A General Framework For Text Semantic Analysis And Clustering On Yelp Reviews

Renfeng Jiang, Yimin Liu, Ke Xu

Mentor: Bryan McCann

Stanford University

{rfjiang, yiminliu, xuke}@stanford.edu

Keywords: Yelp Reviews, Deep learning, word2vec, K-medoids clustering

ABSTRACT

Millions of user reviews have been posted through Yelp. Automatic extraction of useful information from these reviews can be very beneficial for both users and businesses. Recent success in understanding the meaning of a word within the context of natural language processing (NLP) has shed a light on such a practice. *Word2vec,* an implementation of neural network based word-embedding approaches, has shown its ability to accurately capture the semantic similarity among words. The transition from *word2vec* to *doc2vec* (document to vector) or *text2vec* (text to vector), however, has remained an active research. In this study, a *word2vec* based framework for learning Yelp reviews to yield vector/matrix representation of Yelp reviews and Yelp businesses has been developed. It's application in automatic recognition of similarity among different reviews or different businesses has been shown to be successful. Furthermore, the framework is shown to be able to handle practical tasks including businesses recommendation, businesses clustering and reviews clustering.

1. INTRODUCTION

In the area of natural language processing (NLP), understanding the meaning of a word has been critical and intriguing. Many methods have achieved success in capturing the similarity among words. Recently, neural network based word-embedding approach has enabled continuous representation of a word. *Word2vec* is the state-of-the-art program for such an implementation. As the name suggests, each word can be represented by a low-dimensional vector with real numbers. The distances among vectors are shown to be an accurate representation of semantic similarity among words. More surprisingly, word analogy can be achieved by simple vector summation and subtraction.

There are two distinct algorithms behind *word2vec,* skip-gram with negative-sampling training method (SGNS) and continuous bag-of-word model (CBOW). In this study, the CBOW algorithm is chosen for its efficiency. A 1 billion word wikipedia text corpus is used to train the model.

2. METHODOLOGY

The framework we developed incorporates word2vec to map words in reviews into real value vectors, constructs a matrix (essentially a vector set) for each review and each business, a self-defined distance to capture the similarity between two matrices. Basically, our model takes words in reviews as input and output the matrix representation of reviews or businesses. With a proper distance being defined, similarity analysis and category clustering could be performed.

We also notice that *word2vec* could not identify sentiment of words. For example, in the *word2vec* model *"good"* and *"bad"* are semantically very similar words. In other words, the vector of *"good"* and *"bad"* are close to each other and hard to distinguish. As a result, we use the Stanford-POS-Tagger model to tag all words and only nouns are retained.

Our framework has three levels: words, reviews, and businesses (Fig. 1):

- The basic level is the word level. Words in each piece of review are treated as features. POS-tagger is applied to tag all the words and only nouns are retained. Inputting these nouns into word2vec pre-trained model generates different vectors corresponding different words (each noun corresponds to a different vector).
- The second level is the review level. Simple combination of these vectors corresponding to different nouns in a review gives a matrix corresponding to that review.
- The third level is the business level. Applying the same technique above, matrix representing each business is acquired. Here, we use self-defined LXJ distance, named by the first letter of the family name of our group members and defined as below, to calculate the similarities of different businesses. K-medoids clustering or other clustering algorithms can then be applied to categorize different types of businesses.

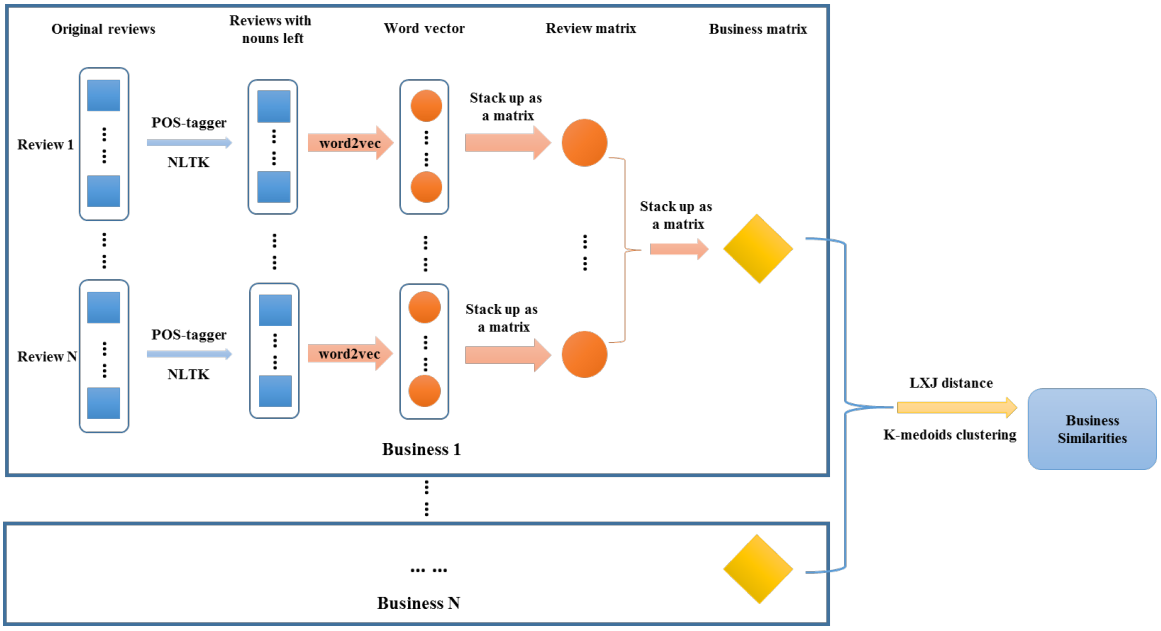

Figure 1. Flow chart of the developed model.

$$A=\begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} \quad B=\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad A \otimes B = \underset{i}{mean}\left\{ \max_j \left(\frac{a_i^T b_j}{\|a_i\|_2 \|b_j\|_2} \right) \right\} \quad LXJ(A,B) = \frac{A \otimes B + B \otimes A}{2}$$

Equation above is the definition of LXJ distance, where A and B are two different business matrices. $\otimes$ is the operator defined by ourselves, which takes the max value among the normalized inner product of different vectors in the two matrices and then take the average. The max operator finds the closest distance between a specified word a_i and all words in matrix B. The mean operator finds the average of such closest distances. The final step is to make the distance definition symmetric.

Overall, our model can find the similarities of reviews and businesses. Combined with K-medoids clustering, the similarities of reviews can be used to categorize different types of reviews and those of businesses can be applied to categorize different types of businesses and even different subsets among one type of business, i.e., American food and Asian food among restaurants. In addition, our model can also calculate the distance between a word and a business, which can achieve smart search. The detailed applications of our model and results are discussion in next chapter.

3. RESULTS

In this section we provide the experimental results on three types of applications based on the proposed model: (1) clustering business and restaurant, (2) keyword search, and (3) clustering reviews.

3.1 Application on the clustering Business and restaurants

To evaluate the effectiveness of the proposed model, we randomly select 500 business and 5000 reviews and perform the clustering based on the proposed model. To begin with we use the proposed model to cluster 500 business into 20 clusters, and partial results are provided below:

Table 1. Clustering results for businesses

Business cluster 1: Name	Yelp tag	Business cluster 2: Name	Yelp tag
Fired Pie	Pizza Salad Restaurants	Emona Studio	Hair Salons Hair Stylists Beauty & Spas
Duzan Mediterranean Cafe & Tapas	Bakeries Food Mediterranean Restaurants	The Barber of Choix +	Hair Salons Barbers Men's Hair Salons Hair Stylists Beauty & Spas
Winery 101	Food Arts & Entertainment Wineries	Hue Salon and Spa	Hair Salons Day Spas Hair Stylists Beauty & Spas
Little Caesars Pizza	Pizza Restaurants	LV Hair @ Alexander's Salon	Hair Salons Hair Extensions Hair Stylists Makeup Artists Beauty & Spas
Adela's Italian	Restaurants Italian	Pure Radiance	Hair Salons Hair Extensions Hair Stylists Beauty
Olio Trattoria	Restaurants Italian	Faces Hair & Color Studio	Hair Salons Beauty & Spas

It can be seen that these business are similar restaurants/hair salons and the yelp tag helps provide a baseline and validation of our clustering model.

To better demonstrate the clustering results, we propose to use the multidimensional scaling (MDS) to visualize the data. MDS projects our data down to two-dimensional space so that the data can be plotted based on the Euclidean distance. Here is the visualization result in figure 2:

All 500 business are plotted based on the relative distances; as a baseline, these businesses are color-labeled based on their business categories obtained from Yelp's category. From the graph, we can clearly observe that the restaurants (colored in red) form a cluster, which matches with Yelp's restaurant label.

After validating the clustering result, we can go one step further to categorize different subsets of businesses among a single business, for example, different types of restaurants among all the restaurants, say Asian, American, Mexican, etc., which is shown in Figure 3, along with our another application of smart search, which is discussed in 4.2.

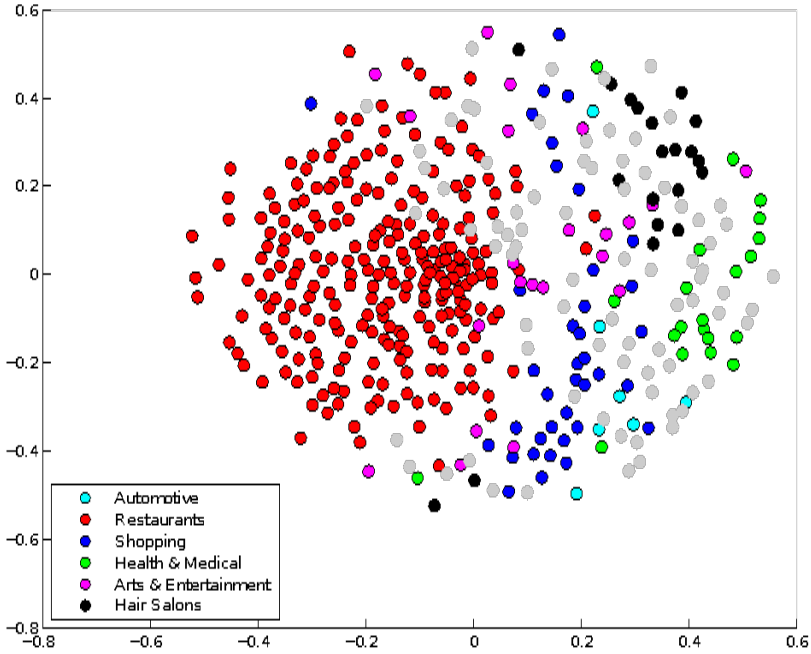

Figure 2. Labeled businesses by similarities.

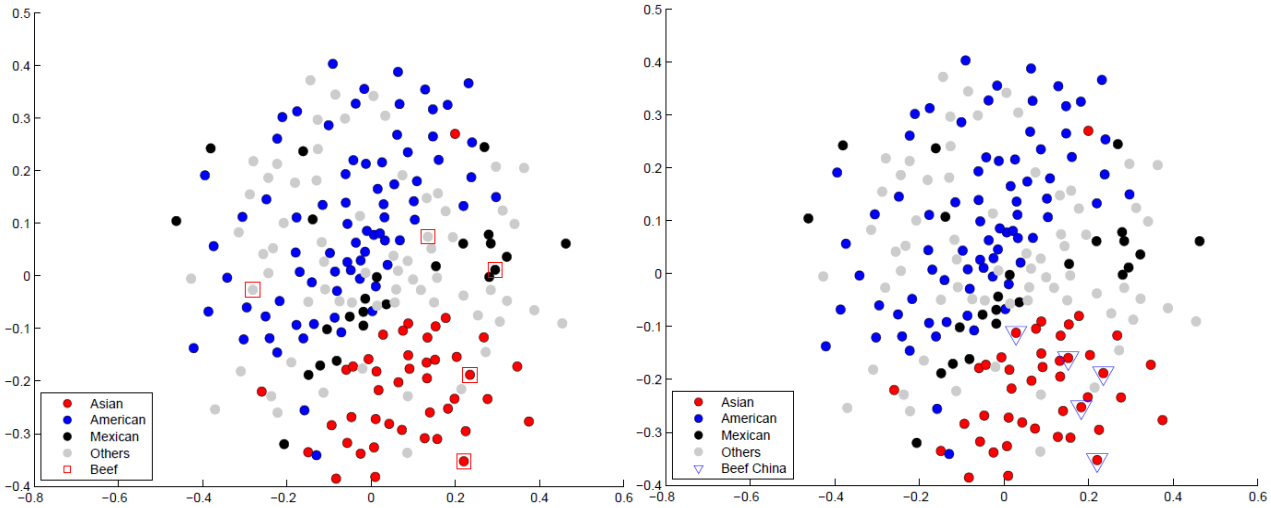

Figure 3. Labeled different types of restaurants and search results after inputting beef & beef China.

3.2 Application on the key word search

To extend the above clustering function so that user can search for a specific business, the proposed model could calculate the similarities between an input item (vector) and a business (matrix). Such system can achieve personalized smart search. A keyword is first decomposed into several tokens and each token would be mapped to a high dimensional vector. We then aggregate these vectors, treated as a 'business matrix' and find its cluster members.

This model works very well, and some results are provided below in Table 2.

Table 2. Search results after inputting beef & beef China

Input	
beef	beef China
Results	
Heng's Kitchen (Chinese Restaurants) Thé & Toast (Food Coffee & Tea Japanese Chinese Restaurants) Billy's BBQ (Soul Food Southern Barbeque Restaurants) ⋮	Heng's Kitchen (Chinese Restaurants) Thé & Toast (Food Coffee & Tea Japanese Chinese Restaurants) Pho Bistro(Vietnamese Chinese Restaurants) ⋮

The first input is 'beef,' which results in a variety of restaurants. The second input is 'beef china,' leading to Chinese-related restaurants. By investigating these restaurants manually, we concluded that these types of restaurants do distinguish themselves. For example, it is hard to know Heng's Kitchen is famous for its beef, but we dig deep into the reviews and found out that most reviews were talking about its delicious beef soup noodles. Considering the data we investigated only has 500 businesses, in which there are only a few Asian restaurants, it is reasonable that the first two results of the two searches are the same.

Using the similar visualization techniques described above, we further labeled out the results by circles and squares on the graph to help further understand and validate the results (Figure 3).

These 'dot' graphs are based on the business type of only restaurants for better visualization. From the left figure of Figure 3, we can see that the results (dots in squares) are more scattered across all types of restaurant types. From the right figure, the results (in circles) are clustered completely within the 'Asian' category.

From these experimental results, we believe that this application can be used for user's searching on (1) business type, (2) food type, or (3) to provide recommendations.

3.3 Application on the clustering the reviews

Finally, we think it would also be interesting to see how the proposed framework could be applied to cluster the reviews within the same business. One motivational scenario would be: for a business with many reviews, it is time-consuming for a customer to read all the reviews. Sometimes reading the oldest/latest review would create certain bias. Therefore, it is helpful to come up with a highlighted review, to give the customer a few review 'exemplars,' each of which represents a group of similar reviews serving as a summary of this business.

Table 3 below is the experimental result for a business with more than 35 reviews. We clustered them into four categories. One exemplified cluster listed below contains reviews regarding wait time (Table 3). 'Cluster 2' is mostly about food. As shown below, each review basically mentions one or two types of food. Also for each cluster, we extracted the centroid and used it as the highlighted summary (exemplar) of similar reviews.

Table 3. Two clustering results: one relates to wait time and the other relates to the food.

Cluster 1: Wait time (centroid)	**Cluster 2: Food (centroid)**
"I waited for 55 minutes:" Went to the newest Crazy Pita at Downtown Summerlin today … **I waited for 55 minutes** and still had not received my food, so I went inside to ask what was going on with my order…the woman next to me said she **waited about 50 minutes**…	Got the Crazy Pita **Salad** with added **chicken**. **Salad** was a great size …However, the **chicken skewer** was pretty overcooked, charred, and dry...Not your typical **whole wheat** hard and **floury pita**…**Key lime pie** was amazing!
Cluster members:	**Cluster members:**
1. **"Wait time is ridiculous:"** There is no excuse to take **50 minutes** to get a pita to a customer. … I watched the line extend out the door multiple times, then an employee would tell the people about the long wait and they would all leave…	1. **"We love Mediterranean food:"** We dined here today. ..We love **Mediterranean** food, especially **kabobs or pitas**. We got the **kabob platers.** While the **chicken and shrimp** kabobs were pretty good the rest of the platter was terrible! The **veggies** were **soggy** and …
2. **"All food is taking 45 minutes to come out:"** The girl behind the counter said "Did you hear? … Would you like to order?" Ummm - no! How can a **pita take 45 minutes**?	2. **"lamb and shrimp:"**…We had **lamb and shrimp** with a side of what they called **feta salad**..
3. **"I waited 45 minutes"** from cash wrap to food on the table to … the food is OK. Not **worth 45 minutes**. It's just a sandwich at the end of the day or a plate of romaine...but I can't afford the time I lose. Thanks.	

4. SUMMARY AND FUTURE WORK

4.1 Summary

In this project, we have demonstrated a framework that enables a wide range of applications on text-review analytics. We have demonstrated strong and convincing experimental results, ranging from (1) clustering business and restaurant, (2) keyword search, and (3) clustering reviews. All the experimental results are poised to demonstrate that the proposed model serves as a systematic and effective way to provide a semantically-meaningful clusters that enable a wide range of applications.

4.2 Future Work

We plan to feed our model with all the reviews of over one million to:

- Categorize all the different types of businesses;
- Do more specific categorizations on restaurants;
- Get better search results with enough reviews feed.
- Compare with Latent Dirichlet allocation

REFERENCE

1. Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." Proceedings of the conference on empirical methods in natural language processing (EMNLP). Vol. 1631. 2013.
2. Le, Quoc V., and Tomas Mikolov. "Distributed representations of sentences and documents." arXiv preprint arXiv:1405.4053 (2014).
3. word2vec: https://code.google.com/p/word2vec/