Dish Discovery via Word Embeddings on Restaurant Reviews

Chih-Yu Chao[1], Yi-Fan Chu[1], Yi Ho[2], Chuan-Ju Wang[1,3], and Ming-Feng Tsai[4]

[1]Department of Computer Science, University of Taipei, Taipei 100, Taiwan
[2]Department of Engineering Science and Ocean Engineering, National Taiwan University, Taipei 106, Taiwan
[3]Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan
[4]Department of Computer Science, National Chengchi University, Taipei 116, Taiwan

ABSTRACT

This paper proposes a novel framework for automatic dish discovery via word embeddings on restaurant reviews. We collect a dataset of user reviews from Yelp and parse the reviews to extract dish words. Then, we utilize the processed reviews as training texts to learn the embedding vectors of words via the skip-gram model. In the paper, a nearest-neighbor like score function is proposed to rank the dishes based on their learned representations. We brief some analyses on the preliminary experiments and present a web-based visualization at http://clip.csie.org/yelp/.

Keywords

dish discovery, word embeddings, dish-word extraction

1. BACKGROUND

With the growth of social media, corporations, such as Yelp, have accumulated a great number of user generated content (UGC). In the literature, some studies have been conducted with a perspective of finding critical information hidden in the content [2]. While much has been proposed on accurate sentiment interpretation towards reviews and recommendation, little has focused on dish-level analysis [4]. In this paper, therefore, we aim to provide a novel framework for automatic dish discovery from restaurant reviews via the embedding techniques. We employ regular expressions to first parse restaurant reviews to extract dish words, and then utilize the processed reviews as training texts to learn embedding vector of each word via the skip-gram model [3]. In addition, a nearest-neighbor like score function is proposed to rank the dishes via their learned representations. Preliminary experiments are conducted on a real-world restaurant review dataset collected from Yelp Data Challenge.

2. METHODOLOGY

Our methodology mainly consists of three parts: 1) dish-word recognition, 2) word embedding learning, and 3) dish score calculation. As alluded to earlier, UGC usually incorporates a degree of noise and different language usages; therefore, extracting dish names from user reviews is a complicated task. For example, observed from the dataset, users tend not to write the full name of a dish in their reviews; instead, the last word or the last two words are often written in the reviews. To grapple with this issue, we use regular expressions (*regexps*) to extract dish names from the user reviews. However, this also give rise to an issue that a certain dish in a restaurant may be of the same name in other restaurants, which may induce the problem of ambiguity and lower the accuracy of matching the correct dish name. So, we attach a dish name with its restaurant name to solve the ambiguity problem.

We then utilize the collection of processed reviews as training texts to learn embeddings of each word in the reviews via a continuous space language model, the skip-gram model. After the training phase, each word (including every dish) is represented by an n-dimensional vector (called the embedding of this word). Inspired by the k-nearest neighbors algorithm, we define the score for every dish d as:

$$S(d) = \sum_{k=1}^{m} \lambda_k f_k(d), \qquad (1)$$

where $f_k(d) = \frac{k}{\sum_{i=1}^{k} \|w_d - w_{s_i}\|}$, m is the total number of positive sentiment words considered, λ_i $(i = 1, \cdots, m)$ is a weighting parameter. In addition, s_i denotes the i-nearest positive sentiment words of the given dish d, and $w_d, w_{s_i} \in R^n$ are the vector representations of the dish d and the sentiment word s_i, respectively.

In an extreme case (1) of $\lambda_m = 1$ and $\lambda_i = 0$ for $i = 1, \cdots, m-1$, this score function implements the concept of the average Euclidean distance between a dish and all the positive sentiment words; while in the case (2) $\lambda_1 = 1$ and $\lambda_i = 0$ for $i = 2, \cdots, m$, the scored is obtained with the closest positive sentiment words to the dish.

3. EXPERIMENTS

Our preliminary experiments involve a real-world restaurant review dataset collected from Yelp Data Challenge.[1] We first choose the top 100 restaurants containing the most reviews in the area of Las Vegas and then manually parse

RecSys 2016 Poster Proceedings, September 15-19, 2016, USA, Boston.

[1] https://www.yelp.com/dataset_challenge

Table 1: Top-3 dishes of Sushisamba Las Vegas.

	Ranking methods		
	Frequency	Case (1) Average distance	Case (2) Minimum distance
precedence ↓	Sea Bass (**364**, 0.706, 0.787)	Soft Shell Crab (4, **0.737**, 0.899)	Seaweed Salad (25, 0.706, **0.910**)
	Peruvian Corn (**125**, 0.713, 0.809)	Lamb Chop (11, **0.735**, 0.858)	Soft Shell Crab (4, 0.737, **0.899**)
	Spicy Tuna (**81**, 0.702, 0.787)	Samba Sushi (14, **0.735**, 0.845)	Green Bean Tempura (20, 0.703, **0.877**)

the menu of each restaurant from its official website. Out of those 100 restaurants, we extract the restaurants with a complete menu, setting the reviews of those restaurants and their menus as our dataset. In summary, there are 69 restaurants and 95,578 reviews in total after the filtering; the number of words per review in average is about 147 and the vocabulary size is 46,017.

For preprocessing the reviews to identify each dish, here we demonstrate the matching rule via the example dish, `Housemade Country Pate`; its *regexps* can be set as:

```
(Housemade*|Country*)+Pat[a-z]+(s|es|ies)?,
```

which is set to match `Country Pate`, `Housemade Pate`, or `Housemade Country Pate`. If a match of the dish is found, we replace the name of the dish with its full name and append the name of the restaurant to an *underscore* symbol, modifying it to `Housemade-Country-Pate_Mon-Ami-Gabi`. After the modification and replacement, the score of each dish d is calculated via the score function defined in Eq. (1), where the positive sentiment words are selected from the lexicon provided in [1], and only top 200 most frequent sentiment words in our dataset are adopted. For the representation learning, the word2vec toolkit[2] and the skip-gram model are adopted, in which the context (window) size for the skip-gram model was set to 5 and the dimensionality of the word vectors was set to 200.

Table 1 tabulates the top-3 dishes ranked by the proposed approach for the restaurant `Sushisamba Las Vegas`. In the table, the dishes in each column are the top-3 results ranked by (a) their number of occurrences, (b) the score based on average distance, and (c) the score based on minimum distance; (a), (b), and (c) correspond to the three numbers in the parentheses. From the table, it can be observed that none of the top-3 most frequently mentioned dishes occurs in the lists ranked by our method (both cases (1) and (2)), which is due to the fact that these high frequent dishes might not be surrounded with positive words and sometimes with negative reviews. For example, there is a review for `Peruvian Corn` within a comment of "`The Peruvian Corn was awful`" in the dataset. This phenomenon indicates that the most frequent dish mentioned in the reviews may not be the most recommended dish by users. In addition, the proposed method is capable of finding dishes that might not frequently occur in reviews, e.g., `Soft Shell Crab`, and thus can provide more diverse results.

Figure 1 visualizes the positive sentiment words and the top-3 dishes ranked by the proposed method based on the learned representations. From the figure, we can observe that the words with similar meanings are usually close to each other, such as the words in the circle including *good*, *best*, and *great*. Furthermore, for the extreme case (1), the dishes close to the centroid of all the positive words tend to have higher scores and their contents in the reviews may be more diverse. On the other hand, for the case (2), the top-ranked dishes are close to a certain sentiment word; for example, the dish `Seaweed Salad` is top-ranked and far from the centroid in the case (2), but its score based on the average distance is rather low than the other top-3 dishes in the case (1).

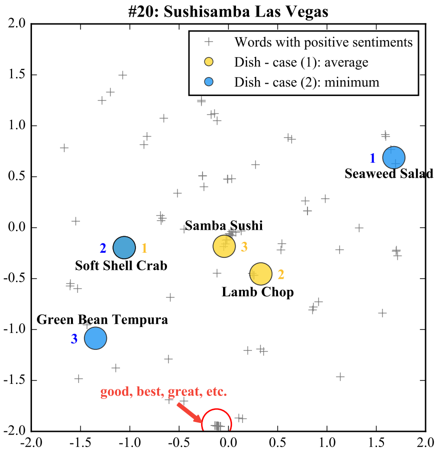

Figure 1: 2-D Visualization on the top-3 recommended dishes and positive words.

4. CONCLUSIONS AND FUTURE WORK

This paper proposes a novel framework for dish discovery from restaurant reviews via word embedding techniques. This framework can be of great help in discovering or recommending dishes via only the review texts based the proposed score function. Although in this preliminary work, we have not conducted quantitative evaluation on our experiments, the given example and the visualization results demonstrate the novelty and the potential of the proposed approach.

In the current work, we only consider two extreme cases of the score function; hence, considering different settings of the score function and quantitatively analyzing the corresponding results will be one of our important future work. Also, a food-oriented lexicon will be considered in the future. Most importantly, the size of the collected texts is vital to representation learning algorithms, so we are now collecting more data from Yelp and plan to conduct our experiments on a much larger dataset.

5. REFERENCES

[1] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proc. ACM KDD*, pages 168–177, 2004.

[2] J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proc. ACM Recsys*, pages 165–172, 2013.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[4] M. Trevisiol, L. Chiarandini, and R. Baeza-Yates. Buon appetito: recommending personalized menus. In *Proc.of ACM HT*, pages 327–329, 2014.

[2] https://code.google.com/archive/p/word2vec/