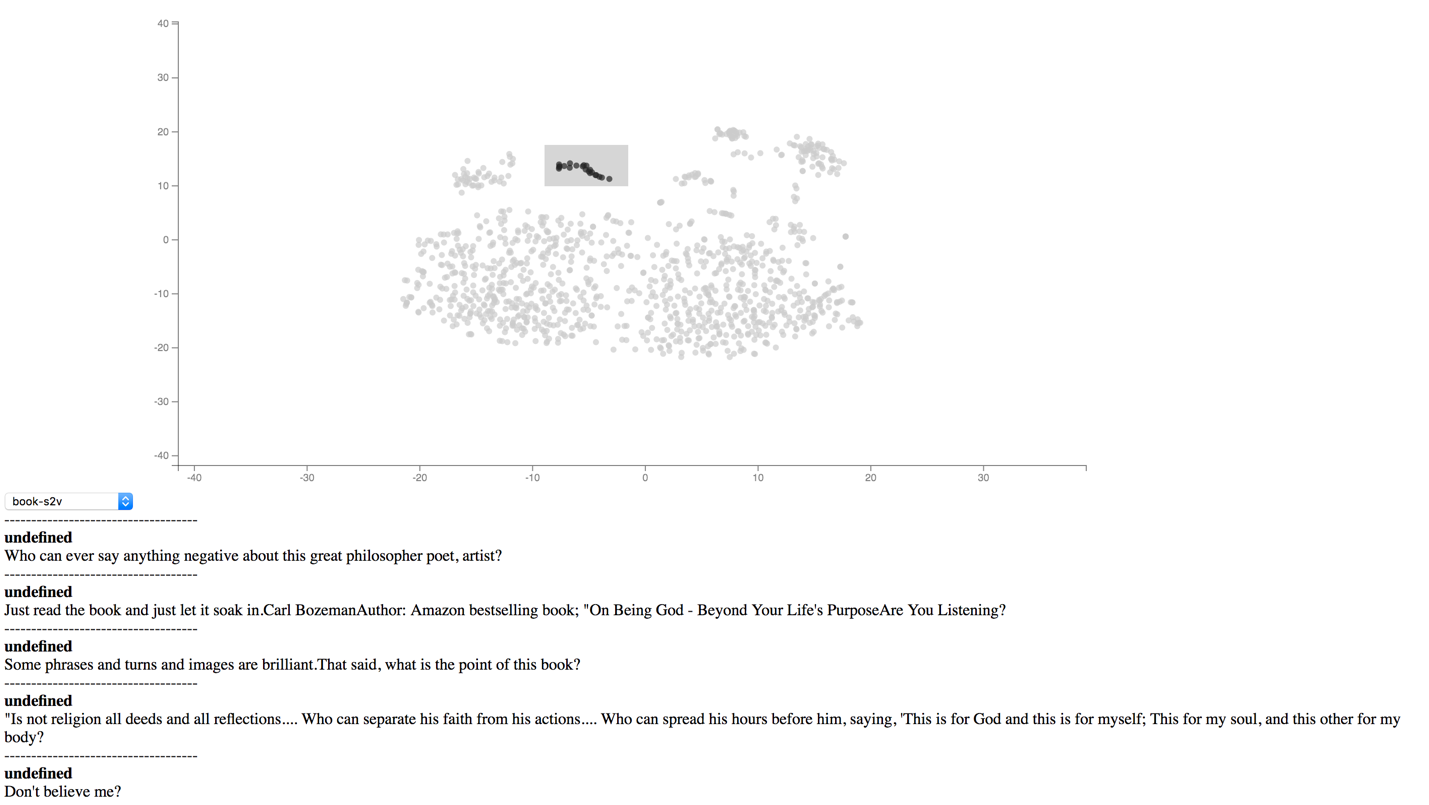
From the plot, we can see that clusters are mostly based on categories as well. Two most obvious clusters are Mexican and Indian restaurants. Color is mapped to the average rating (the darker color the higher stars). I was hoping that, within certain cluster, there would be a graduate change from light to dark color, indicating the rating of the restaurant within a category.

plotted data sample:

{"review\_count":61,"business\_id":"Vq\_gO\_Wsf90AZl0KLB\_gjQ","cat":["Restaurants","Italian"],"stars":4,"y":-30.196680037270418,"x":-1.3372296205093062}

sentence embedding + tsne on amazon books dataset

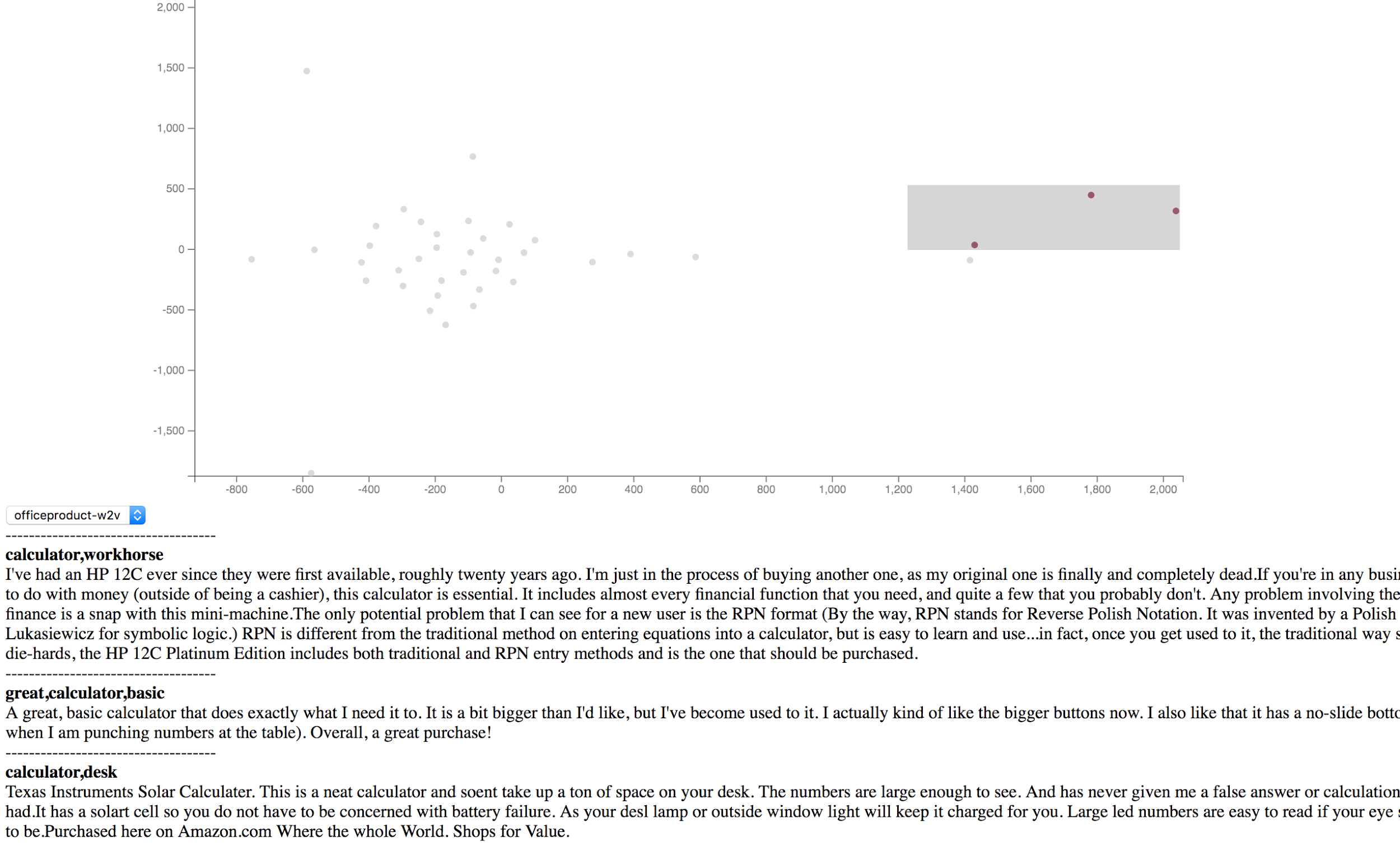


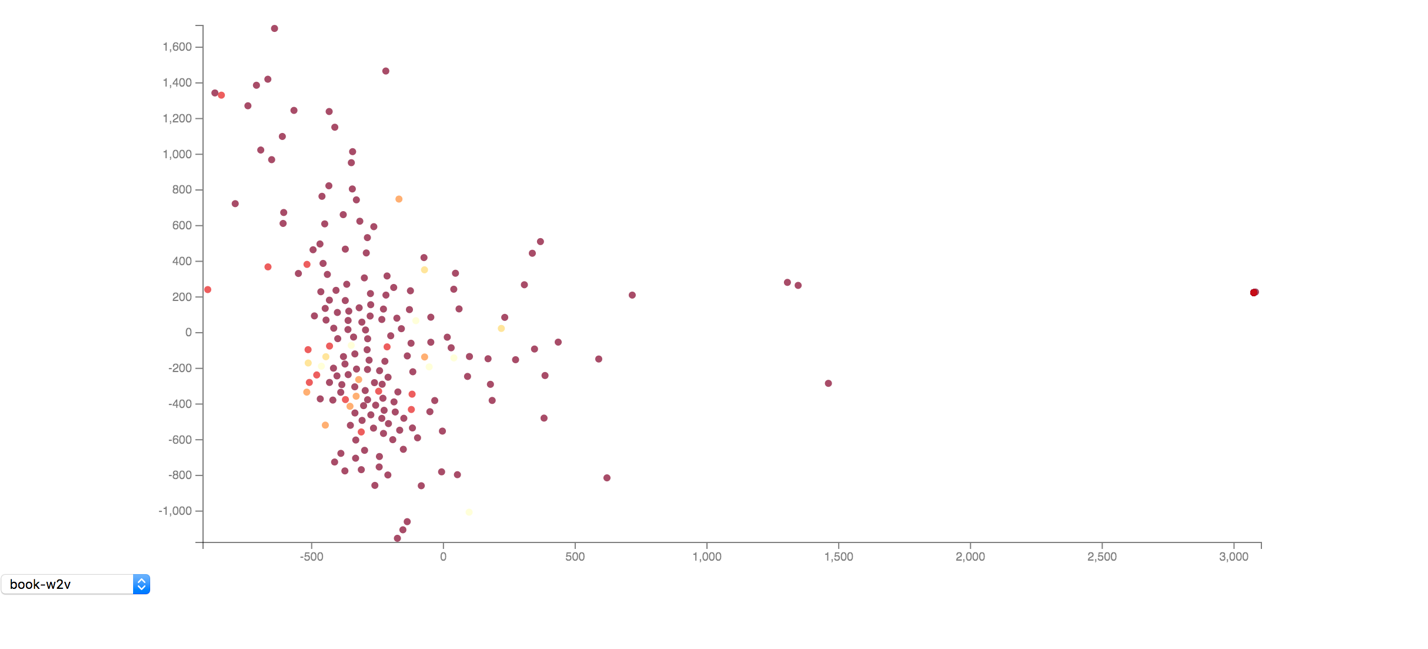


Reviews of a book is chopped into sentences and get a vector from pre-trained sentence embedding model. A circle in the plot is a sentence in the reviews. We can see clusters. The most obvious ones are clustered by the tone of the sentence. For example, the two images show that one of the obvious cluster contains sentences end with “!”, and the other obvious cluster contains sentences end with “?”

It’s hard to say how much this visualization would help people better understand the reviews. I also feel that the embedding model is not perfect. Unlike works, sentences can be less represented by the context. The sentence itself can mean something that is opposite to what the previous sentence mean. And there are much more sentences than words, considering sentences are order of words. So much bigger dataset must be used for sentence embedding models. And I’m not so optimistic that there be a good enough pre-trained model out there for public use. But it’s fun trying it out.

word embedding+ tsne on amazon dataset







The data used there are two items in amazon dataset. The first is the reviews of a calculator. The second is part of reviews of a book. I only use the words in summary. I thought it’s most representative of the reviews. The words go through word2vector pre-trained model (googld model), and a 300 dim vector is outputted for each word. Then I took the average of these vectors (the few words in summary), and plotted it.

Each circle in the plot is a review of the item. The text below are the review within the brush. Using brush is easier for observing the clusters. As plot shows, similar words are grouped together. But unless there are clear clusters that contains useful information, how helpful this is for comprehending reviews is hard to say.

This might be useful for items with large number of reviews. I expect to small clusters outside of one big cluster that are about a certain feature of the item.

TODO:

1 use all words in the review

2 add phrases