EXTRACT RESTAURANT ASPECT WORDS BY USING WORD2VEC MODEL FROM YELP REVIEWS

by

Penghao Wang

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

May 2017

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Julia Taylor Rayz, Chair
Department of Computer and Information Technology
Dr. John A. Springer
Department of Computer and Information Technology
Dr. Victor Raskin
Department of Linguistics

Approved by:

Jeffrey L. Whitten
Head of the Departmental Graduate Program

For my parents, Zhenying Wang and Qing Xia.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my thesis committee for their useful advice and comments. Dr. Julia Rayz, Dr. John A. Springer, and Dr. Victor Raskin gave me invaluable help.

I also thank my friends, Xiangying Sun, Peng Li, and Yining Teng, who helped me labeling the data. More than that, I would like to thank the Statistical Consulting Service Department for their advice in selecting the sample size and its justification.

Moreover, I would like to thank Anita Schafer who gave me editing advice. The help from her was extremely useful.

Finally, I wish to thank my parents for their support. The six years I spent at Purdue have been an invaluable gift.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CRF	Conditional Random Fields
LDA	Latent Dirichlet Allocation
LSI	Latent Semantic Indexing
NLP	Natural Language Processing
POS	Part-of-Speech Tagging
t-SNE	t-Distributed Stochastic Neighbor Embedding

GLOSSARY

Aspect extraction - a sub-task of sentiment analysis that consists of identifying opinion targets in opinionated text. (Poria, Cambria, & Gelbukh, 2016)

Opinion mining tool - a tool that processes a set of search results for a given item, generating a list of product attributes (quality, features, etc.) and aggregating opinions about each of them (Dave, Lawrence, & Pennock, 2003).

Word embedding - natural language processing technique to map words or phrases into a vector of numbers.

ABSTRACT

Author: Penghao, Wang. M.S.
Institution: Purdue University
Degree Received: May 2017
Title: Extract Restaurant Aspect Words by Using Word2vec Model from Yelp Reviews
Major Professor: Julia T. Rayz

Many customers use Yelp restaurant reviews to determine where they will eat. However, it is almost impossible for users to read all the reviews because of the large number of reviews. Customers usually have different concerns about restaurants like ambiance, service, and food quality. Providing customer reviews that concentrate on the aspects that they are concerned with would save customers time and help them to find better restaurants. This thesis is an investigation into selecting such useful aspects. To achieve that, one needs to detect which aspects are mentioned in the review. In this study, the researcher used a Word2vec model to detect the aspect words and tested the performance of the model on the manually labeled data and compared the results to a statistical model. The author presented three tasks the Word2vec model could do: (1) detecting food categories, (2) detecting informal words, and (3) detecting typos and abbreviations. As a result, the author found the Word2vec model yielded better performance results than the statistical topic model on the aspect extraction task.

CHAPTER 1. INTRODUCTION

In recent years, many reviews of restaurants were created. In December of 2016, there were about 121 million reviews in the Yelp website from all around the world. Many customers choose what to eat and where to visit based on online reviews. The reviews and star ratings did change some of the customers' behaviors. Numerous customers looked at a single overall star and read some reviews to make decisions. Even after reading some of the recommended reviews, customers may not find the information they are seeking. For instance, if a customer wants to eat sushi, the reviews he or she wants to read are those that contain opinions about sushi. Customers might waste time reading reviews that are not of interest to them. In this case, recognizing which aspects are in the reviews is useful for customers to help choose which reviews they want to read. Aspect words mean the words of terms that reflect the aspect of the entities (Mukherjee & Liu, 2012). This study used the Word2vec model to extract aspect words and tested the performance of the Word2vec model.

Word2vec model (Mikolov, Chen, Corrado, & Dean, 2013) is a state-of-the-art method to represent words in vector space. It has become a widely used word embedding model in recent years. By using a Word2vec model to map words, similar words have similar vector value. The Word2vec model can find similar words by calculating the similarity of word vectors. For instance, the five most similar words to "France" found by the Word2vec model were Austria, Belgium, Germany, Italy, and Greece (Collobert et al., 2011). Based on the results in the previous paper, the author believed Word2vec might be useful to extract words that reflect different aspects of the restaurants.

1.1 Scope

The purpose of this study was to test the state-of-art method for extracting aspect words from Yelp reviews of restaurants. More specifically, the study focused on the performance of a Word2vec model for extracting different aspect words that relate to food, service, restaurant ambience, and price.

The task to detect various opinions about different aspects of restaurants could be separated into two parts. The first part is to identify restaurant aspects that were mentioned in the review. The second part is to extract the opinions about the aspect. This research focused on the first part.

The research focused on the reviews from Yelp. In the study, topic model, clustering technique, and a Word2vec model were used to analyze the reviews and extract the aspect terms in the reviews. The study tested the performance of the Word2vec model in extracting aspect words from the reviews of restaurants.

1.2 Significance

Reviews are critical for restaurants. For a restaurant, reviews could be used to improve the food, service, and other aspects. For the customers, the reviews help gather other consumers' opinions about the restaurants. In that case, analyzing the reviews is crucial. However, because of the numerous reviews that exist on Yelp, it is impossible for customers or restaurants to read all the reviews and make decisions.

Many customers made decisions based on several reviews and overall stars given to a restaurant because it was impossible to spend a lot of time reading reviews to make decisions. Reading only a limited number of reviews has its drawbacks, that is, the customer gets a small amount of details about the restaurant. For instance, a consumer who cares about the food quality wants to read the reviews that contain opinions about food quality. However, Yelp does not offer the customer the options to do that. In this case, analyzing opinions about different aspects of restaurants is important.

To analyze the opinions about various aspects, the first step is to extract aspect words of the restaurants in the reviews. The second step is to find the opinions that targeted the reviews. The quality of the extracting aspect words massively affected the following analysis. Therefore, extracting the correct restaurant aspect words and figuring out which aspects were mentioned in the reviews are significant for the customers and the restaurants.

1.3 Research Question

The research sought to answer the following research question: is it possible to extract words or terms that reflect restaurant aspects from Yelp reviews by using the Word2vec model? More specifically, what is the performance of the Word2vec model for extracting the words from the reviews that relate to four aspects: food, service, ambiance, and price (Sajnani, et al., 2012)? For instance, “Love this place. I have been there twice and the food has been outstanding. try the catfish - it is yummy,” is a review from the dataset. The research tried to extract the words like “food” and “catfish” which were related to different aspects.

1.4 Assumptions

The assumptions for this study include:

- The restaurants which shared the same Restaurants ID were the same restaurants.
- The datasets contained the correct information about the reviews and restaurants.

1.5 Limitations

The limitations of this study include:

- The study was limited to the data from the Yelp challenge dataset.
- The study focused on the English reviews.
- The study was only focusing on the reviews of the restaurants.
- The study concentrated on cities in Canada and the United States.

1.6 Delimitations

The delimitations of this study include:

- The study did not focus on analyzing the location of the restaurants.
- The study did not focus on analyzing the pictures.
- The seasonal trend was out of the scope of this study.
- The research did not focus on short reviews (Less than four words).

CHAPTER 2. REVIEW OF RELEVANT LITERATURE

2.1 Definitions and Background

Many efforts were invested in opinion mining and review mining. These techniques were very useful for analyzing the reviews of different restaurants. In the framework of opinion mining and review mining, aspect-based review mining was one of the main foundations (M. Hu & Liu, 2004; Liu, 2012; Pang & Lee, 2008). An important task was to extract aspects of different entities. The task could be divided into two sub-tasks. The first sub-task was extracting aspect terms from an opinion corpus. The second sub-task was clustering synonymous aspect terms into categories where each category represented a single aspect, which was called an aspect category (Mukherjee & Liu, 2012). The study focused on the first sub-task.

M. Hu and Liu (2004) studied summarization of customer reviews based on product features which were based on word frequency. In their study, they divided the features into two types: explicit and implicit. Some of the features are explicit. For instance, in the review "The size of the laptop is OK," the 'size' is the explicit feature. For a review such as, "The laptop can't fit in my backpack," has the implicit feature 'size' (M. Hu & Liu, 2004). In this study, the author concentrated on the explicit features or terms that reflect food, service, ambience, and the worthiness of the restaurants.

There were lots of useful studies that focused on classification approaches for the aspect extraction and opinion mining. Pang and Lee (2008) made a study of opinion mining. The focus area of the research was "methods that seek to address the new challenges raised by sentiment-aware applications, as compared to those that are already present in more traditional fact-based analysis" (p. 17). In the study, several topics were discussed. Pang and Lee (2008) explained the factors that made opinion mining difficult and the limitation of textual analysis in the recent years: how to come up with the right set of the keywords and order dependence problems. Also, several approaches were listed in the paper: domain adaptation (Dave, Lawrence, &Pennock, 2003), unsupervised lexicon induction methods (Beineke, Hastie, Manning, & Vaithyanathan, 2004; M. Hu & Liu, 2004) and classification approaches based on relationships between product features

(Popescu & Etzioni, 2007; Snyder & Barzilay, 2007). Lerman, Blair-Goldensohn, and McDonald (2009) did a study that aimed to test different approaches of summarizing the important points from the reviews by training a ranking SVM model. Lerman, et al. (2009) announced that “humans prefer sentiment informed summaries over a simple baseline. This shows the usefulness of modeling sentiment and aspects when summarizing opinions” (p. 521). In their study, they tried to evaluate different methods to summarize the important points in the reviews. A ranking SVM model was created that reduced the 30% error over the previous best summarizer.

An ideal opinion extraction tool would generate a list of product attributes and find opinions about the attributes. For instance, by analyzing the reviews of laptops, the ideal opinion extraction tool would extract the features of the product and find the opinions about these features. To achieve this goal, Dave et al. (2003) conducted a study aimed at finding an ideal opinions tool. They trained a classifier “based on the relative frequency of each part of speech tags” (Dave et al., 2003, p.2). The classifier used a corpus of reviews and tried to find the latent principles of the product reviews. In this study, several interesting conclusions were announced. Dave et al. found that “some reviewers use terms that have negative connotations, but then write an equivocating final sentence explaining that overall they were satisfied. Others compare a negative experience with one product with a positive experience using another" (p. 526). That was critical for the proposed study because it also happened in the reviews of the restaurants. How to deal with them was a problem. These reviews also could be noise in the data. Additionally, Dave et al. (2003) found another useful fact that "many of the reviews are very short" (p. 526). The idea was very significant for this study because it was hard to extract the aspect words in the short reviews. In their study, they suggested that cutting them out would help to improve classification performance.

These topics gave the author a view that the opinion mining and review mining have lots of challenges. Also, it gave the author an overall view of the history of the opinion mining process.

2.2 Aspect Extraction Based on Statistics or Lexicon

Hu and Liu (2004) tried to study a problem of opinion summarization of customer reviews based on product features. In the study, they separated the task into two sub-tasks. The first sub-task was to identify the features in the reviews and rank the product features based on the frequencies. The second sub-task was to find the opinions based on the extracted features. In their study, they mainly focused on the first sub-task, that was, extracting the product features that were mentioned in the product reviews. In the study, they used an Apriori algorithm (Agrawal, Srikant, et al., 1994), association rule mining, and Part-of-Speech Tagging (POS) technique to extract and prune the features. They separated the product features into frequent features and infrequent features. Also, they created an opinion word list to help find the opinion in the review. Hu and Liu used a manually tagged features list for the product to test. They stated, "In summary, with the average of recall of 80% and the average precision of 72%, we believe that our techniques are quite promising." (Hu and Liu, 2004, p. 760)

After the previous study, another important paper about product features and opinion extraction was published. The new system and method improved the approach used in the earlier study. Popescu and Etzioni (2005), created the system OPINE, an unsupervised system, which enhanced the method used in the paper of Hu and Liu.

Popescu and Etzioni (2005) described the problem of extraction opinions or features as four following sub-tasks:

- Identify product features.
- Identify opinions regarding product features.
- Determine the polarity of opinions.
- Rank opinions based on their strength. (p. 339)

Their study focused on the first three parts. There were two important differences between the Hu's system and the OPINE system. First, Hu's system was based on the association rule mining and Part-of-Speech Tagging. The OPINE system was based on Point-wise Mutual Information (PMI) to extract product features. Second, Hu's system was only based on the reviews from the dataset. The OPINE system used Web PMI statistical data to help the model. As a result, according to Etzioni and Popescu (2005), "OPINE's precision of the feature extraction task was 22% better, though its recall was

3% lower on Hu’s data sets.” (p. 339) These studies were primary product features extraction studies. They had a big influence on this area of the research. Their approaches were obsolete because of the using of old version of PMI. These studies were heavily dependent on the lexicon, which can be a problem when the lexicon is not accurate.

Qiu, Liu, Bu, and Chen (2011) did a study to solve the problems in opinion extraction from the different features in the previous studies. Qiu et al. stated the problem as "opinion lexicon expansion and opinion target extraction." (p. 10) They came out with a new method, double propagation, that only required an initial opinion lexicon to extract the features and the opinion about the features. In their research there were many propagation rules defined based on the different relations they observed. The assumption of the "double propagation" method was that the feature was always a noun or nouns. Opinion words are always adjectives. They used the initial opinion word lexicon to find the features that the opinion was targeting. After that, they tried to find the opinion words based on the discovered features. It would end when there were no more opinion words or feature words. Then the authors used the different pruning skills to prune the opinion and the features.

There were some advantages of using this method. First, it did not require additional opinion words. The method started with an opinion word lexicon. It was an unsupervised approach that did not require too much manual labeling work. Another advantage was that this method would increase the recall when compared with the previous work.

However, there were some serious disadvantages. The precision would drop sharply when the corpus was large because it might extract a noun or nouns which were not features. The data they used to test was only about 600 sentences, a very small corpus for the training. The second drawback was that the method was rule-based. It would fail if there were some different relationships beyond the rules. Therefore, the approach did not work so well when the amount of data was large. More than that, it only analyzed and tested the English data. Most of the rules changed when switching the language.

Zhang, Liu, Lim and O’Brien-Strain (2010) tried to improve the double propagation method. In the previous study mentioned, the definition of double propagation technique was introduced. There were some problems of the double propagation. One of them was low precision when the corpora are large. To solve the problem, Zhang et al. conducted two improvements: part-whole patterns and “no” pattern were introduced to solve the problems.

Part-whole pattern means the "part of" relation. For instance, "car hood" has a part-whole relation, that is, the hood was part of the object "car." The authors explained the "no" pattern as the relation between the word "no" and the noun or nouns that followed "no". For instance, "no noise" was a "no" pattern. Liu et al. made a list of such patterns manually. In the ranking part, Liu et al. (2010) used HITS algorithm. Liu et al. stated, "Hyperlink-induced topic search (HITS) is a link analysis algorithm that rates Web pages" (p. 1467). In their HITS algorithm, the features only had an authority score, and the feature indicator only had hub scores. They used data sets from a commercial company. As a result, the precision of the new method was about a 5% improvement when compared with the double propagation.

The overall approach was rule-based. The test data size was about 30,000 sentences, which is a relatively small size data when compared with Yelp datasets. It may not work well when the corpora are very large. That was the biggest problem of this method.

2.3 Aspect Extraction Based on Topic Modeling

There were about two useful topic modeling approaches in these studies: Latent Semantic Indexing (LSI) and Latent Dirichlet Allocation (LDA). There were many studies based on topic modeling to extract the aspects and cluster the aspects (Chen & Liu, 2014; Y. Hu, Boyd-Graber, Satinoff, & Smith, 2014; Huang, Rogers, & Joo, 2014; McAuley & Leskovec, 2013; Sauper, Haghighi, & Barzilay, 2011). More than that, there are studies focused on extracting the local and global aspects (Titov & McDonald, 2008), summarizing aspects from comments (Lu, Zhai,& Sundaresan, 2009), and latent aspect extraction and rating (Moghaddam & Ester, 2011; H. Wang, Lu, & Zhai, 2010).

Hofmann (1999) described the core of Probabilistic Latent Semantic Indexing (LSI) model as “a statistical model which has been called an aspect model. The latter is a latent variable model for general co-occurrence data which associates an unobserved class variable” (p. 51).

Blei, Ng, and Jordan (2003) describe the latent Dirichlet allocation (LDA) as following:

> A generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document. (p. 1)

The latent Dirichlet model resembles unobserved "factors" in the data with the factors that were learned from certain topics or sentiment categories. After the comparison between the LSI and LDA, this author found that LDA solved some problems caused by LSI such as the overfitting problem, as described below.

McAuley and Leskovec (2013) reported on a study that aimed to reveal "latent rating dimensions (such as those of latent-factor recommender systems) with latent review topics (such as those learned by topic models like LDA)" (p. 165). They collected data from different public sources–35 million reviews from Amazon.com, 6 million beer and wine reviews, and 220,000 reviews from Yelp–and divided them into 26 parts based on categories. In the study, they created a "Hidden Factor as Topics" model based on the Latent Dirichlet Allocation (LDA) model. Their HFT model could "accurately fit the user and product parameters with only a few reviews, which existing models cannot do using only a few ratings" (McAuley and Leskovec, 2013, p. 172).

Huang, Rogers, and Joo (2014) performed a study that aimed to improve restaurants by extracting subtopics from Yelp reviews. In their study, they tried to describe latent subtopics by using the Latent Dirichlet Allocation (LDA) algorithm. They used data from the Yelp Dataset Challenge. In that dataset, there are about 5,000 restaurants and over 158,000 reviews.

In their study, they found about 50 subtopics. Huang et al. (2014) found that "some latent subtopics that were extracted from Yelp reviews include service, value, decor, and healthiness. Additionally, temporal topics such as breakfast, lunch, and dinner also came up in our findings and proved useful for peak hour observations" (p. 1). They also gave a comparison of the overall star rating with the subtopic rating. The overall hidden topic rating has a positive correlation with the overall star ratings of the restaurants. Huang et al. said, "Overall, it turned out that users care most about service, and subsequently value, take out, and decor" (p. 4).

This information gave the author an idea of how to find the relative aspects in the reviews of the restaurant. This study talks about the useful subtopics. However, the comparison and prediction part was vague. The researcher would improve that part.

Sauper, Haghighi, and Barzilay (2011) conducted a study about review snippets. The goal of their study was to provide a model for review content aggregation that will evaluate the product beyond numerical scores. Sauper et al. stated that "they are interested in identifying fine-grained product properties across reviews (e.g., battery life for electronics or pizza for restaurants) as well as capturing attributes of these properties, namely aggregate user sentiment" (p. 350). They used Yelp reviews generated by their previous system as their data set, and they finished a probabilistic topic model to identify and evaluate the product review snippets. They set several baselines to examine their model. One of the baselines was the maximum entropy discriminative classifier over the unigram. As a result, their model improved the accuracy of identifying from 67% to 74% correction.

2.4 Aspect Extraction Based on Machine Learning

There were some studies based on Machine Learning technique. Some of the studies used supervised machine learning combined with dependency and part-of-speech rules (Zhuang, Jing, & Zhu, 2006). Some of the aspect extraction algorithms used the conditional random fields to predict the opinion target (Choi, Cardie, Riloff, & Patwardhan, 2005; Jakob & Gurevych, 2010; Toh & Wang, 2014). However, there are some problems about the conditional random fields (CRF) and other supervised machine learning algorithm. CRF is a linear model, which means it requires large numbers of the data to perform well. These models demand massive labeled data for training to work well. Some semi-supervised models based on the seeding aspects (T. Wang et al., 2014) and seeding aspect words from a lexicon (Jagarlamudi, Daum´e III, & Udupa, 2012) were introduced. In the following paragraph, several aspect extraction models based on machine learning were introduced.

Mukherjee and Liu (2012) presented a new semi-supervised model to extract the aspect words. They conducted two models. One was Seeded Aspect and Sentiment model (SAS). The other one was the model based on the maximum entropy and DF-LDA model (Andrzejewski, Zhu, & Craven, 2009). The DF-LDA model added cannot-link and must-link constraints to the standard LDA model. By training the Maximum entropy, they did not need manually labeled data.

The model was related to the Latent Dirichlet Allocation (LDA) model and had the problem that it would also extract non-specific features. In their study, Mukherjee and Liu (2012) stated: "We employ seed sets to address this issue by guiding the model to group semantically related terms in the same aspect thus making the aspect more specific and related to the seeds." (p. 341)

This research provided a new approach based on seed selection. It was a different type of the topic model with the seed to extract related features. It was useful if users were familiar with what reviews were about, and if the users could select meaningful seeds. However, if users are not familiar with the reviews, this approach may not work very well.

Deep learning approaches became state-of-the-art in different areas like Natural Language Processing, Computer Vision and so on. Poria, Cambria, and Gelbukh (2016) conducted a study of aspect extraction based on the convolutional neural network. The authors tried to detect the aspect words or non-aspect words in the reviews.

In their study, a 7-layer deep convolutional neural network was built to tag the words in the review. They used the word embedding technique to present each word into a 300-dimensional vector. In their study, they used the Google word embedding and train, a word-embedding based on Amazon reviews. Additionally, they created five rules based on the Part-of-Speech Tags as the features.

There were two datasets. One was provided by Hu (2004). The dataset used to test their model was relatively small and manually labeled. They used another dataset from Amazon to train their model. Their model improved the Popescu and Etzioni (2005) approach by 5% to 10%. They also compared the model by using different features. One used the word embedding as features, and another one used the word embedding feature and the POS features. The test results showed that the second version of features did improve the performance. Poria et al. (2016) concluded: "while the word embedding features are most useful, the POS feature also plays a significant role in aspect extraction" (p. 46).

This study represented the performance of state-of-the-art approaches on aspect extraction. The result showed that both Part-of-Speech Tags and word embedding helped the convolutional neural network to select aspect words.

2.5 Vector Representations of Words

Vector representation of words, same as “word embedding,” tried to represent words into continuous vector space where semantically similar words are nearby each other. There were lots of word embedding models in the history of natural language processing (NLP). Most of them are based on Distributional Hypothesis, which means words that are used or occur in the same contexts are likely to have similar meanings (Harris, 1954). In another way to explain it, a word is characterized by the other words that are around it (Firth, 1957). The approaches could be divided roughly into two parts. One part is count-based approaches and were described in the previous part of the article. For instance, Latent Semantic Analysis (Hofmann, 1999) is one of the count-based approaches. These approaches count how often some words co-occur with nearby words in a vast corpus. They then assign the count-statistics to each word to create the vectors of the words. The other approach used was predictive models, which tried to predict the words from the neighbor words to assign values to the vectors of the words. The state-of-art predictive vector representation model is Word2vec model (Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Mikolov, Yih, & Zweig, 2013).

Mikolov, Chen, et al. (2013) created two different neural network models for vector representations of words. One of the models was called Continuous Bag-of-Words model (CBOW). Another model was called Skip-Gram model. The combination of these models is called a Word2vec model. The CBOW model tries to predict the central words by using context words or neighbor words. The Skip-Gram model predicts the surrounding words by the given words. The results showed the model achieved an excellent performance. For instance, “word big is similar to bigger in the same sense that small is similar to smaller" (Mikolov, Chen, et al., 2013, p. 5). To find the vector of "small", the Word2vec model could simply compute it by using vector(small) = vector(biggest) − vector(big) + vector(small). More than that, Mikolov, Chen, et al. (2013) stated that "the resulting vectors can be used to answer very subtle semantic relationships between words, such as a city and the country it belongs" (p. 2).

Baroni, Dinu, and Kruszewski (2014) conducted a study to compare the count-based approaches and the predictive models. They compared several count-based models like Non-Negative Matrix Factorization model (Hoyer, 2004), LDA model (Blei, Ng, and Jordan, 2003) and other models with the predictive model: Word2vec (Mikolov, Sutskever, et al., 2013). As a conclusion, the predictive model defeated the count-based model. They stated that "the predict models are so good that, while the triumphalist overtones still sound excessive, there are very good reasons to switch to the new architecture" (p. 239).

These studies gave the author a clear idea that the predictive model of word embedding outperformed the count-based word embedding model. In that case, the predictive model for representing the word to the vector was used in the study.

2.6 Clustering Technique and Comparison

Documents clustering means to cluster different documents into groups, such as clustering different news into different categories. The common approaches converted documents into the vectors, and cluster the document based on the vectors. There were lots of studies that analyzed the clustering algorithm (Guha, Rastogi, & Shim, 1999; Jain & Dubes, 1988; Kowalski, 1998). There were two kinds of common clustering techniques: hierarchical clustering and *k*-means. In the hierarchical clustering technique, three commonly used techniques were discussed. They were Intra-Cluster Similarity Technique (Grabmeier & Rudolph, 2002), Centroid Similarity Technique and UPGMA (Jain & Dubes, 1988).

UPGMA: Jain and Dubes (1988) defined the similarity between two clusters as following formula.

$$\mathrm{similarity}(\mathrm{cluster1}, \mathrm{cluster2}) = \frac{\sum \mathrm{cosine}(\mathrm{d}_1 + \mathrm{d}_2)}{\mathrm{size}(\mathrm{cluster1}) * \mathrm{size}(\mathrm{cluster2})}$$

The Centroid Similarity Technique: The technique defined the similarity between two clusters as the cosine similarity between the centroids of two clusters (Steinbach, Karypis, Kumar, et al., 2000).

For the *k*-means technique, there are standard *k*-means and bisecting *k*-means (Hartigan & Wong, 1979; Steinbach et al., 2000). Steinbach et al. (2000) explained how the bisecting *k*-means work as following steps:

1 Pick a cluster to split.
2 Find 2 sub-clusters using the basic *k*-means algorithm.
3 Repeat step 2, the bisecting step, for a fixed number of times and take the split that produces the clustering with the highest overall similarity.
4 Repeat steps 1, 2 and 3 until the desired number of clusters is reached. (p. 2)

In 2000, Steinbach et al. compared three hierarchical clustering techniques mentioned in the previous paragraph and two k-Means techniques (standard *k*-means and bisecting *k*-means). The result showed that "bisecting *k*-means technique was better than the standard *k*-means approach and as good the hierarchical approaches" (Steinbach et al., 2000, p. 2).

This study gave the author a comparison of commonly used clustering algorithms. Because the *k*-means are linear algorithm but hierarchical clustering are not (Larsen & Aone, 1999), the running time of *k*-means algorithm is less than hierarchical clustering when the vector dimensions are relatively large. However, these studies did the experiments on small specific text mining tasks. When the author tried to implement the clustering algorithm, the runtime of the hierarchical approaches was better than the bisecting algorithm. In that case, the hierarchical cluster approaches were used.

CHAPTER 3. FRAMEWORK AND METHODOLOGY

3.1 Quantitative Framework

3.1.1 Proposed Approach

The research problem was, “whether it is possible to extract words or terms that reflect restaurant aspects from Yelp reviews by using Word2vec model?”

To extract the aspect words in reviews by using Word2vec model, the problem was split into four sub-tasks. The first sub-task was identifying and defining the grammatical tagging of the aspect terms. The second was finding similar aspect words in a large corpus. The third sub-task was clustering different words. The final task was testing the performance of the methodology. The steps to solve these sub-tasks are represented in the following paragraph.

3.1.1.1 Defining the Grammatical Tagging

In previous work, the aspect terms were marked as nouns, either singular or plural (M. Hu & Liu, 2004; Lerman, Blair-Goldensohn, & McDonald, 2009; Popescu & Etzioni, 2007). The research focused on noun words to find aspect words. To define the noun words in the reviews, the Part-of-Speech technique was used. Moreover, in the restaurant's review, there were some food names that contained noun chains. For example, "Moscow mules" is a food name which has two nouns. I hypothesized that treating noun chains as a single entity would result in better performance.

The following three steps were used to mark noun chains as single entity:

1. Changed case of the text and uncapitalized each letter in the reviews. In most cases, the program treated lower case words and upper case words as different tokens. To avoid the confusion, the first step was to make all words lower case.
2. Used Stanford Parser to tag each word in the reviews as noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction, interjection, and so on.
3. Selected the first word in the sentence. If the word was a noun, and the word on the left or the right of the chosen word was a noun, then an underscore was used to

connect the two words and label the new word as a noun. If not, the process moved on to the next word and repeated step 3. For example, “They also had Stella Cider and Moscow mules!” was one review in the dataset. After using Stanford Parser to tag the speech, the “Stella Cider” and “Moscow mules” were identified as the nouns. After finishing step 3, the sentence was, “They also had Stella_Cider and Moscow_mules!” Also, Stella_Cider and Moscow_mules were the words that were tagged as nouns.

3.1.1.2 Extracting Aspect Words in a Large Corpus

The second sub-task was finding similar aspect words in a large corpus. In the previous chapter, many papers relevant to this sub-task were introduced. They could be divided into three parts: rule-based approaches, lexicon-based, and vector-based models (or word embedding models). The details, pros, and cons were discussed in the previous section. When the corpus is relatively small, the rule-based and lexicon-based models have fairly good performance. When the corpus is large, the rule-based or lexicon model performance may suffer. The models based on predictive vector model became state-of-art and had better performance when the corpus was large (Baroni et al., 2014). In that case, the Word2vec model that represented words into a vector approach should be used. The following step was used to extract concept words for the large corpus:

4. Used Python package gensim to train skip-gram and CBOW models (Word2vec model) by using reviews as the training set. Then assigned each word a vector as values.

3.1.1.3 Clustering

The third sub-task was to cluster different words. Two approaches were used for clustering: a hierarchical approach and a standard *k*-means approach. These approaches were chosen based on studies that analyzed different clustering techniques (Steinbach et al., 2000)

The following steps were chosen to compare the approaches:

5. Selected singular or plural nouns.

6. Clustered the words into different groups by using cluster algorithm. To achieve that, a Python package, sklearn, was used.
7. In the cluster, sorted the noun or nouns by frequency.
8. Manually selected which clusters remained and which did not.

Because of the large amount of the data, it took a long time to cluster. For that reason an efficient algorithm would have been more suitable for the task. The author compared a hierarchical approach and *k*-means algorithm based on the efficiency. The results showed that the hierarchical approach performed better than standard *k*-means based on running time. Thus, the hierarchical approach was chosen for clustering.

3.1.1.4 Testing Performance

The last sub-task was testing the performance of the methodology. The following steps were followed:

9. To test the executed model, 1,076 manually labeled reviews were used. For each of the words in the review, we determined if the word was or was not an aspect word by using the manually selected clusters of words.
10. Used the result to calculate the precision, recall, and F-1 score.

Most of the aspect word extraction approaches, which were discussed in the literature chapter, required labeled data to train the model as they used supervised ML. However, in many cases labeled data is not available. Thus, to be consistent with real world availability and expectations, this research used an unsupervised model to represent the words into a vector. Also, it included the clustering technique and POS technique. Such an approach is useful if the dataset is large and labeled data is not available. The methodology was aimed to use the state-of-art word embedding technology, Word2vec, to extract aspects words and test the performance.

3.1.2 Baseline Model

The baseline model was LDA model (Blei et al., 2003) that was discussed in section 2.4. As the section describes, many semi-supervised models are based on LDA model or DF-LDA model. There were several reasons to select LDA model as a baseline. First, both Word2vec model and LDA model are unsupervised. It was reasonable to compare different unsupervised models. Second, the LDA model was a count-based model while the Word2vec model is a predictive model. From the study conducted by Baroni et al. (2014), the predictive model may outperform the count-based model, and it was reasonable to test whether it is the case here.

The LDA model required changes to Steps 4-6 described above:

Step 4: Used Python package gensim to train LDA model on the reviews. The words were assigned to a 200-dimensional vector.

Step 5: Clustered the words into different groups by using cluster algorithm.

Step 6: In the clustering of the different topic, manually decided which cluster remained and which did not.

Step 7: Used the test data to calculate the precision, recall, and F-1 score.

3.1.3 Programming Method

The program was based on Python. The reason for using Python is the useful Python packages. The following packages were employed: numpy, gensim, nltk, tensorflow, and sklearn.

3.2 Data

3.2.1 Data Sources

The study used the Yelp Academic Dataset. In general, the data set was divided into two sets. First was the training set. The datasets were used for training the Word2vec model. The second one was the test dataset. The test dataset was manually labeled by four people.

3.2.2 Data Collection Procedures

The data was provided by Yelp. It came from the Yelp Dataset Challenge ("Yelp Dataset Challenge", 2016)

.

3.2.3 Data Analysis Strategy and Procedure

The original dataset included about 2 million reviews and 591,000 tips by 552,000 users for 77,000 businesses. Also, it included 566,000 business attributes, e.g., hours, parking availability, ambiance. In this study, the data about the restaurants were used. The dataset included 11 cities across four countries. The reviews of restaurants from the cities in Canada and American were used for study. The final data contained about 1.7 million reviews.

3.2.4 Test Data Summary

3.2.4.1 Sample Size

In the project, there were 1,746,271 reviews about restaurants in the dataset. The author chose 1,076 random reviews from the dataset as the test data.

To determine the sample size, three concepts were important. First was the population size. In the project, the population size was 1,746,271 reviews. The second was the margin of error which means how closely the answers from the sample are to the real population value. The third concept was confidence level which means how much certainty there is that the sample reflects the population. The sample size formula was represented as follows:

$$Sample\ Size = \frac{\frac{z^2 \times p \times (1-p)}{e^2}}{1 + \frac{z^2 \times p(1-p)}{e^2 \times N}}$$

In the formula, *z* means the z-score. When the confidence level is 90%, the z-score is 1.65. The margin of error is the *e* parameter in this formula. Population size is the *N* parameter in the formula. The author decided to use a sample size with a margin of error of 2.5% and a confidence level of 90%. After calculation the sample size is 1,076. It means the author is 90% confident that the sample reflects the population with the margin of error of 2.5%.

3.2.4.2 Sample Data Production

In the beginning, the program loaded the original dataset from the Yelp Challenge Dataset, cleaned the dataset, and got the reviews that met the standard explained previously. The author used the program to randomly select 1,076 samples from the cleaned dataset. Four people manually labeled the data. The author labeled all the sample data. Then, the sample data was divided into three parts. Three other people labeled one part of them again. The method was chosen to reduce any potential bias. The author told three people to select the noun words including the food name, food portion, and food quality. Then the author told them to select the words that reflected the service. These words included the staff and service. Further, the author told them to select the words that reflected the price. For example, words like "price," "fee," and so on. In the end, the noun words that reflected the atmosphere were selected.

The words labeled by at least one person were selected for final processing. In the end, the author combined the sample data where the aspect words were detected and labeled.

3.3 Researcher Bias

At least three biases are known in this study. The first one was the limitation of test data. The data was manually labeled by friends of the author. The subjectivity could affect the results of the testing procedure. To avoid parts of individual subjectivity, three people labeled the first version of the test data. The author labeled the data again, and combined the two versions of the test data together.

The second bias was the accuracy of the part of speech tagging system. The author used the Stanford POS tagging tool to tag words in the sentence. However, it was not perfect. The author combined the Stanford POS tagging tool with a tagging tool from the nltk Python package. When both tagging tools agreed the word was noun, the noun word remained. The version of nltk Python package was 3.2.2. The version of Stanford POS tagging tool was 3.7.0.

The third bias was the process of selecting the relative clusters. The author manually selected the relative clusters. The subjectivity of the author affected the test results.

CHAPTER 4. SYSTEM OVERVIEW AND DESIGN

4.1 System Overview

Figure 4.1 showed the overview of the entire system. The aim of the system was to test how a Word2vec model works to extract concept words that reflect different aspects of the restaurants. The typical four aspects are ambiance, price, food, and services (Sajnani, et al., 2012).

The outcome of the system was a dictionary that contained the aspect words mentioned before. Then, the system used the test dataset to evaluate how the Word2vec model performed and then analyzed why the Word2vec model worked or not. In this section, five major parts of the system were introduced.

The system consisted of following seven essential parts:

1. Load and clean the data.
2. Train the Word2vec model. (Mikolov, Chen, et al., 2013)
3. Cluster the words based on the vector of the word.
4. Select useful cluster words and create the final dictionary.
5. Represent the word in vectors by using a baseline model.
6. Cluster words and selected relevant clusters by using a baseline model.
7. Test the results and analyze the results.

The project was written in Python, version 3.6.0. In the project, several packages were used: nltk, gensim, tensorflow, pickle, numpy, pycluster, re, xlrd, and scipy packages.

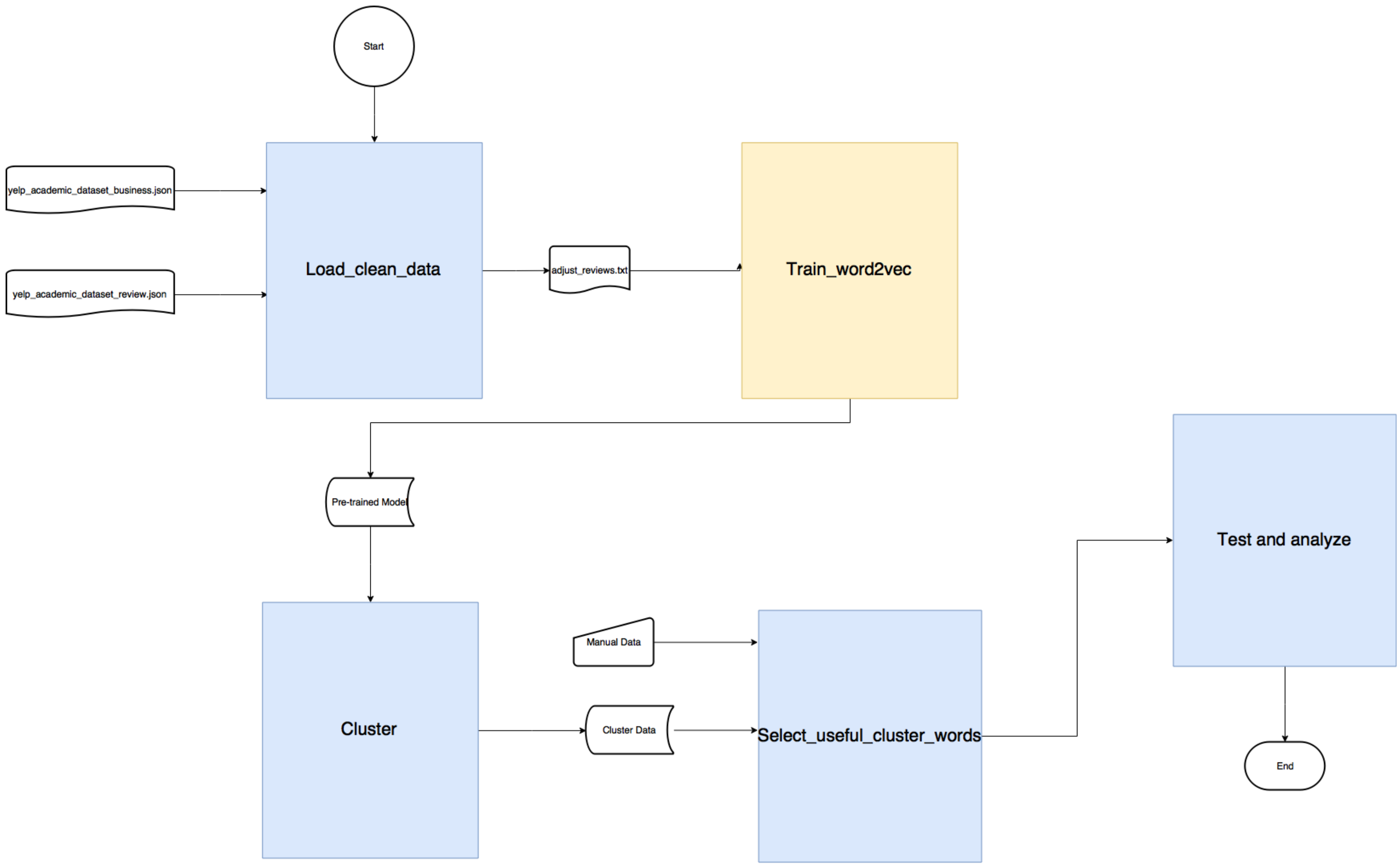

Figure 4.1 Basic System Diagram

4.2 Load and Clean the Data

One of the most important parts of the system was to clean and load the data. The Figure 4.2 represents the flowchart of the cleaning and loading data process of the system. This part of the system could be divided into four subparts.

The input data are from two files. First dataset file was yelp_academic_dataset_business.json. The data set contained the business ID, business name, categories, and other features of the business. Another file was yelp_academic_dataset_review.json. This file included the reviews of the different business. The system combined the data from two datasets and cleaned them for further analysis.

The first subpart was to create a dictionary that contained the business data that would be used in other functions. In this Python dictionary, the key value was the 'business_id', and the value was a list that contained the business categories, business name, business city, and review counts of the business. The function read every line from the two files and extracted the useful information. After the extraction process, the Python dictionary would be used as input for other functions.

The second subpart was to load relative reviews. The research focused on the reviews from Canada and the United States of America, and the system chose to load reviews from these countries. Also, it replaced the restaurant's name with the string "RESTAURANTS_NAME".

The third subpart was to check whether the sentence was in English and if it needed to adjust sentences. In the dataset, there were some Spanish and French reviews. In those instances, the system would configure which ones were the English reviews and which were not. Also in the reviews, there were several continued noun or nouns. The system found them and connected them with an underscore. In this subpart of the program, the nltk packages were used to help the author identify the English reviews.

The final subpart was to create a file, adjust_review.txt. The file contained cleaned reviews for other parts of the system. In this subpart of the system, the short reviews were deleted. Then the system wrote reviews into a text file for further use.

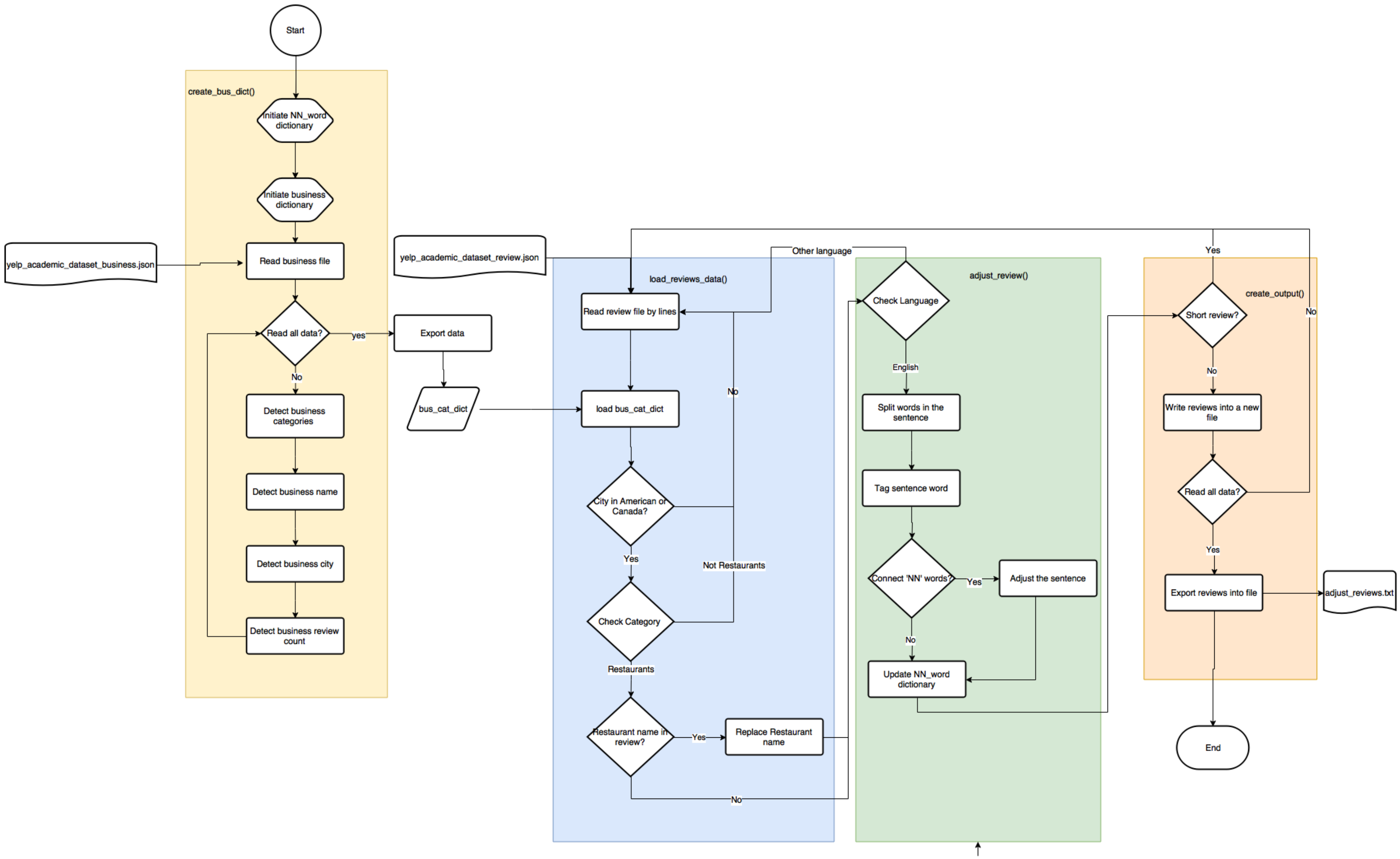

Figure 4.2 Flowchart diagram of loading and cleaning dat

4.3 Train the Word2vec Model

Figure 4.3 shows the flowchart diagram of converting words to vectors using the Word2vec model. First, the system loaded the cleaned data. Then the system loaded the each sentence for training. In the experiment, if the program tried to load too many reviews at once, the computer would run out of memory. In those cases, the program loaded each review as an input to train the Word2vec model.

The next step was to initiate a vector with the random small real number and determine how many dimensions of the vector there would be. After the vector initialization step, a count number function was used to delete the words with low frequency with the threshold set to 3. This step greatly increased the speed. It saved about half of the time when compared with training on every word. In retrospect, this step was necessary.

The final step was to save the trained model for future use. In the system, the pickle package was used to save the model. In the next section, the saved model was used as the pre-trained model. It was easy for reuse.

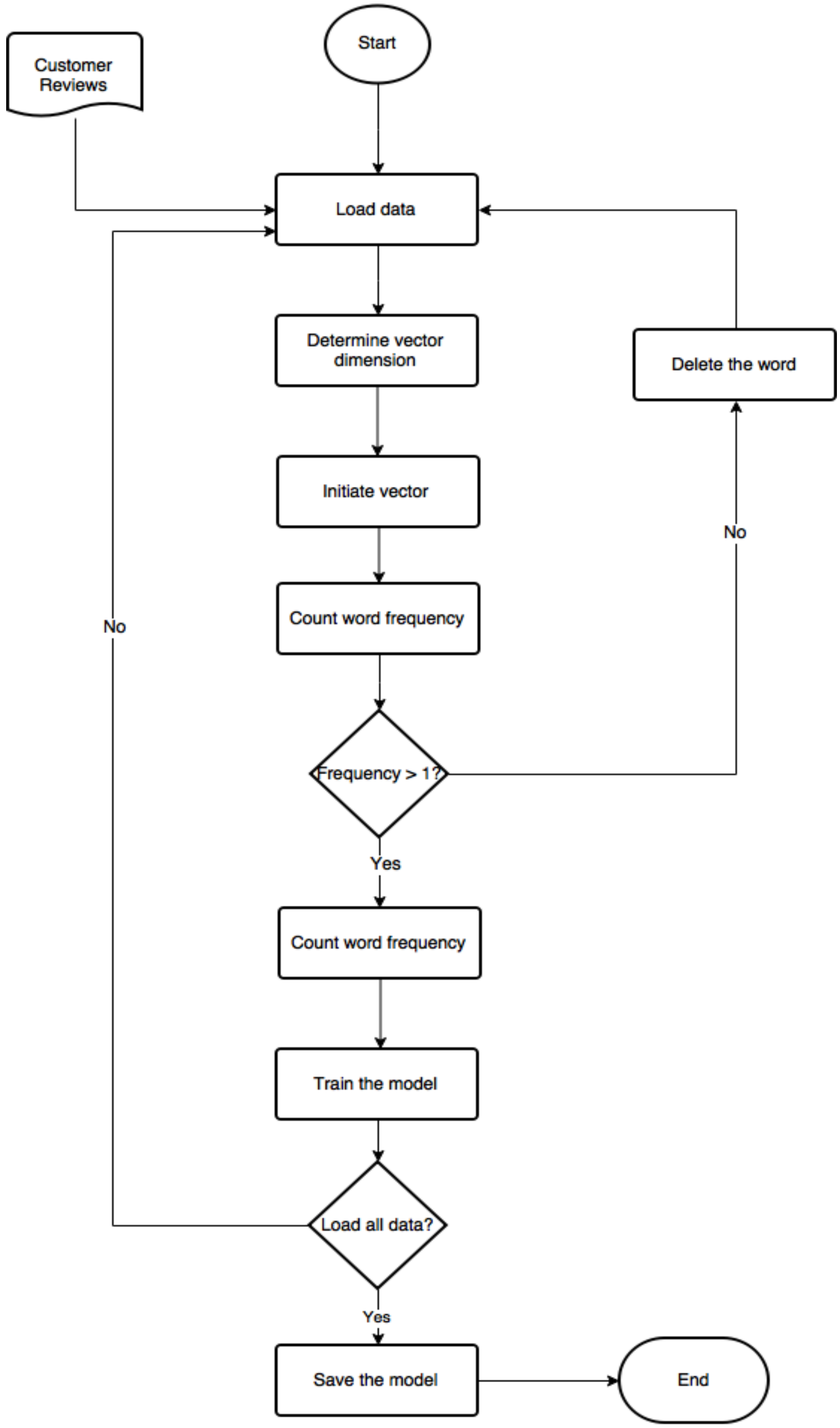

Figure 4.3 Flowchart diagram of training the word2vec model

4.4 Create Cluster of the Words

The aim of this part of the system was to group different words into different clusters by comparing the similarity of the vectors. The inputs were the pre-trained model that were created in the previous part of the system, and the Python dictionary that contained noun or nouns which was created in the clean and load data part. The output of this part of the system was a Python dictionary that contained noun words, the vector value of the word, the frequency number of appearance and to which cluster it belonged.

In Figure 4.4, the flowchart diagram of creating clusters of the words part is represented. First, the pre-trained model and a Python dictionary of nouns were imported. Then a Python dictionary, NN_w2v_cont, was initiated. After initialization, the system loaded the words from the previously created noun Python dictionary and vector into the NN_w2v_cont dictionary. It calculated the word frequency and imported them into the Python dictionary. The key of the NN_w2v_cont was the noun or noun words. The value of that key was another Python dictionary with three keys: vector_value, freq_num, and cluster_num.

After executing the first function of this part, the function cluster_w2v() grouped the words into different clusters and assigned the value into the NN_w2v_cont Python dictionary. First, the function detected the number of dimensions of each vector. Then, the function decided what the maximum number of the clusters was. Further, a two-dimension numpy array was initialized with zeros. After the initialization, the program assigned the value of the vector of each word into the two-dimensional array. The row number of the array was the total number of the noun words, and the column number was the dimension of each vector. After that, the cluster method was implemented.

After the clustering, the program assigned the word cluster number into the NN_w2v_cont Python dictionary and saved it to a file by using a pickle package.

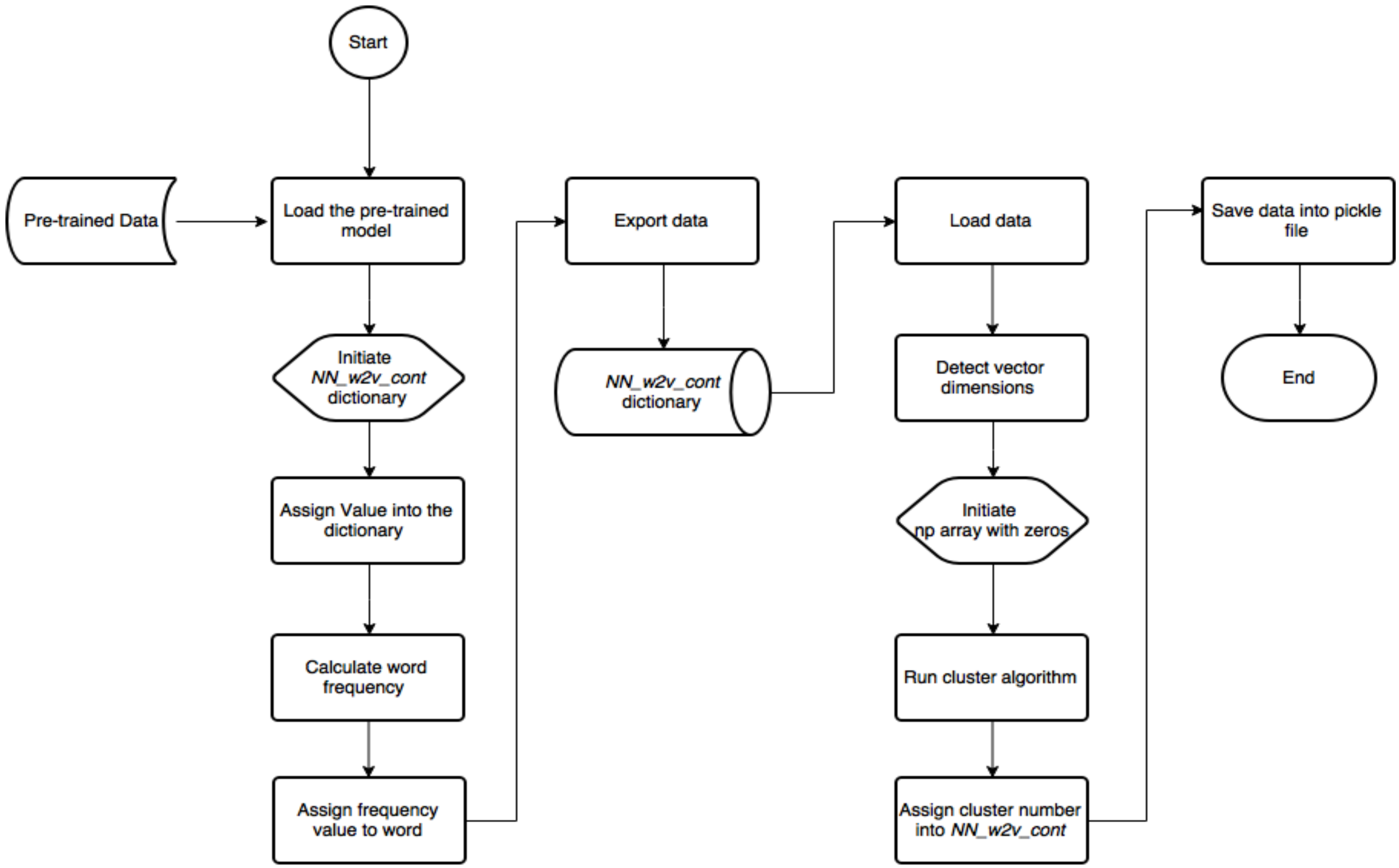

Figure 4.4 Flowchart diagram of creating cluster of word

4.5 Select the Relevant Cluster from Word2vec Model

In Figure 4.5, the flowchart diagram represents the process of selecting the relevant clusters that contained the aspect words from the Word2vec model.

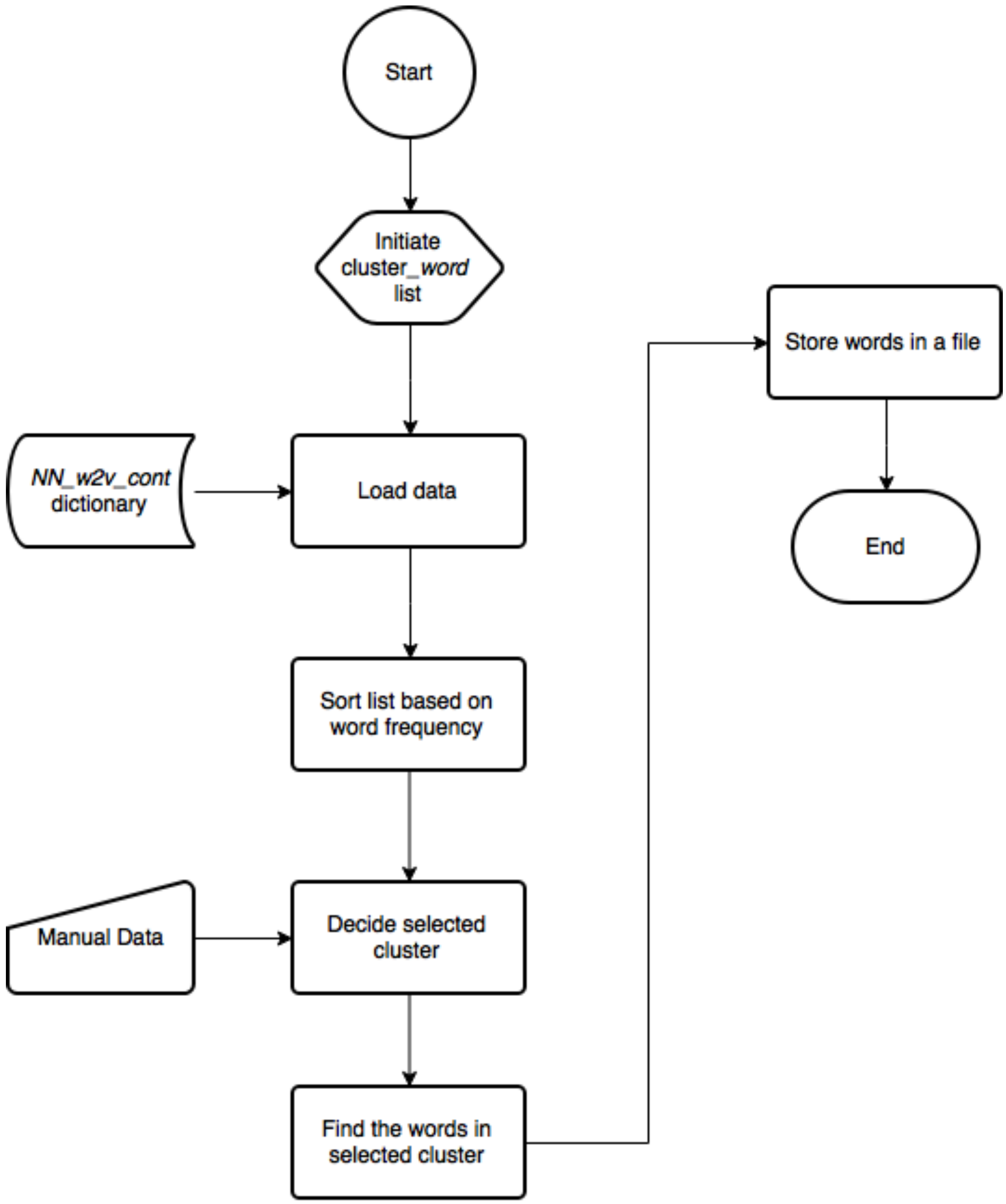

Figure 4.5 Flowchart diagram of selecting relevant cluster from word2vec

In this section of the system, the input was the NN_w2v_cont Python dictionary that was previously stored. After it imported data, the system created a two-dimensional list to store the data. The row number was the number of the cluster. Each row contained the words in the specific cluster. Then, it sorted words in the clusters by word frequency. Further, the system created a file for selection. After the preparation, the author needed to manually select and decide which cluster contained the words that reflected the different aspects of the restaurants. The selection process was followed using several steps:

1. For each cluster, the program sorted words in the cluster by the word frequencies.
2. The author selected the top 20 words with higher word frequencies. If a cluster had less than 20 words, then all the words in that cluster were selected.
3. For the each word, the author checked the words with each criterion.
    3.1 The author checked whether the word was a noun or not. If it was not, it was marked with value "0" and then continued to the next word. If the word was a noun, then the author would continue to check the next several criteria.
    3.2 The author checked whether the word was a food or not. If it was food name or food quality, then the word was marked as value "1". If it was not, the author continued to check the words with the next criterion.
    3.3 The author checked whether the word reflected the food portion or food quality. If it was, then the word was marked as value "1". If it was not, the author continued to check the words with next criterion.
    3.4 The author checked whether the word reflected a person in the restaurants. For instance, the word "staff" was one of the words. If the word reflected the person in the restaurants, the word was marked as "1". Otherwise, the author continued checking the words with next standard.
    3.5 The author checked whether the word was related to the ambience or atmosphere. If it was, then it was marked as "1". Otherwise, the author would continue to check.
    3.6 The author checked whether the word was reflecting price or fees. If it was, the word was marked as "1". If not, the author checked it with next standard.
    3.7 The author checked whether the word was related to service or not. If it was, then it was marked as "1". Otherwise, the word was marked as "0".

4. After checking the top 20 words and assigning the value, the author added the value together and divided it by 20. If the rate was larger than 60%, then the cluster was selected.
5. Author repeatedly checked each cluster.

4.6 Represent the Word in Vectors by Using Baseline Model

In Figure 4.6, the flowchart diagram represents the process of training the baseline model and transforming the word into vectors.

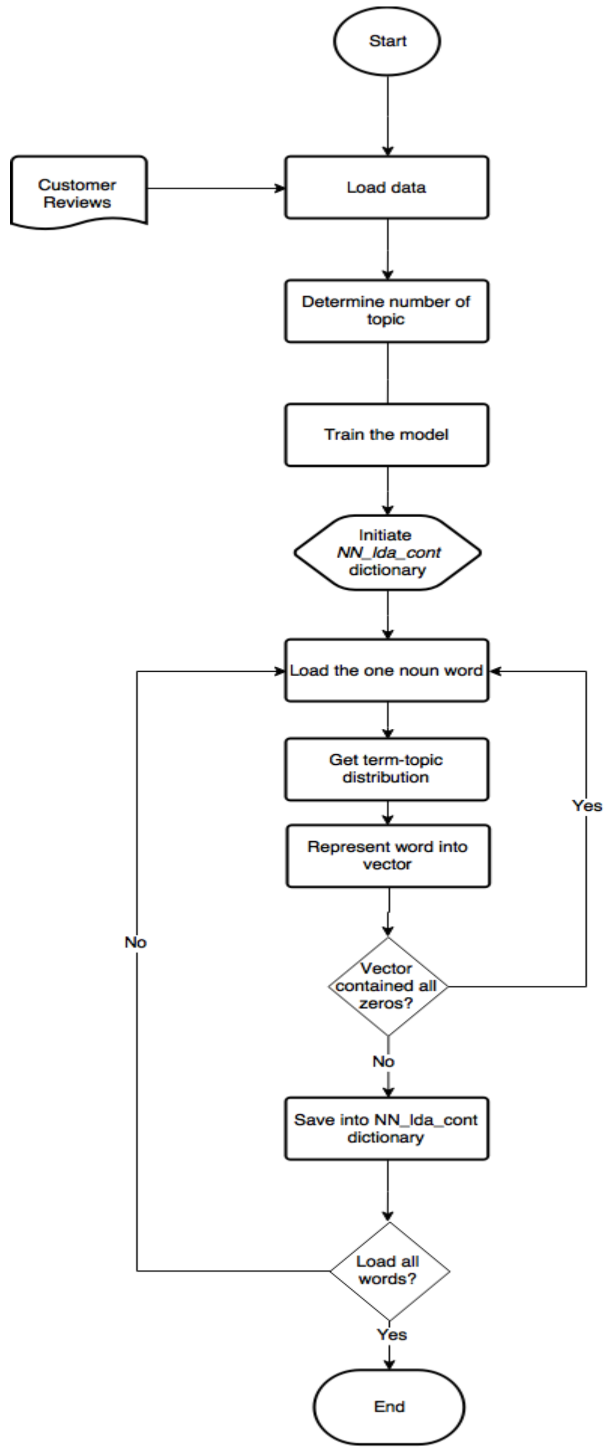

Figure 4.6 Flowchart diagram of representing words into vectors (LDA)

In this of the program, the input was the customer reviews. After the author decided the number of the topic, the program started to train the LDA model. Then the program initialized NN_lda_cont Python dictionary. The system calculated the term topic distribution by using get_term_topic function. Then the system initialized a 200 numpy array that contained zeros and assigned the value into the vector based on the term-topic distribution that was calculated previously. For instance, the word “burger” had 0.2 probability that it was related to the topic 2. The system changed the third position of zeros vector from value 0 to 0.2. Then the system checked whether the vector contained 200 zeros or not. If it contained 200 zeros, then the system deleted it and checked the next words. If it didn’t contain 200 zeroes, then the vector was saved into the NN_lda_count Python dictionary.

4.7 Cluster Words and Select Relevant Clusters by Using Baseline Model

In Figure 4.7, the flowchart diagram represents the process of clustering the words and selecting relevant clusters.

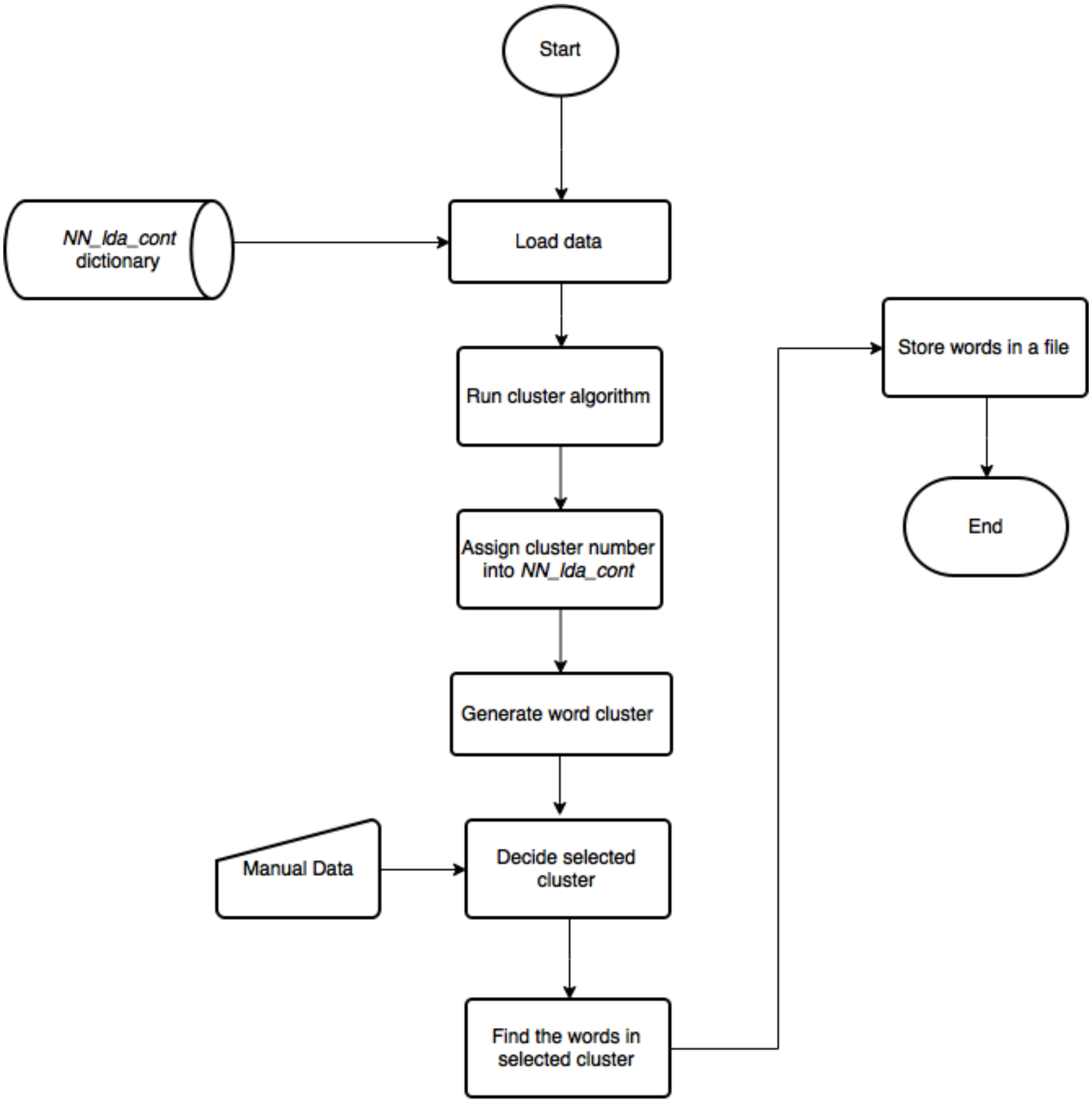

Figure 4.7 Flowchart diagram of selecting relevant cluster by LDA model

In this section of the system, the input was the NN_lda_cont dictionary conducted in the previous section. After loading the data, the system ran the clustering algorithm and assigned the cluster value into the NN_lda_count Python dictionary. Then the system generated represented the clusters that contained words. Further, the author manually selected and decided which cluster contained the words that reflected the different aspects of the restaurants. The selection process followed the criteria described in section 4.5.

4.8 Test the Results and Analyze the Results

To test and analyze the how the Word2vec model worked and why it worked or not, there were two parts to the system. The first part was to demonstrate features of the Word2vec model and analyze them. In this part, the tensorflow package was used. This part mapped words into 3D or 2D pictures by using the t-Distributed Stochastic Neighbor Embedding technique (Maaten & Hinton, 2008). The most similar words to the target word were found by using the cosine distance and Euclidean distance.

The second part was to calculate the precision, recall, and F-1 score. At first, the final Python dictionary needed to be imported. Then, an empty Python dictionary was created to store the labeled words and the words that the system found. Moreover, the system used the aspect nouns Python dictionary that was created in previous part of the system to detect the words in the test reviews. After finishing the discovering word jobs, the program saved the data into the previous Python dictionary. The final step was to calculate the precision, recall, and F-1 score.

CHAPTER 5. RESEARCH DATA AND RESULTS

5.1 Introduction

The results and analysis are represented in this section. There are two parts of the section. The first part analyzed the results of the trained Word2vec model. Words were separated into four aspects of the restaurants: food, service, ambience, and price. The relationships among different words were represented and analyzed. The second part tested the Word2vec model base on the test data. The precision, recall, and F-1 score were represented. It compared the results to the baseline model.

5.2 Analyze Word2vec Model Result

Before analyzing the Word2vec model based on the test data, how the Word2vec model works and what kinds of words were in the same cluster should be discussed and analyzed. In this section, the performance of the Word2vec was examined and analyzed based on the different words. This section is divided into four subsections to analyze the performance of the Word2vec model on different aspects of the restaurants. As discussed previously, the research mainly focused on four aspects: food, service, price, and ambience.

To visualize different word vectors into two- or three-dimensional space, a technique called t-Distributed Stochastic Neighbor Embedding was introduced. Maaten and Hinton (2008) represented t-Distributed Stochastic Neighbor Embedding as a technique to “visualize high-dimensional data by giving each data point a location in a two or three-dimensional map" (p. 2). In this research, the t-SNE technique was used to visualize different words based on the trained Word2vec model. The similar words would be close together in the visualization figures.

To give more details of the word2vec model, the author calculated the similarity among words and represented the most similar words. These results showed the performance of the Word2vec model to find similar words. One of the keys to test whether the Word2vec model worked or not was that the alike words should have been clustered together. In this case, discussing and analyzing the similarity among the words was necessary for this research. Two methods were used in the study to calculate the similarity of the high dimensional vector: cosine distance and Euclidean distance.

The first one was the cosine distance. For the vector A and B, the cosine distance formula was as follows:

$$D_c(A,B) = 1 - S_c(A,B)$$

Where

$$\text{similarity} = S_c = \frac{A * B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

For the vector A and B, the Euclidean distance formula is defined as follows:

$$d(A,B) = \sqrt{(A_1 - B_1)^2 + (A_2 - B_2)^2 + (A_3 - B_3)^2 + \cdots + (A_n - B_n)^2}$$

These two methods were used to find top ten similar words. The results showed the performance of the word2vec model to cluster the similar words together.

5.2.1 Food words

Before the implementation of the Word2vec model, the author thought the Word2vec model might cluster the food names together, but might not represent what category the food belongs. However, the results showed that Word2vec model worked better than author's expectation. For instance, the Word2vec model considered the word "burger" had the same meaning as the word "hamburger" but did not have the same meaning of Asian food like "ramen". The following paragraph presents these characters of the Word2vec model.

5.2.1.1 Asian Food

In Figure 5.1 and Figure 5.2 represents the words with the frequency greater than 100. The words that have the similar meaning of the word “ramen” were highlighted.

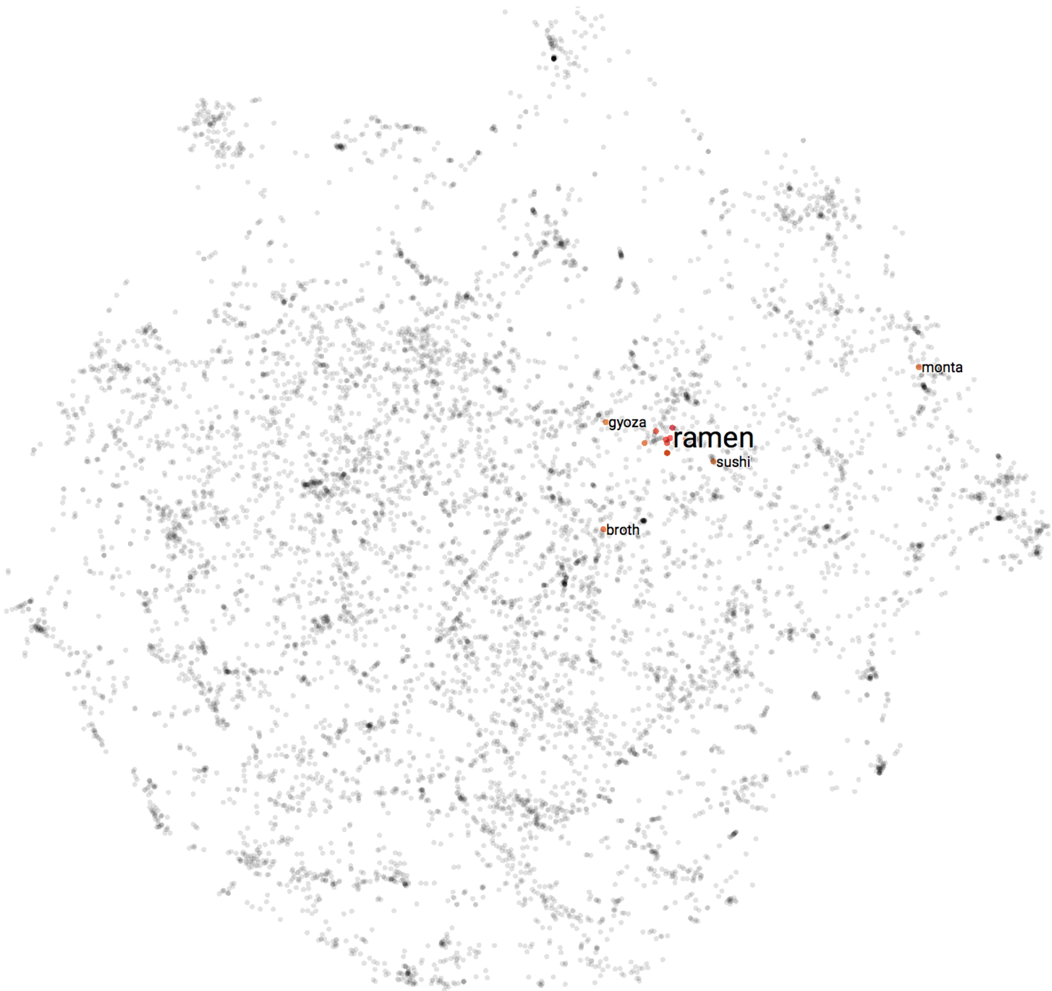

Figure 5.1 Similar words to “ramen” in 2D

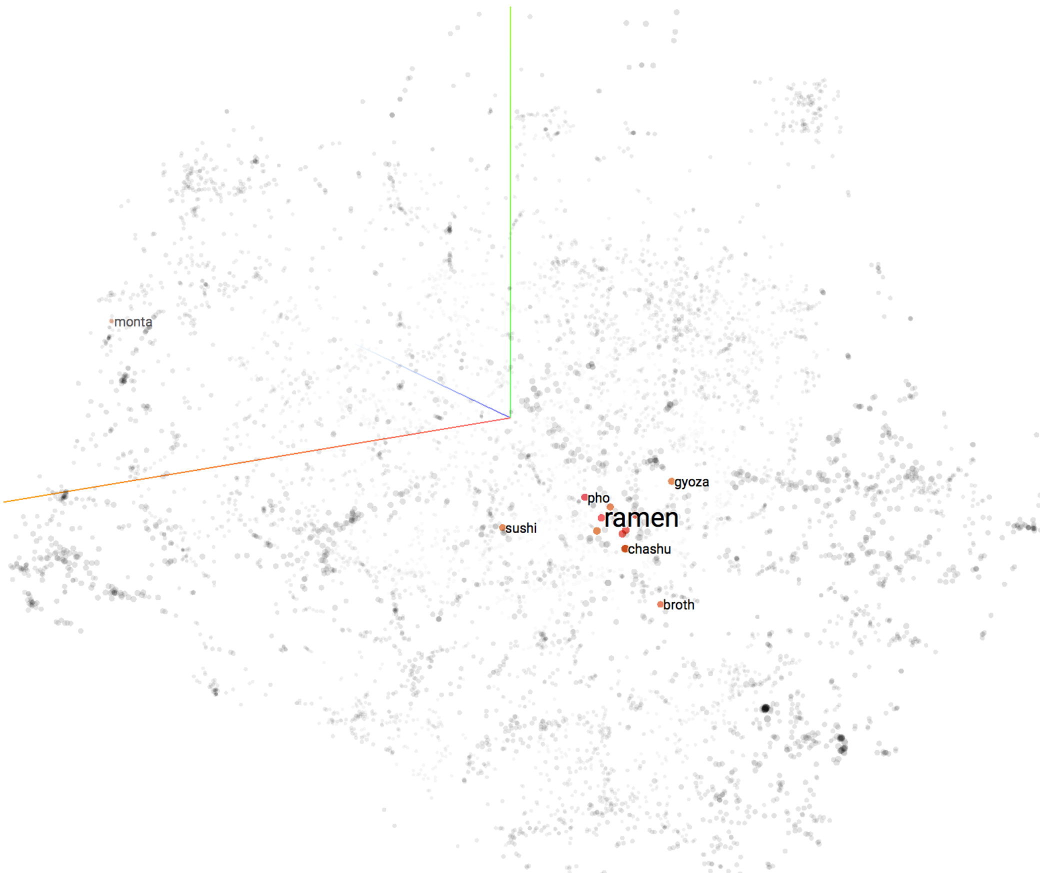

Figure 5.2 Similar words to “ramen” in 3D

As shown above, the similar words to “ramen” were pho, tonkotsu, tonkatsu, undo, broth, noodle, and so on. These are all noodle-like foods in Asia. For instance, pho is a Thai noodle. The author also found that Word2vec could group the typo words with the correct words. For example, the word tonkotsu should be “tonkatsu”. The cosine distance between two words was 0.197. The Euclidian distance was 0.628. The Word2vec model recognized both words and found these words belonged to the same category of the word “ramen”. Table 5.1 shows the cosine distance and Euclidean distance between “ramen” and the top ten most similar words.

Table 5.1 Top 10 similar words to “ramen”

Similar words to “ramen”	Cosine distance	Euclidean distance
pho	0.279	0.747
tonkotsu	0.338	0.823
tonkatsu	0.340	0.824
udon	0.351	0.838
broth	0.424	0.921
noodle	0.443	0.942
monta	0.448	0.946
chashu	0.451	0.949
sukiyaki	0.451	0.960
sushi	0.460	0.972

As table and figures showed, the Word2vec model similar food of “ramen”. It also provided some hints to what type of the food it is. These results show the food like “ramen.” All of them are Asian noodle-like food.

5.2.1.2 Burger

In this study, Word2vec model not only could detect Asian noodle food but also could show some hints of different categories of food. In Figure 5.3 and Figure 5.4, the words that have similar meanings of the word "burger” are highlighted.

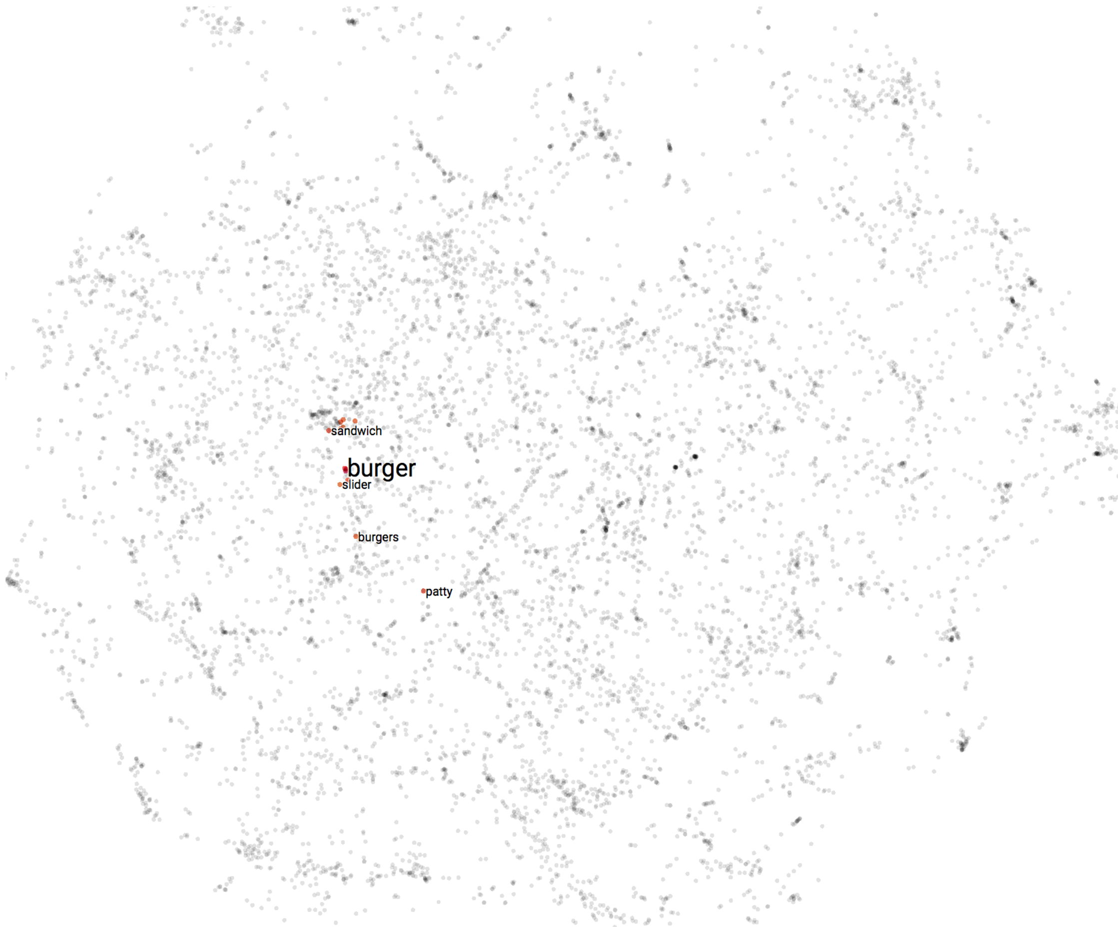

Figure 5.3 Similar words to “burger” in 2D

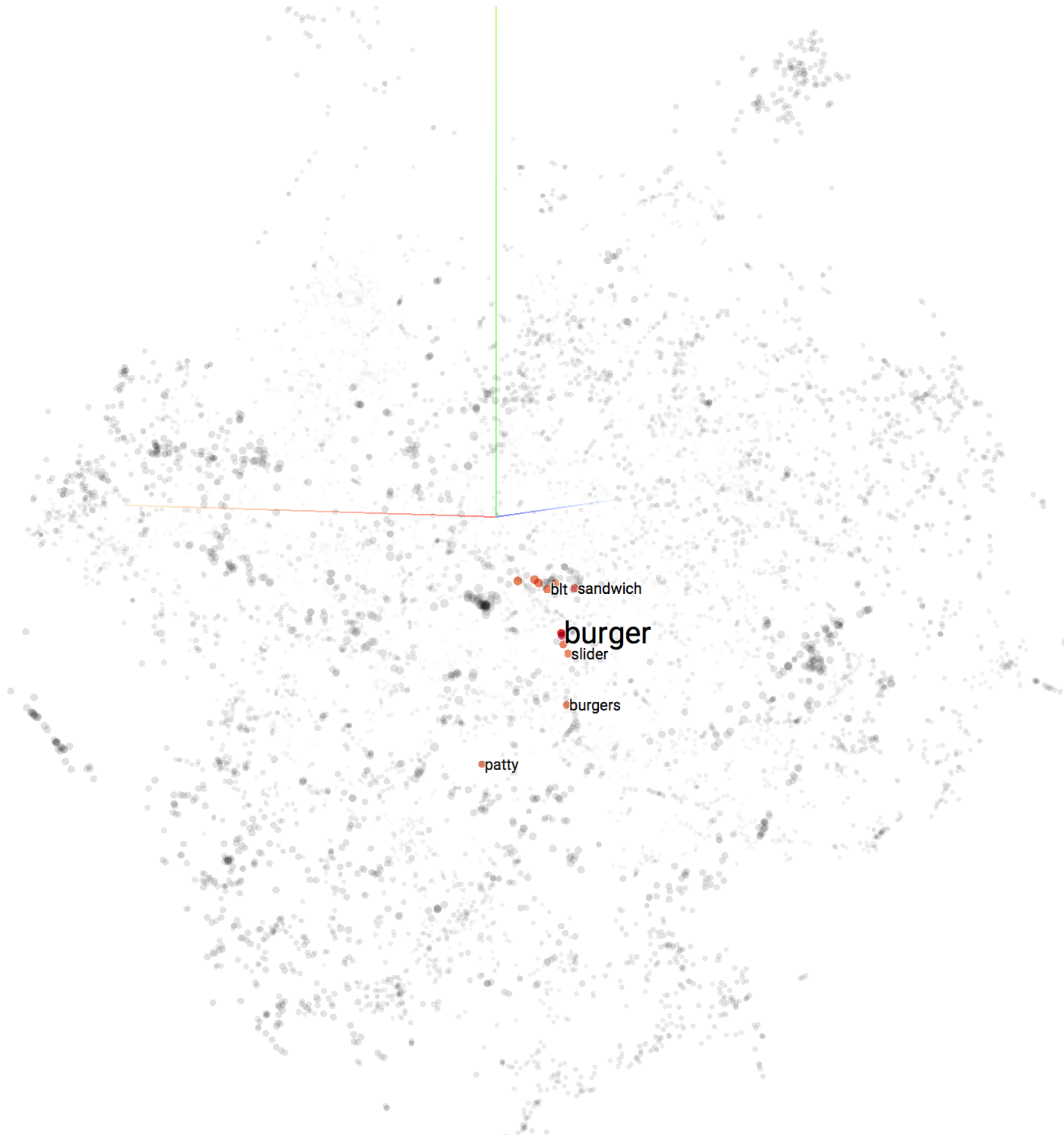

Figure 5.4 Similar words to “burger” in 3D

As demonstrated above, the similar words to “burger” were hamburger, cheeseburger, sandwich, hotdog, patty, veggie_burger, turkey_burger, and so on. When comparing Figure 5.3 with Figure 5.1, the author found the position of similar words of "ramen" was far from the position of the similar words of “burger.” That meant the system defined "burger" and "ramen" as two different things, even though they are both foods. The Word2vec model did find some differences between "ramen" and "burger." Table 5.2 shows the cosine distance and Euclidean distance between “burger” and the top ten most similar words.

Table 5.2 Top 10 similar words to "burger"

Similar words to "burger"	Cosine distance	Euclidean distance
Hamburger	0.221	0.665
Cheeseburger	0.238	0.689
Sandwich	0.353	0.840
Hotdog	0.383	0.875
Patty	0.383	0.875
Veggie_burger	0.399	0.894
Turkey_burger	0.412	0.907
Chicken_sandwich	0.413	0.909
Burgers	0.419	0.916
blt	0.423	0.920

5.2.1.3 Beverage

As the author suspected, the words that were similar to "beverage" should be far from the words that are like "ramen" and "burger". The cosine distance between "drink" and "beverage" was 0.245. The cosine distance between "drink" and "burger" was 0.973. The cosine distance between "drink" and "ramen" was 0.793. The results showed the word "drink" were more similar to "beverage" when compared with the similarity with "ramen" and "burger". The Figure 5.5 and Figure 5.6 show the similar words to "beverage".

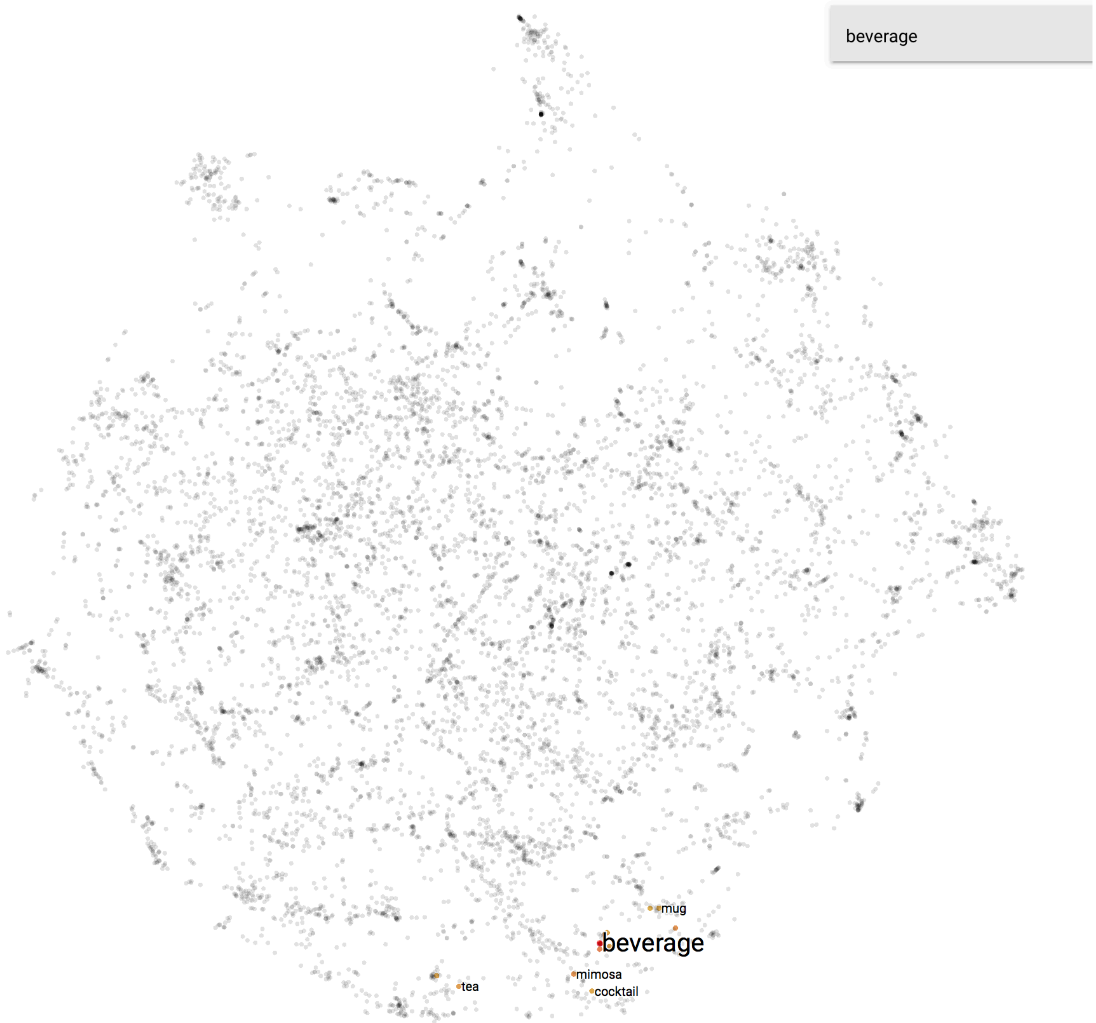

Figure 5.5 Similar words to “beverage” in 2D

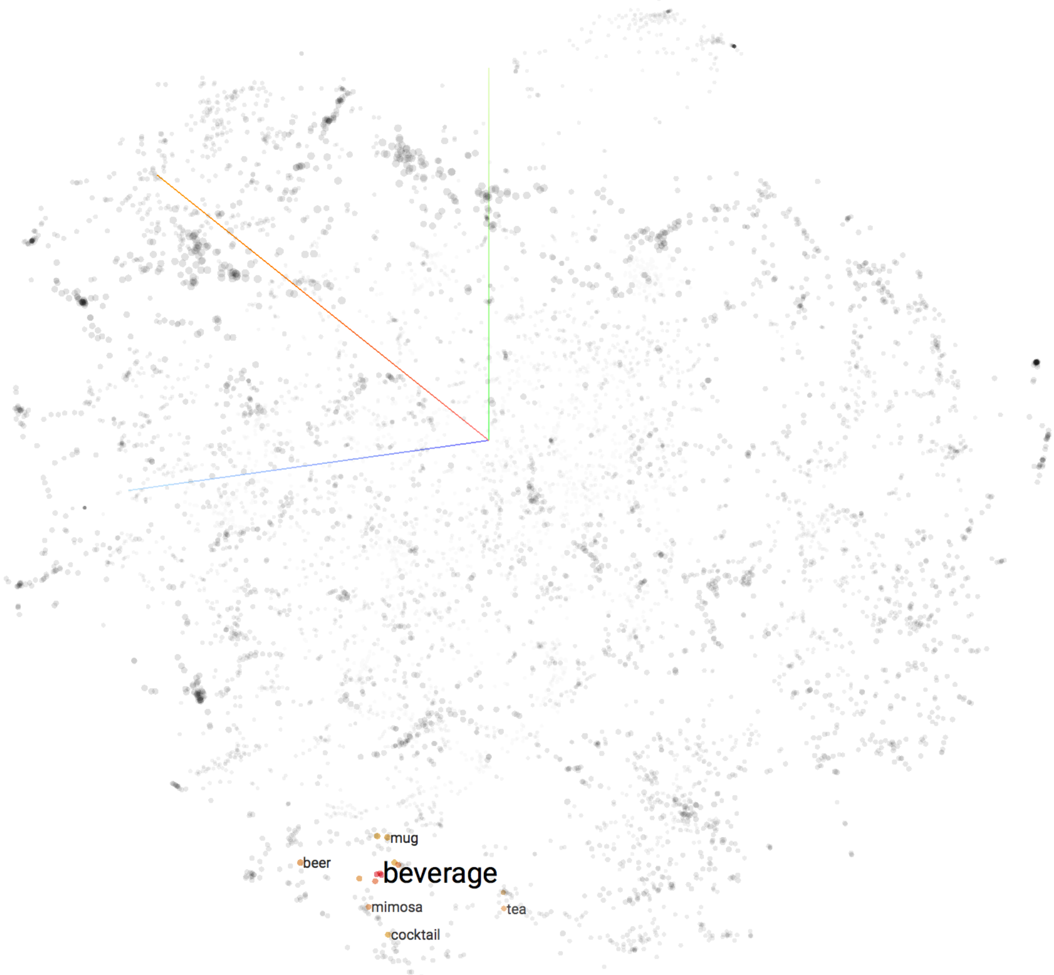

Figure 5.6 Similar words to “beverage” in 3D

As demonstrated above, the words similar to “beverage” were drink, beverages, drinks, soda, mimosa, beer, tea, and so on. These words are highly related to "beverage." Some of them had the same meaning as beverage; some of them were kinds of beverages. Also, the author found the similar words to beverage were far from the words like "ramen" and "burger". In Table 5.3, the cosine distance and the Euclidean distance between “beverage” and top ten most similar words are represented.

Table 5.3 Top 10 similar words to “beverage”

Similar words to “beverage”	Cosine distance	Euclidean distance
drink	0.245	0.700
Beverages	0.281	0.750
Drinks	0.441	0.940
Soda	0.460	0.960
Mimosa	0.499	0.999
Beer	0.505	1.005
Tea	0.536	1.035
libation	0.549	1.048
coffee	0.569	1.067
sodas	0.578	1.075

5.2.2 Service Words

In this section, two separate parts are discussed. The first part is about service. The second part is about "service person". Similar words to "service" are represented in the next section. The words that are alike to “staff” are represented in the following paragraph.

5.2.2.1 Customer Service

The Figure 5.7 and Figure 5.8 represent the word phrases similar to “customer service.”

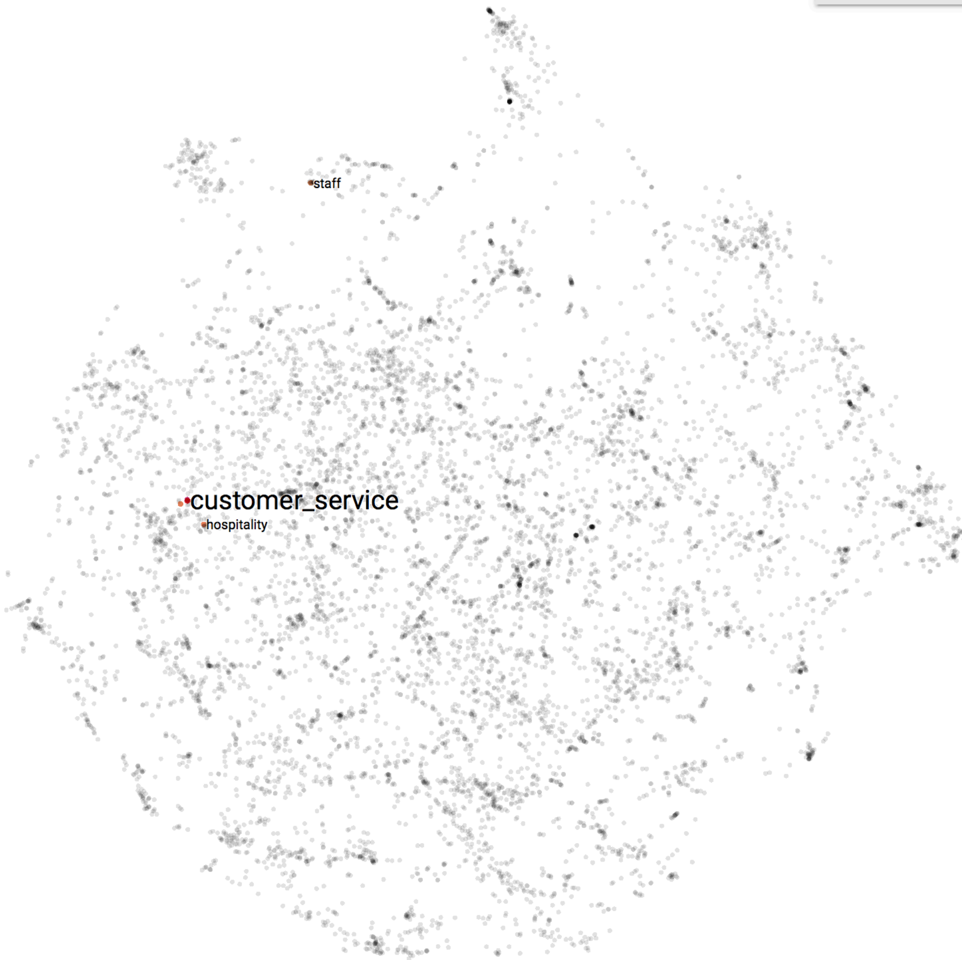

Figure 5.7 Similar words to "customer_service" in 2D

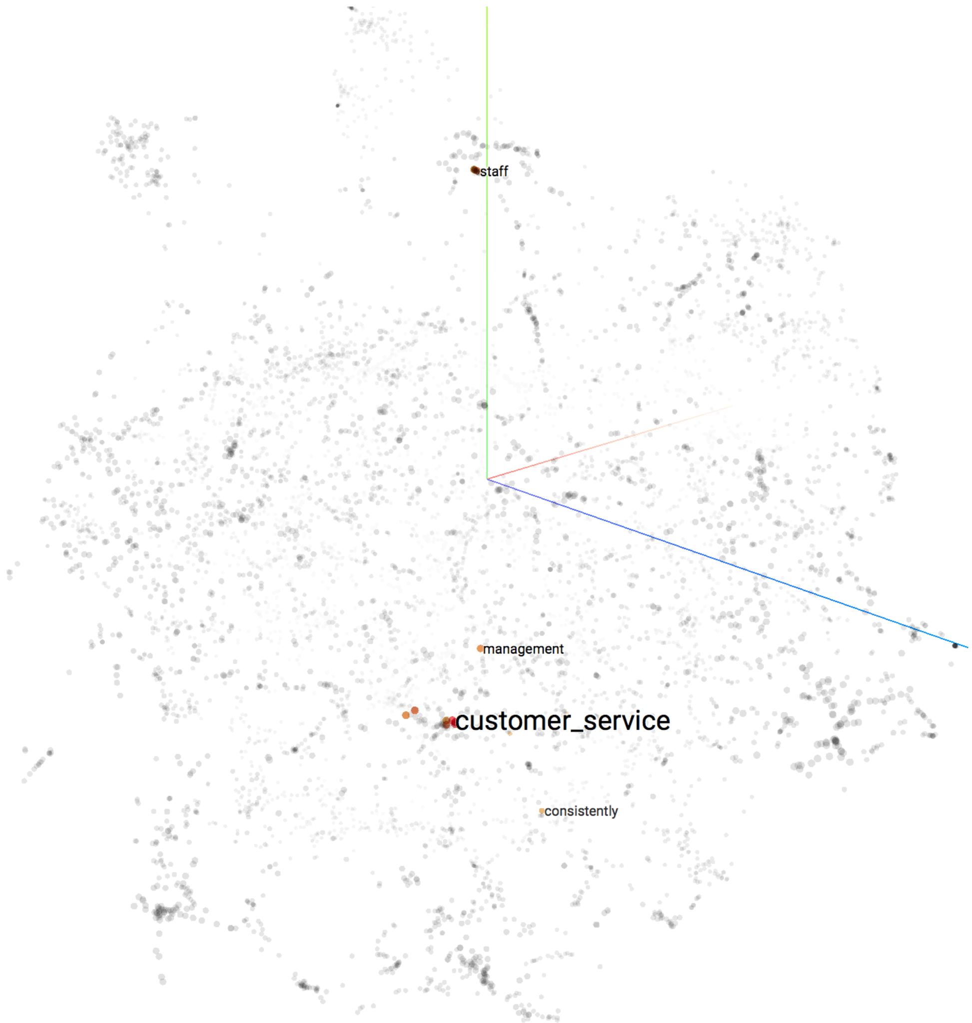

Figure 5.8 Similar words to “customer_service” in 3D

As Figure 5.7 and Figure 5.8 represent, the most similar words to “customer_service” are service, services, hospitality, attitudes, staff, attitude, and so on. These words are related to customer service. However, staff was also recognized as similar to customer_service. That means the Word2vec model was not perfect in demonstrating all related words. Some concepts were vague. The following Table 5.4 shows the cosine distance and Euclidean distance among “customer_service” and the top ten most similar words.

Table 5.4 Top 10 similar words to "customer service"

Similar words to "Customer_service"	Cosine distance	Euclidean distance
service	0.191	0.618
services	0.281	0.750
hospitality	0.424	0.920
attitudes	0.430	0.928
staff	0.463	0.963
attitude	0.597	0.997
management	0.502	1.002
waitstaff	0.533	1.032
professionalism	0.537	1.036
Beer_selection	0.546	1.045

From the Table 5.4, the author found that "beer_selection" was not similar to "customer_service". The Word2vec model did not work perfectly in finding similar words in this task.

5.2.2.2 Staff Words

The Figure 5.9 and Figure 5.10 represent the word phrases similar to "staff".

Figure 5.9 Similar words to “staff” in 2D

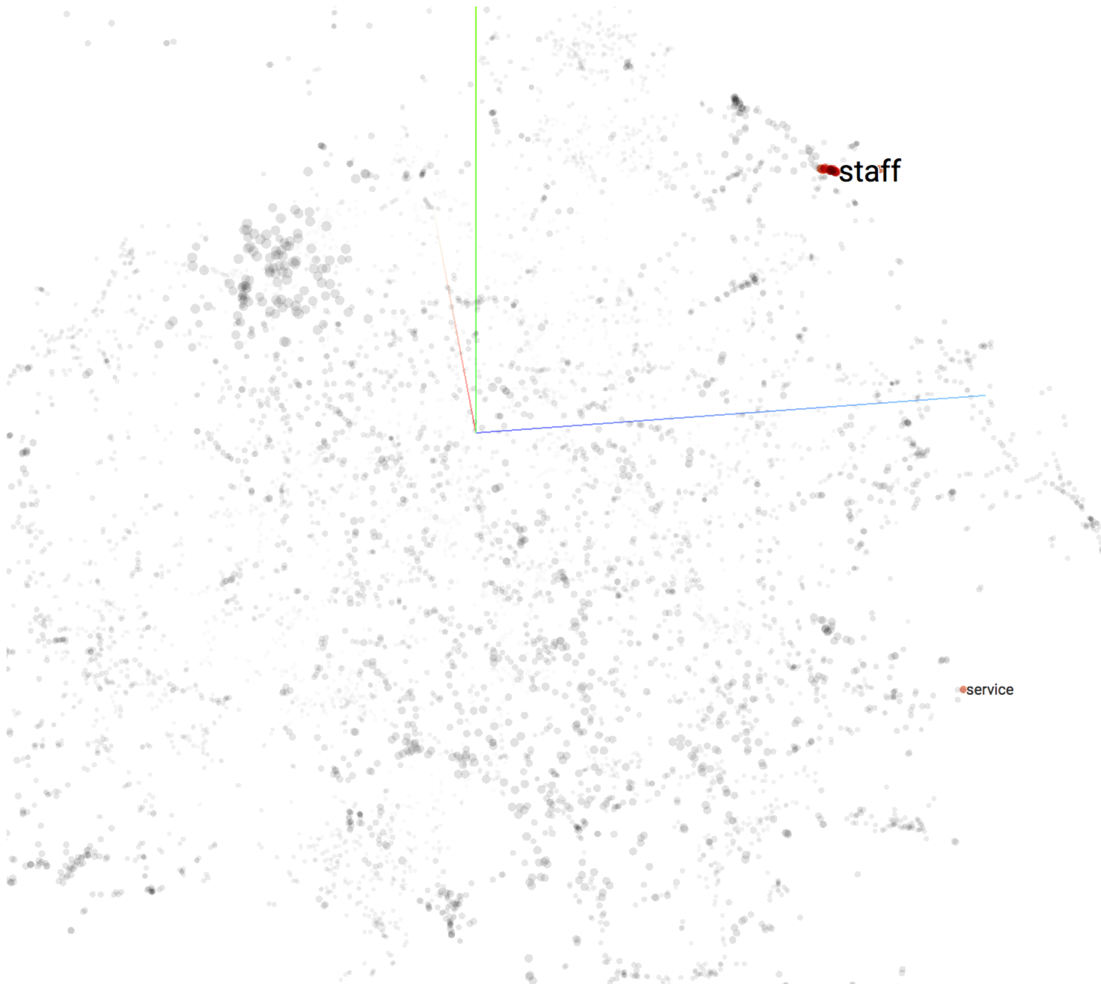

Figure 5.10 Similar words to “staff” in 3D

Unlike the previous part, these similar words were so close that were hard to split into the three-dimensional view. The most similar words to “staff” are waitstaff, staffs, servers, employees, wait_staff, bar_staff and so on. The Word2vec model did find the reasonably similar words. Table 5.5 represents similarity among “staff” and the top ten most similar words.

Table 5.5 Top 10 similar words to “staff”

Similar words to “staff”	Cosine distance	Euclidean distance
Waitstaff	0.154	0.555
staffs	0.292	0.776
servers	0.310	0.787
employees	0.313	0.792
Wait_staff	0.326	0.808
Bar_staff	0.343	0.828
bartenders	0.376	0.867
baristas	0.377	0.868
service	0.397	0.891
waiters	0.407	0.902

Table 5.5 presented the top ten similar words to staff. Most of them are related to staff. However, the word "service" is not very reasonable when compared with other words.

5.2.3 Ambience Words

In this section, some similar words to “ambience” were found and represented. Figure 5.11 and Figure 5.12 show these results.

Figure 5.11 Similar words to “ambience” in 2D

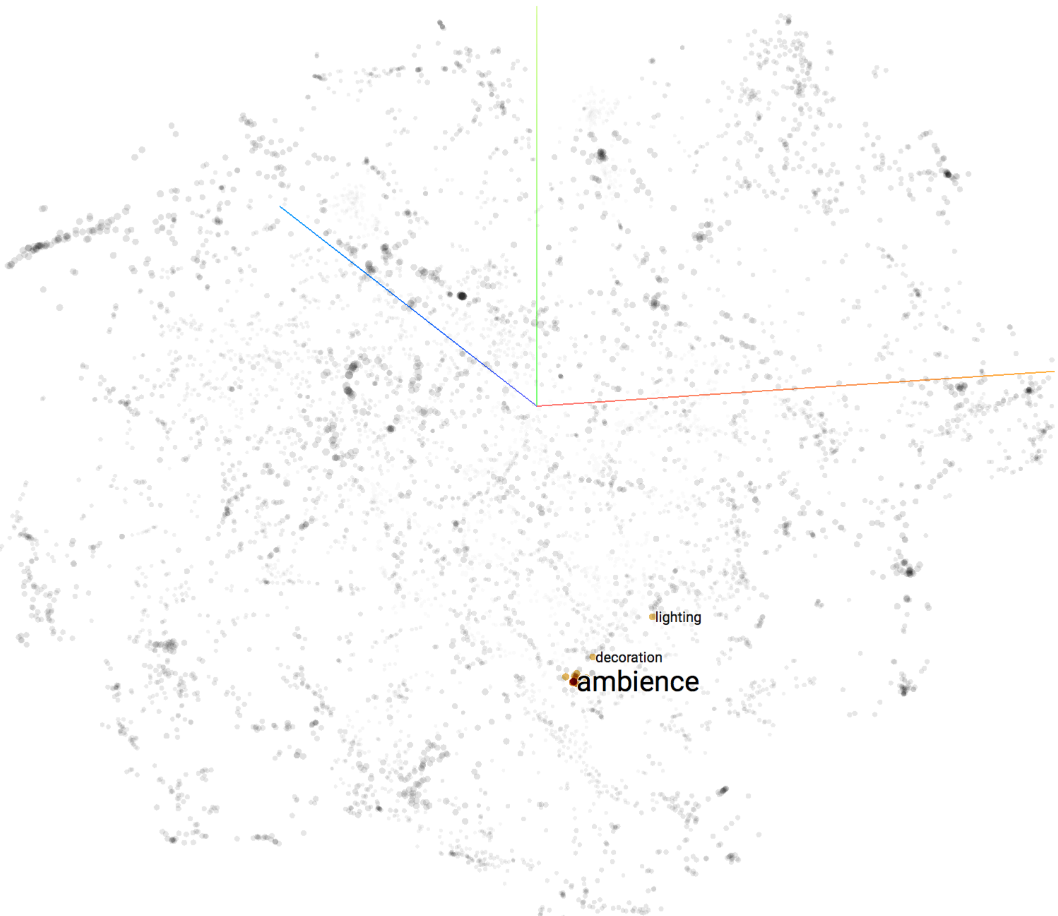

Figure 5.12 Similar words to "ambience" in 3D

As figures show, the author found the words most similar to "ambience" were ambiance, atmosphere, environment, decor, setting, interior, layout, and so on. It was clear that these words are highly related to ambience. The word "ambiance" is same word "ambience." In Table 5.6, the distances among these word vectors are shown.

Table 5.6 Top 10 similar words to "ambience"

Similar words to "ambience"	Cosine distance	Euclidean distance
ambiance	0.019	0.194
atmosphere	0.071	0.376
environment	0.180	0.600

Table 5.6 (continued)

decor	0.181	0.602
setting	0.284	0.754
interior	0.299	0.773
layout	0.384	0.877
scenery	0.407	0.902
lighting	0.420	0.917
surrounding	0.422	0.918

The author thought these words were related to the ambience of the restaurants. The cosine distance between the word “ambience” and “ambiance” is extremely small, which is reasonable because these are words with same meaning. More than that, the system could find abbreviations like “deco,” which means the decoration. The cosine distance between the word “deco” and “decoration” is 0.431 and the Euclidean distance is 0.929. That means these are similar words.

5.2.3 Price Words

In this section, some similar words that are related to the price concept were found and represented. Figure 5.13 and Figure 5.14 show words similar to “fee.”

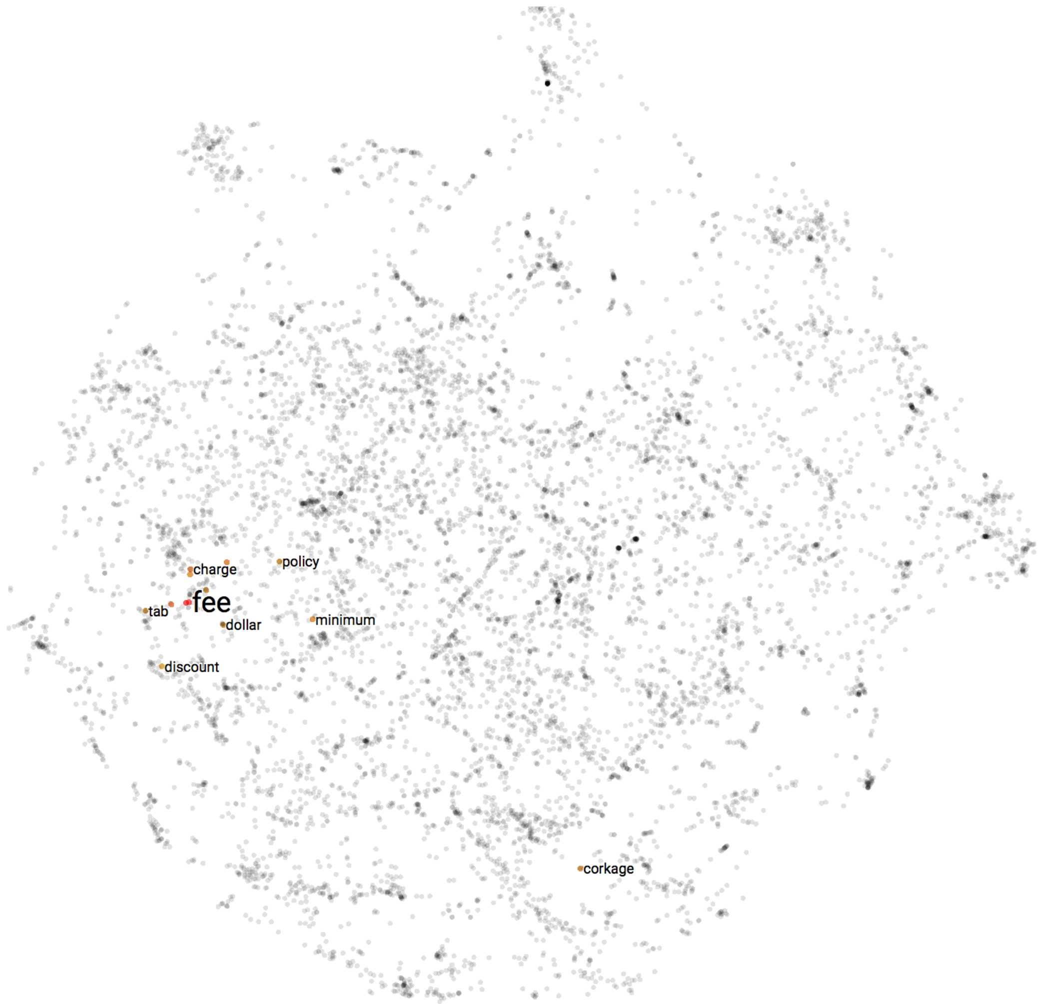

Figure 5.13 Similar words to “fee” in 2D

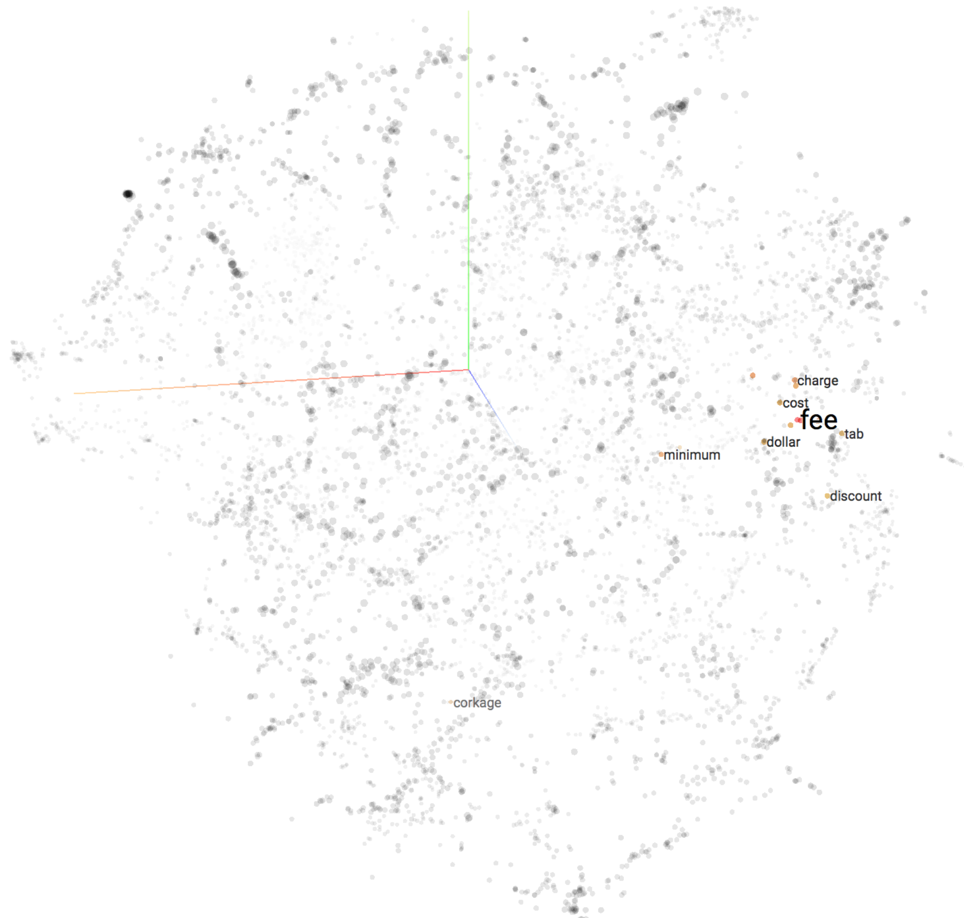

Figure 5.14 Similar words to “fee” in 2D

As figures show, the author found the words most similar to “fee” are fees, gratuity, charge, upcharge, minimum, charges, corkage, and so on. It was clear that most of these words are highly related to “fee.” However, the system found the word “minimum” was similar to “fee.” From the author's perspective, this result is wrong. Table 5.7 shows the distances between these word vectors.

Table 5.7 Top ten words similar to “fee”

Similar words to “fee”	Cosine distance	Euclidean distance
fees	0.302	0.778
gratuity	0.465	0.964
charge	0.486	0.986

Table 5.7 (continued)

upcharge	0.487	0.986
minimum	0.517	1.017
charges	0.530	1.030
corkage	0.535	1.035
taxes	0.560	1.059
cost	0.561	1.064
policy	0.566	1.071

Most of the results were reasonably related to the word "fee." However, there were two unreasonable words: "minimum" and "policy." The reason may be that the words like "minimum" and "policy" co-occurred often with words like "fee" and "gratuity." This is one of the drawbacks that the author found in implementation.

5.3 Analyze Baseline Model

The baseline model was LDA model, which was introduced in previous sections. Compared with the Word2vec model, there were several tasks that the LDA model could not achieve.

Firstly, the LDA model could not detect the words with low frequency. In the Word2vec model, the words with low frequency could be trained. The author chose to train the Word2vec model that frequency was larger than two because of the efficiency of the program. The Word2vec model could be trained on all the words in the reviews even with the word frequency set to one. However, the LDA model could not detect the low-frequency words and clustered them. When used as a baseline model to represent the words into vectors, some words with low frequency were ignored. One of the reasons is that, the LDA model is a count-based model, and it was more sensitive to the words with high frequency. When the program transformed the words into vectors, some words with low frequency that were represented as vectors contained 200 zeros. The vector with 200 zeros could not represent the useful information of the words. The author deleted them and analyzed the other words.

Secondly, the LDA model could not detect the hint of the food categories. There was no hint to which categories the food belonged. The Figure 5.15 shows the similar words to “pizza” based on the LDA model.

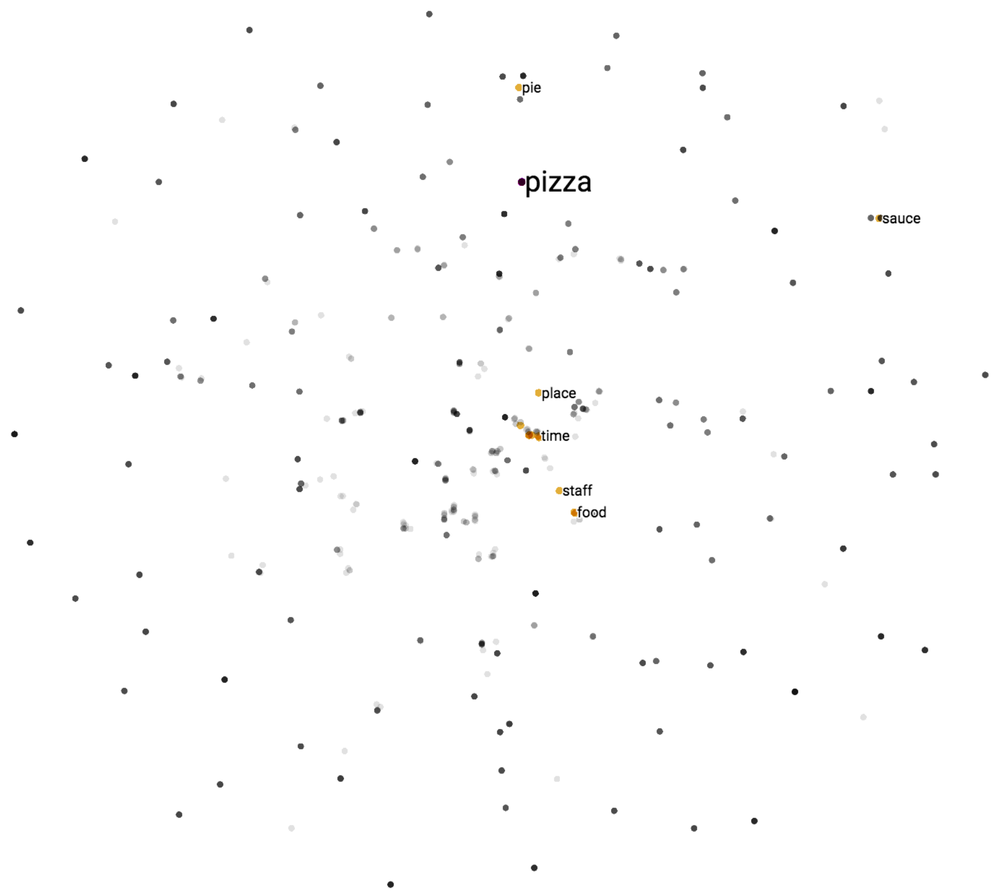

Figure 5.15 Similar words to “pizza” in 2D based on LDA model

As figures show, the author found the words most similar to “pizza” and wings, oven, garlic, sauce, pie and so on. Some of the words were related to food. However, it could not detect the hint of the food category. That was another task that LDA model could not achieve. Table 5.8 shows the distance among these similar words to “pizza”.

Table 5.8 Top ten words similar to “pizza” based on LDA model

Similar words of “pizza” (LDA)	Cosine distance	Euclidean distance

Table 5.8 (continued)

wings	0.001	0.001
oven	0.001	0.001
garlic	0.024	0.221
sauce	0.720	1.200
pie	0.876	1.323
food	0.993	1.409
service	0.995	1.411
time	0.996	1.411
wish	0.996	1.411
fish	0.996	1.412

There were some reasonable words related to the word "pizza" like oven, sauce and food. However, the baseline model could not detect the categories of the food. More than that, there were some unreasonable results like time and service.

5.4 Test on Test Data

5.4.1 Test Results

5.4.1.1 Measure Method

The aim of the system was using Word2vec model to find aspect words. It could be viewed as an information retrieval problem. In other words, the system tried to search for aspect words from the review. To test an information retrieval system, the common measure is precision, recall, and F-score. The measure methods used are described in following paragraph.

The precision formula was showed as follows:

$$precision = \frac{\{relevant\ words\} \cap \{retrieved\ words\}}{\{retrieved\ words\}}$$

In this formula, the relevant words were correct aspect words. The retrieved words were the words that system found. Precision measured how many retrieved words are right. For instance, there were ten aspect words in a review. The system found eight, and six of them are right. The precision rate is 6/8, which is 75%.

The recall formula is shown as follows:

$$recall = \frac{\{relevant\ words\} \cap \{retrieved\ words\}}{\{relevant\ words\}}$$

Recall measures how many correct aspect words were found by the system. For example, there were 10 aspect words in a review. The system found eight, and six of them are right. The recall rate is 6/10, which is 60%.

The F-score formula is shown as follows:

$$F = 2 \times \frac{precision \times recall}{precision + \ recall}$$

It measured the harmonic mean of the precision and recall. The F-score gave the author an overall performance of the system.

5.4.1.2 Test Result

The Figure 5.16 and Table 5.9 show the test results. The author tested two models on the test data. The data contained 1,076 manually labeled data.

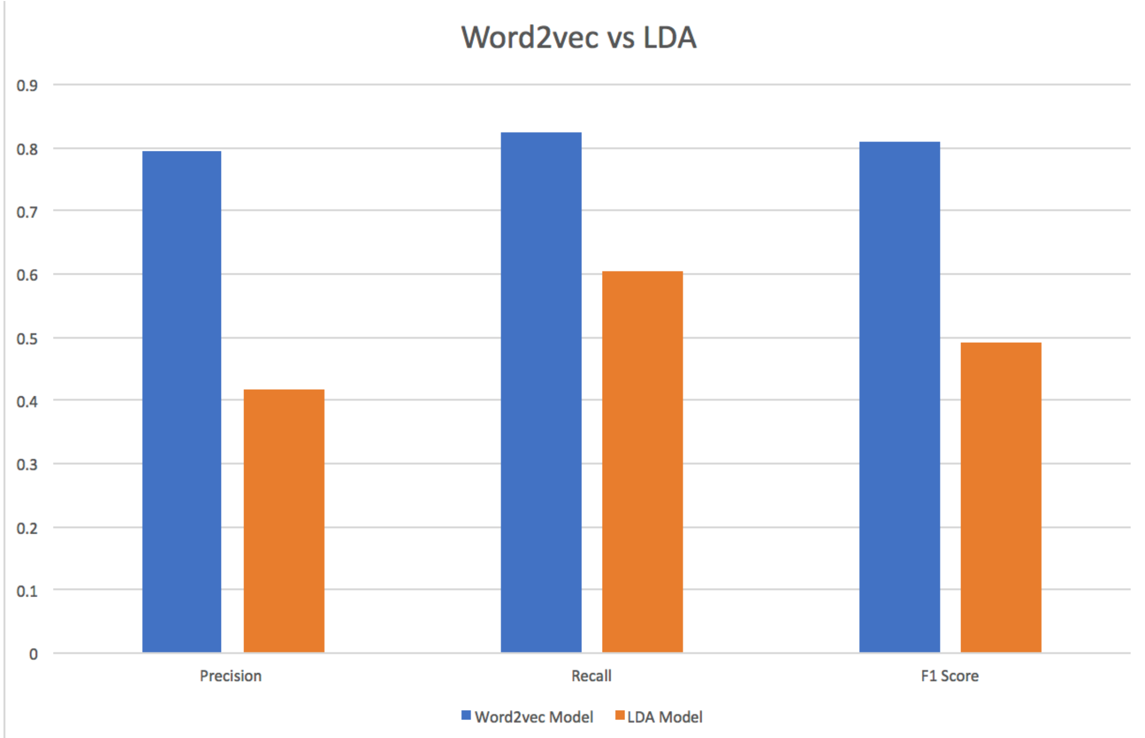

Figure 5.16 Test result of Word2vec model and LDA model

In the Figure 5.16, the blue bars reflect the performance of the Word2vec model. The orange bars reflect the performance of the baseline model, which is the LDA model. As a result, the Word2vec model outperformed LDA model under precision, recall, and F-score measure. The Table 5.9 shows the details value.

Table 5.9 Test result of Word2vec model and LDA model

	Precision	Recall	F score
Word2vec Model	0.7941	0.8228	0.8079
LDA Model	0.4162	0.6038	0.4927

As a result, the author has 90% confidence to conclude that the precision of Word2vec model is 79.41% (±2.5%), the recall is 82.28% (±2.5%), and the F-score is 80.79% (±2.5%). The recall of the LDA model performed poorly. The precision of the Word2vec model outperformed the LDA model. One reason is that the LDA model ignored some aspect words with low frequency. The Word2vec could retrieve them even though the word appeared only several times.

The Figure 5.17 and Table 5.10 represent the test results of the Word2vec model and LDA model that used only high frequency words in the clusters.

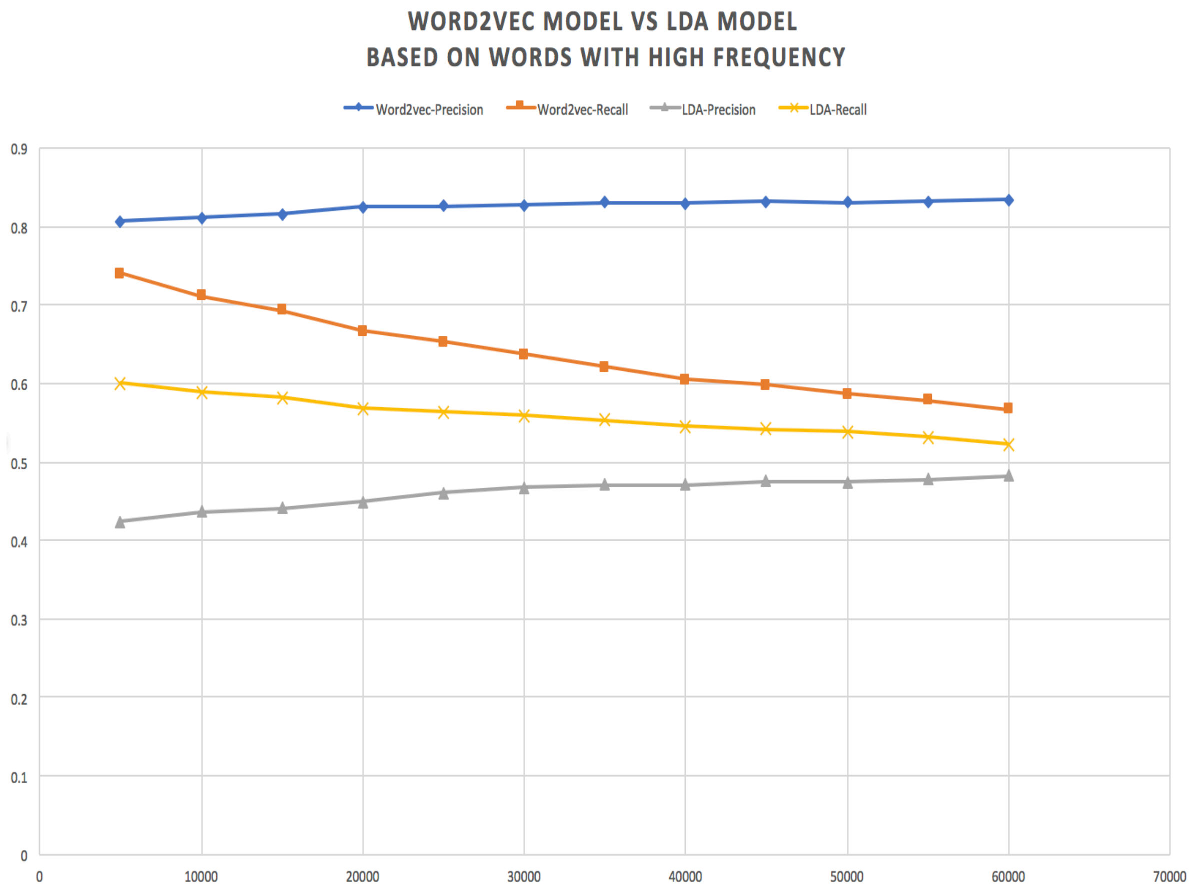

Figure 5.17 Word2vec model vs LDA model with high frequency words

In the Figure 5.17, the blue line represents the precision of the Word2vec model that used only words with high frequency. The orange line shows the recall of the Word2vec model based on the high-frequency words. The yellow line shows the precision of the LDA model with high-frequency words. The grey line shows the recall of the LDA model based on high-frequency words. The horizontal axis shows the lower bound of the word frequency. For instance, the value 10,000 means the words that are equal to or higher than 10,000 from the clusters were used for extracting the aspect words. In the Table 5.10, the details are shown.

Table 5.10 Test result of Word2vec model with high frequency words

Word frequency	Word2vec-Precision	Word2vec-Recall	LDA-Precision	LDA-Recall
5,000	0.8066	0.7401	0.4235	0.6000
10,000	0.8114	0.7114	0.4370	0.5890
15,000	0.8162	0.6926	0.4413	0.5820
20,000	0.8247	0.6668	0.4488	0.5678
25,000	0.8262	0.6534	0.4606	0.5640
30,000	0.8276	0.6365	0.4676	0.5595
35,000	0.8310	0.6204	0.4707	0.5536
40,000	0.8289	0.6048	0.4705	0.5452
45,000	0.8313	0.5979	0.4753	0.5423
50,000	0.8302	0.5866	0.4743	0.5383
55,000	0.8320	0.5783	0.4777	0.5321
60,000	0.8339	0.5675	0.4821	0.5219

When the number of the lower bound of the word frequency increased, the recall of the two model started to drop, and the precision increased a little bit. It happened because the number of words that were selected by models kept decreasing, which made the system find less relevant words. When the lower bound increased from 5,000 to 60,000, the recall of the Word2vec model dropped by about 28%. The recall of the LDA model dropped by about 8%. It meant that the Word2vec model could find more aspect words with low frequency. Elimination of low-frequency words affected more on the Word2vec model. From all perspective previously tested, the Word2vec model performed better than LDA model in extracting aspect words.

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

The purpose of this study was to test the performance of the Word2vec model on the aspect words extraction. In the previous section, the author analyzed the results of Word2vec model and the performance on the test data compared with the baseline model. Based on the results, the author concludes that the Word2vec model works better than the LDA model on the aspect words extraction task. However, the model is far from perfect. In the following paragraph, the author explains the advantages and disadvantages of the model.

At first, the author's approach could detect some hints of different categories of the food. Before the implementation of the Word2vec model, the author believed that the food names should be grouped together but food words could not distinguish food categories. However, the results showed that the Word2vec could find some hints of different food categories. The Asian food words were dissimilar to the American food, like burger and pizza. The system found similar words to "noodles" such as rice noodle, ramen, and pho. These foods are noodle-like foods.

Secondly, the Word2vec model could find some typos in the reviews. The reviews of restaurants were different from the many other formal articles like books, news, and formal documents. There were lots of typos and wrong words in the reviews. For instance, the similar words to "atmosphere" were "atmostphere", "atomosphere" and "atmospher." Customers may have the ability to understand these typos. For the computer, these words could mean totally different things. However, by using the Word2vec model, these words are represented as similar vectors. When using word vectors as inputs for computers, these words are considered similar.

Thirdly, the Word2vec model could help people to detect some informal words which were found in many of the online reviews. The people may use "awesooooome" in the review instead of "awesome." The Word2vec model was able to find some of these informal words. For example, the similar words to "yum" found by the system were "mmmmm", "yummm" and "yumm." These words are informal words, which are not in the dictionary but appear in the reviews. It is difficult for the traditional dictionary to find the informal words like the example provided here. However, the Word2vec model could easily find them.

The Word2vec model did not work perfectly. It had some unreasonable results. For instance, it found one of the similar words to “fee” is “minimum.” It could happen if “minimum” and “fee” always co-occur together. In that case, the words around these two words are most likely the same.

In summary, it is possible to use the Word2vec model to extract aspect words. The Word2vec model outperformed the LDA model in this specific task. Based on the test data, the Word2vec model performed better than LDA model. More than that, the Word2vec could detect the same category food and find misspelled words and informal words from the reviews.

6.2 Recommendations

There are several recommendations for the future study in two perspectives. First is about how to improve the model to find more accurate aspect words. The model was heavily dependent on the part of speech tagging tool. In this study, the aspect words are a noun or nouns. If the tagging tool could not find the correct noun or nouns, the performance would drop. To improve the performance, a more accurate tagging tool would lead a better performance.

The second recommendation for future studies is to find the opinion of the specific aspect. The Word2vec model also could be used to detect sentiment adjective words and cluster the words with the positive or negative sentiment. For instance, the trained model found the words most similar to "excellent" were "superb," "terrific," and "awesome." The most similar words of "suck" were "screw" and "stink." The outcome showed that the Word2vec model may group the positive adjective words together. Also, the model could cluster the negative words together.

APPENDIX A. SELECTED CLUSTERS BASED ON WORD2VEC

2 ['eggs', 'egg', 'pancakes', 'gravy', 'waffles', 'waffle', 'crepe', 'hash', 'omelet', 'benedict', 'biscuits', 'grits', 'pancake', 'omelette', 'biscuit', 'browns', 'skillet', 'quiche', 'scramble', 'hash_browns', 'hashbrowns', 'chilaquiles', 'huevos', 'breakfast_burrito', 'croque', 'benny', 'rancheros', 'breakfast_sandwich', 'beef_hash', 'frittata', 'eggs_benedict', 'home_fries', 'florentine', 'omlette', 'monsieur', 'huevos_rancheros', 'homefries', 'lorraine', 'scrambler', 'spuds', 'breakfast_potatoes', 'omlet', 'egg_sandwich', 'omelete', 'everything_bagel', 'madam', 'omlete']

6 ['chocolate', 'ice', 'yogurt', 'cupcake', 'donut', 'cookie', 'shake', 'vanilla', 'caramel', 'custard', 'frosting', 'pastry', 'brownie', 'nutella', 'milkshake', 'brulee', 'pistachio', 'doughnut', 'sundae', 'mousse', 'icing', 'oreo', 'whip', 'toffee', 'hazelnut', 'puff', 'marshmallow', 'barrel', 'buttercream', 'carmel', 'peanut_butter', 'chocolate_cake', 'butterscotch', 'marshmallows', 'crumb', 'graham', 'meringue', 'ganache', 'choc', 'fondant', 'fraiche', 'choco', 'praline', 'brule', 'shortbread', 'whip_cream', 'phyllo', 'wafer', 'chocolate_sauce', 'wafers', 'chocolate_chips', 'caramel_sauce', 'filo', 'chocolate_syrup']

8 ['whites', 'yolk', 'supreme', 'drop_soup', 'wich', 'grinder', 'yolks', 'hollandaise_sauce', 'shogun', 'wiener', 'meat_lovers', 'hashbrown', 'cfs', 'applesauce', 'dawg', 'pattie', 'latkes', 'lyonnaise', 'knish', 'pide', 'luau', 'marina', 'rossini', 'bacon_jam', 'chopped_salad', 'perfecto', 'athens', 'corn_beef', 'tartine', 'pbj', 'sausage_gravy', 'mcmuffin', 'pilgrim', 'buffalo_chicken', 'burgushi', 'missus', 'egg_salad', 'pellini', 'stallion', 'sauerbraten', 'bagel_sandwich', 'monte_cristo', 'pelmeni', 'corn_dog', 'matador', 'spud', 'family_meal', 'panino', 'lettuce_wrap', 'pimento_cheese', 'corndog', 'galette', 'billionaire', 'hubster', 'zuchinni', 'farm_burger', 'dipper', 'mezze', 'shorty', 'chicken_burger', 'haluski', 'pannini', 'pierogie', 'carvery', 'rib_tips', 'yaya', 'brizza', 'veggie_pizza', 'cornbeef', 'muffaletta', 'margherita_pizza', 'pizza_pie', 'shipwreck', 'meat_lover', 'garden_salad', 'gobbler', 'sandwitch', 'dynasty', 'mezza', 'wolfpack', 'wurst', 'paellas', 'commuter', 'cevapi', 'briskets', 'ofcourse', 'hock', 'falafel_sandwich', 'latke', 'burek', 'bunz', 'hash_brown', 'lamb_burger', 'dinner_salad', 'muffuletta', 'banh_xeo', 'greek_salad', 'currywurst', 'pork_shank', 'reuben_sandwich', 'saurkraut', 'tomato_bisque', 'raviolo', 'chop_salad',

'beef_patty', 'sonoran_dog', 'brava', 'bagel_sandwiches', 'lavraki', 'burger_patty', 'totes', 'teaser', 'cheese_pizza', 'zuchini', 'shrimp_ceviche', 'absolutley', 'boyger', 'tru', 'gladiator', 'buger', 'hade', 'fritatta', 'duck_fat', 'blta', 'crapes', 'trotter', 'frenchie', 'chicago_style', 'mcnuggets', 'mambo', 'lgbtq', 'hasbrowns', 'wheat_toast', 'wieners', 'sausage_sandwich', 'fois_gras', 'shrimp_sauce', 'country_potatoes', 'chencho', 'schreiners', 'wedding_soup', 'croissant_sandwich', 'pastie', 'submarine', 'meze', 'hashes', 'stuffed_dates', 'kono', 'bobbi', 'plaintains', 'jaeger', 'santa_fe', 'chicken_panini', 'chiptole', 'porkie', 'breakfast_pizza', 'britannia', 'savory_crepes', 'griller', 'zucchini_sticks', 'bisquits', 'waffle_sandwich', 'juicy_burger', 'medium_spicy', 'goodnight', 'tasso', 'corn_nuggets', 'tartares', 'peri', 'variant', 'bennie', 'meatloaf_sandwich', 'poutin', 'beef_rib', 'croquet', 'phillies', 'caddy', 'lamb_chop', 'egg_benedict', 'gilroy', 'shack_burger', 'beer_bread', 'curry_puffs', 'casablanca', 'croque_monsieur', 'pizza_slice', 'pupu', 'strata', 'chicken_club', 'steakburgers', 'salmon_sandwich', 'chipoltle', 'blts', 'margarita_pizza', 'gravey', 'bacons', 'spicy_sausage', 'butter_burger', 'end_result', 'cobia', 'tadas', 'frittatas', 'oxtail_fries', 'pan_pizza', 'specialty_pizza', 'corn_fritters', 'wafflewich', 'lobster_salad', 'cannonballs', 'ciopinno', 'shepherds_pie', 'calamaris', 'steak_burger', 'side_items', 'forager', 'omlets', 'hand_cut', 'rock_shrimp', 'corn_soup', 'urth', 'chorizo_burrito', 'linguica', 'lamb_skewers', 'oyster_shooters', 'meat_pies', 'knishes', 'griddlecakes', 'crepe_station', 'rice_balls', 'pub_burger', 'weiner', 'catchup', 'wer', 'samich', 'chimmichurri', 'marguerita', 'cheesteak', 'ranches', 'steak_n', 'veggie_wrap', 'zucchinis', 'matchstick', 'rouladen', 'frog_legs', 'breakfast_wrap', 'potato_puree', 'croquettas', 'kinoko', 'nog', 'sub_sandwich', 'taos', 'spicy_foods', 'bizness', 'egg_dishes', 'sage_chicken', 'korokke', 'zinger', 'patty_melt', 'hunka', 'hubbs', 'bbq_shrimp', 'cheese_steak', 'sausage_patties', 'bacon_waffles', 'tsoynamis', 'parmesain', 'dippin', 'size_pizza', 'casseroles', 'porkwich', 'yumyum', 'smoke_meat', 'consomme', 'beef_stroganoff', 'chef_salad', 'borsch', 'pistachio_baklava', 'sweet_tea', 'rissoto', 'ham_sandwich', 'calypso', 'falafel_pita', 'macncheese', 'meat_pizza', 'stuffed_pizza', 'juicy_sauce', 'desert_menu', 'sopaipillas', 'broc', 'fruit_plate', 'duet', 'biscuts', 'handcut', 'phillys', 'lunasagna', 'flavor_combination', 'lobster_pasta', 'provolone_cheese', 'burnt_ends', 'cappicola', 'chicago_pizza', 'meatball_pizza', 'chicken_pesto', 'clucker', 'boulettes', 'rockfish', 'gourmet_burgers', 'paninni', 'acapulco', 'chicken_sausage', 'unami', 'mcrib', 'hott', 'overeasy', 'pasta_trio', 'chicharones', 'minestrone_soup', 'cajon', 'sald', 'chitterlings', 'omellete', 'mezzes', 'brie_sandwich', 'kobe_steak', 'pastrami_burger', 'sausage_links', 'roast_chicken', 'buffalo_burger', 'plum_sauce', 'foldover', 'appys', 'california_club', 'shoestrings', 'cheesesteak_sandwich', 'chimichuri', 'cajun_pasta', 'potato_tacos', 'chimichurri_sauce', 'corn_muffin', 'beef_patties', 'pirogies', 'quesarito', 'crabe', 'hangover_burger', 'baconator', 'moist_brisket', 'octupus', 'breakfast_buzz', 'chicken_avocado', 'rosarito', 'potpie', 'bbq_pizza', 'sando', 'twisters', 'walu', 'crispy_bacon', 'tnb', 'lvl', 'kolache', 'hake', 'paper_thin', 'gnochi', 'tri_tip', 'tempura_shrimp', 'shrimp_burger', 'gobblerito', 'dagwood', 'shishito_peppers', 'calazone', 'baja_sauce', 'mcmuffins', 'ceaser_salad', 'lobster_spaghetti', 'country_gravy', 'veggie_dog', 'potato_pancake', 'beef_tips', 'rubens', 'duck_tacos', 'hangar_steak', 'kettle_corn', 'fritata', 'hotsauce', 'enchantment', 'breakfast_item', 'pepperoni_rolls', 'crab_ragoons', 'breakfast_burger', 'sirachi', 'medium_spice', 'savory_crepe', 'dali', 'dbl', 'yucca_fries', 'ravs', 'banoffee_pie', 'ham_steak', 'spicy_crab', 'fritte', 'vegetariana', 'coctail', 'kim_chi', 'liverwurst', 'burgher', 'yummy_yummy', 'capicolla', 'poutine_fries', 'stacker', 'ronin_burger', 'spaetzel', 'plaintain', 'orange_sauce', 'tastykakes', 'mini_burgers', 'rockport', 'breakfast_plate', 'heros', 'piquancy', 'wedgie', 'tofu_fries', 'portobella', 'yummi', 'croquets', 'vesuvius', 'sanwiches', 'whoppers', 'jazz_fries', 'margo', 'perlou', 'pain_perdu', 'sasquatch', 'cabbage_rolls', 'salmon_tartare', 'turducken', 'firey', 'crawpuppies', 'margerita', 'chicken_meal', 'wienerschnitzel', 'lunch_combos', 'russet', 'perogi', 'ckn', 'crab_bisque', 'hlk', 'egusi', 'veggie_omelette', 'manti', 'cheese_sticks', 'baha', 'cheesy_potatoes', 'footlongs', 'risottos', 'perogie', 'marguerite', 'escargo', 'beach_club', 'rice_ball', 'chk', 'mussells', 'california_burger', 'pork_bbq', 'manifesto', 'avocado_rolls', 'ricotta_pizza', 'srirachi', 'breakfast_bowl', 'chipotle_chicken', 'beaters', 'gyro_pita', 'gnocchis', 'civiche', 'scotch_eggs', 'chicken_nuggets', 'barbque', 'lambchops', 'rissotto', 'veggy', 'crab_wontons', 'bolillo', 'jagerschnitzel', 'combo_plates', 'meatlovers', 'mango_habanero', 'bmoc', 'spinach_pizza', 'chicken_entree', 'seafood_paella', 'johnnycakes', 'smokestack', 'divers', 'gnoochi', 'dinner_plates', 'trifecta_sauce', 'glass_noodles', 'meat_pie', 'bennies', 'coney_dog', 'rib_steaks', 'lunch_sandwiches', 'salmon_blt', 'boba_drinks', 'pork_plate', 'omikase', 'pea_soup', 'bacon_board', 'pizza_muffins', 'pork_ramen', 'scramblers', 'samburger', 'brocollini', 'canelones', 'golds', 'chicken_skins', 'beefeater', 'kiki_burger', 'sanwich', 'barbeque_sauce', 'thai_spicy', 'spicy_ketchup', 'ploughman', 'protein_style', 'winnah', 'migas', 'belgium_waffle', 'katsu_sauce', 'medium_heat', 'vinegar_sauce', 'salmon_burger', 'siu_mai', 'stuffed_chicken', 'side_orders', 'mezzaluna', 'langostine', 'barbq', 'roast_duck', 'strawberry_salad', 'kitchen_sink', 'juicy_lucy', 'potato_gratin', 'sandwhichs', 'supremes', 'delite', 'mediterranean_salad', 'pizookies', 'phili', 'hotlinks', 'tomas', 'duck_salad', 'hocks', 'tritip', 'pittsburger', 'veal_marsala', 'zuccini', 'outlaw_burger', 'rosti', 'cilantro_rice', 'fico', 'panroast', 'mojave', 'pirogi', 'quail_eggs', 'marinera', 'potato_pizza', 'meatball_sliders', 'frickles', 'dimsums', 'chicken_karaage', 'food_spicy', 'mcdouble', 'sweet_potato', 'samps', 'deli_sandwich', 'chicken_caesar', 'bacon_pancakes', 'pesto_chicken', 'dawgs', 'beef_tartar', 'potato_cakes', 'piquante', 'bacon_sandwich', 'bottomless_fries', 'dill_pickle', 'hot_dog', 'ginger_sauce', 'veggie_rolls', 'potato_cake', 'turkey_wrap', 'mcgriddle', 'pomme_frites', 'cupcake_atm', 'bough', 'fesenjoon', 'rice_omelette', 'flower_soup', 'pompano', 'wimps', 'dollar_menu', 'calamari_appetizer', 'headwiches', 'ban_chan', 'seafood_chowder', 'crab_benedict', 'jowl', 'tra', 'hamburger_patty', 'bbblt', 'spatzel', 'seafood_crepe', 'lasagnas', 'feta_salad', 'rad_na', 'ravoli', 'tentacle', 'cevice', 'hotlink', 'kimchi_fries', 'greenbeans', 'dijonnaise', 'cheesesticks', 'lemon_sauce', 'chillie', 'house_burger', 'mcchicken', 'pumpkin_pancakes', 'pho_dip', 'katsu_chicken', 'burger_patties', 'bourbon_chicken', 'brocolini', 'gardenburger', 'mushroom_sandwich', 'giardinara', 'knockwurst', 'vareniki', 'bulgogi_fries', 'bluegill', 'cheesecurds', 'tartars', 'bastille', 'weiners', 'cheese_crisp', 'chicken_bites', 'sourcream', 'pork_hash', 'willis', 'turk', 'osso_buco', 'duck_wings', 'yukaejang', 'crab_angels', 'chanpon', 'shakshuka', 'turkey_burgers', 'maxi', 'apple_salad', 'dungeness_crab', 'morcilla', 'mushroom_pasta', 'veggie_salad', 'rashers', 'sandwitches', 'fritta', 'bbq_beans', 'omi', 'lobster_pho', 'breakfast_meal', 'squash_blossoms', 'maine_lobster', 'mushroom_appetizer', 'penne_vodka', 'turkey_sub', 'napolitano', 'perogis', 'cabbage_salad', 'casings', 'cowboy_beans', 'apple_sauce', 'steam_rice', 'belacan', 'dayboat', 'codfish', 'ruben_sandwich', 'benidict', 'veggie_roll', 'dover_sole', 'bacon_waffle', 'chicken_sandwhich', 'cheddar_grits', 'gluton', 'caesar_salads', 'cannonball', 'tbone', 'steak_skewers', 'tuna_carpaccio', 'pretzel_fondue', 'tomato_juice', 'lumpias', 'house_salsa', 'curry_rice', 'lemon_juice', 'rocoto', 'jerk_wings', 'beef_dip', 'cranburkey', 'crack_fries', 'pogo', 'flavor_combo', 'pommes_frites', 'jalopeno', 'cashu', 'pizzettes', 'chicken_picatta', 'breakfast_sausage', 'shogun_burger', 'fondeaux', 'quiche_lorraine', 'fish_sauce', 'jamacian', 'steamies', 'bronzino', 'cheesburger', 'spinach_ravioli', 'crispy_potatoes', 'blood_sausage', 'fradiavolo', 'pookie', 'something_spicy', 'pepporoni', 'chicken_w', 'lyonnaise_potatoes', 'vovo', 'egg_sandwiches', 'won_tons', 'baby_backs', 'broccolli', 'summer_menu', 'cheddar_burger', 'chowder_fries', 'babybacks', 'cheez_whiz', 'foood', 'pork_stew', 'griddles', 'khao_soi', 'portobello_mushroom', 'pinwheel', 'veggie_sub', 'cauliflower_soup', 'crack_sauce', 'pork_sausage', 'chichen', 'pika', 'roti_canai', 'corvina', 'blueberry_pie', 'tico', 'meat_patty', 'cane_sauce', 'gyro_combo', 'bisquit', 'porterhouse_steak', 'brown_sauce', 'tabasco_sauce', 'seafood_stew', 'potato_gnocchi', 'xxxl', 'queso_sauce', 'padak', 'verdi', 'homemade_sauces', 'genova', 'spicy_burger', 'cat_fish', 'emerald_chicken', 'dessert_crepes', 'bruschetta_board', 'vico', 'nomnom', 'turkey_melt', 'haba']

13 ['spread', 'animal', 'nacho', 'press', 'hoagie', 'focaccia', 'injera', 'onion_soup', 'cottage', 'whiz', 'laundry', 'crusts', 'wiz', 'tapenade', 'kalamata', 'knot', 'royale', 'lavosh', 'foccacia', 'sheep', 'bread_basket', 'cage', 'pizza_crust', 'crouton', 'pita_bread', 'chevre', 'apricots', 'yorkshire', 'roquefort', 'oaxaca', 'kaiser', 'lavash', 'daiya', 'pepperoncini', 'gratis', 'maytag', 'breadstick', 'pizza_sauce', 'lardo', 'honeycomb', 'goats', 'raclette', 'shmear', 'cheesy_bread', 'bialy', 'rye_bread', 'pretzel_bun', 'pizza_dough', 'radicchio', 'pesto_sauce', 'arugala', 'fruit_salad', 'crispy_crust', 'schmear', 'boursin', 'mozzarella_cheese', 'pimiento', 'cranberry_sauce', 'pine_nuts', 'pretzel_bread', 'wheat_bread', 'oaxacan', 'bialys', 'stilton', 'yogourt', 'apple_butter', 'beefsteak', 'feta_cheese', 'pepitas', 'violet', 'cornichons', 'tomato_salad', 'press_coffee', 'quince', 'fruit_cup', 'chihuahua', 'arugula_salad', 'popover', 'lettuces', 'dip_sandwich', 'iron_steak', 'semolina', 'romain', 'chedder', 'honey_butter', 'asadero', 'mozerella', 'crostinis', 'cottage_cheese', 'banana_peppers', 'dandelion', 'chartreuse', 'tomato_jam', 'marcona', 'maraschino', 'fruit_bowl', 'chuck_e', 'currants', 'gourmando', 'ricotta_cheese', 'curls', 'pepperoncinis', 'comb', 'dinner_rolls', 'plums', 'crustini', 'side_salads', 'cherry_tomatoes', 'shacksauce', 'merengue', 'cukes', 'oxtail_chili', 'cojita', 'evo', 'huile', 'parmasean', 'homeade', 'bacon_bits', 'strawberry_jam', 'breadbasket', 'ciabatta_bread', 'kumquat', 'homemade_bread', 'boquerones', 'grana', 'crostada', 'colby', 'kalamata_olives', 'camembert', 'blintz', 'bacon_mac', 'brioche_bun', 'corn_muffins', 'pomegranates', 'gremolata', 'romaine_lettuce', 'sorrel', 'juniper', 'artichoke_hearts', 'homemade_chips', 'gala', 'microgreens', 'banana_muffin', 'yogurt_parfait', 'mizithra', 'prunes', 'padano', 'tartlet', 'breadcrumb', 'crudite', 'ricotta_pancakes', 'portobellos',

'tillamook', 'tapenades', 'sabayon', 'argula', 'cornichon', 'corn_chips', 'taleggio', 'garbanzos', 'persimmon', 'craisins', 'kumquats', 'tlt', 'pretzel_roll', 'medjool', 'moz', 'lardon', 'mesclun', 'crust_pizzas', 'endives', 'crudites', 'avacodo', 'tomato_slices', 'avacados', 'humboldt', 'zaatar', 'bread_pizza', 'tortilla_strips', 'pretzel_sticks', 'broccoli_rabe', 'pumpkin_bread', 'atar', 'cabrales', 'creme_fraiche', 'mache', 'kiwis', 'cibatta', 'espuma', 'cut_bacon', 'panchetta', 'crutons', 'wilt', 'fuzzies', 'crisp_crust', 'bread_rolls', 'aspargus', 'overripe', 'pinenuts', 'cello', 'cuke', 'comte', 'heirloom_tomatoes', 'blk', 'liquer', 'chewy_crust', 'gaspacho', 'nectarine', 'pepperocini', 'tomato_sandwich', 'tapanade', 'luxardo', 'crostata', 'burata', 'water_chestnuts', 'bacon_egg', 'tangerines', 'clementine', 'lettuce_leaves', 'kohlrabi', 'gorgonzola_salad', 'buffalo_mozzarella', 'grissini', 'chuckie', 'gorganzola', 'crunchy_crust', 'sourdough_bread', 'blinis', 'huckleberries', 'lettus', 'watermelon_salad', 'fonduta', 'zabaglione', 'twigs', 'blini', 'tendrils', 'homemade_jam', 'banana_nut', 'basil_leaves', 'challah_bread']

16 ['appetizer', 'appetizers', 'meals', 'entrees', 'combo', 'platter', 'app', 'bruschetta', 'courses', 'apps', 'sampler', 'trio', 'starters', 'charcuterie', 'burrata', 'platters', 'mains', 'antipasti', 'arancini', 'bruchetta', 'salumi', 'misto', 'saganaki', 'brushetta', 'spinach_dip', 'antipasta', 'artichoke_dip', 'fritto', 'fritti', 'bruscetta']

19 ['food', 'service', 'everything', 'price', 'quality', 'prices', 'customer_service', 'presentation', 'pricing', 'quantity', 'freshness', 'preparation', 'services', 'food_quality', 'quality_food', 'execution', 'price_point', 'price_tag', 'quality_ingredients', 'preparations']

22 ['soda', 'sake', 'coke', 'alcohol', 'hookah', 'liquor', 'sodas', 'booze', 'pepsi', 'bud', 'cola', 'saki', 'pbr', 'soju', 'miller', 'sprite', 'cigar', 'coca', 'sapporo', 'corona', 'coors', 'cokes', 'cigars', 'saucer', 'goblet', 'kirin', 'mixer', 'draft_beer', 'mixers', 'sakes', 'bombers', 'hookahs', 'jug', 'equis', 'tobacco', 'shisha', 'bomber', 'carafes', 'modelo', 'house_wine', 'domestics', 'stein', 'coronas', 'fiji', 'budweiser', 'goblets', 'gasoline', 'heineken', 'tecate', 'teapot', 'mason_jar', 'ice_water', 'fishbowl', 'pabst', 'bud_light', 'flute', 'pacifico', 'perrier', 'negra', 'mason_jars', 'negro', 'liters', 'jugs', 'steins', 'canister', 'ice_cubes', 'plastic_cups', 'vats', 'chiller', 'water_cup', 'cerveza', 'tumbler', 'doses', 'flask', 'flutes', 'syringe', 'glass_bottles', 'shot_glass', 'jouet', 'dos_equis', 'gulps', 'voss', 'snifter', 'pint_glass', 'glass_bottle', 'evian', 'martini_glass', 'oberon']

30 ['tables', 'games', 'seats', 'chairs', 'booths', 'tvs', 'areas', 'cars', 'rooms', 'screen', 'screens', 'couches', 'stools', 'outlets', 'spaces', 'benches', 'televisions', 'lanes', 'sofas', 'entrances']

33 ['bathroom', 'machine', 'fountain', 'restroom', 'bathrooms', 'machines', 'floors', 'restrooms', 'toilet', 'soap', 'drip', 'pump', 'towels', 'dispenser', 'spill', 'pumps', 'stalls', 'spilling', 'blender', 'spray', 'nitrogen', 'filter', 'dishwasher', 'coals', 'storage', 'skim', 'dispensers', 'squirt', 'sinks', 'toilets', 'repair', 'vacuum', 'surfaces', 'spills', 'flush', 'wipes', 'urinal', 'washroom', 'sweep', 'juicer', 'propane', 'rinse', 'paper_towels', 'fluid', 'scrub', 'hose', 'lotion', 'mens', 'broom', 'toilet_paper', 'unisex', 'cleaners', 'ladies_room', 'sanitizer', 'toothpaste', 'locker', 'conditioner', 'liquids', 'refrigeration', 'pepto', 'faucet', 'naps', 'shampoo', 'circulation', 'mouthwash', 'condensation', 'freshener', 'filtration', 'bismol']

42 ['habanero', 'bbq_sauce', 'poppers', 'marinade', 'ghost', 'jalepeno', 'tomato_sauce', 'cream_sauce', 'peanut_sauce', 'dot', 'molasses', 'marinara_sauce', 'dipping_sauce', 'spicy_sauce', 'chili_sauce', 'popper', 'fiery', 'suicide', 'meat_sauce', 'sause', 'buffalo_sauce', 'inferno', 'sea_salt', 'teriyaki_sauce', 'tartar_sauce', 'franks', 'honey_mustard', 'bleu_cheese', 'tzatziki_sauce', 'harissa', 'curry_sauce', 'gyro_meat', 'giardiniera', 'tzaziki', 'serranos', 'nuoc', 'habenero', 'catsup', 'wine_sauce', 'butter_sauce', 'aoili', 'mam', 'barbecue_sauce', 'cocktail_sauce', 'wall_place', 'habaneros', 'fry_sauce', 'wasabe', 'soya', 'tobasco', 'taziki', 'togarashi', 'pepper_sauce', 'eel_sauce', 'mustard_sauce', 'marinades', 'yogurt_sauce', 'steak_sauce', 'sesame_seeds', 'crunchies', 'mayonaise', 'sissy', 'achiote', 'fresno', 'oli', 'pepper_shrimp', 'pablano', 'tarter_sauce', 'worcestershire', 'pepper_flakes', 'blue_cheese', 'house_sauce', 'tamari', 'cincy', 'corns ', 'spicy_mayo', 'chipotle_sauce', 'caraway', 'gochujang', 'berbere', 'amarillo', 'sumac', 'poblanos', 'drenching', 'spicy_mustard', 'tatziki', 'hormel', 'ponzu_sauce', 'guajillo', 'malt_vinegar', 'garlic_sauce', 'giardinera', 'mexicali', 'chili_oil', 'tomatillos', 'rellanos', 'pasilla', 'chili_flakes', 'hoison', 'pepper_chicken', 'cham', 'ranch_sauce', 'bell_pepper', 'habanero_salsa', 'habanero_sauce', 'chili_paste', 'shichimi', 'chili_peppers', 'chillis', 'ortega', 'duck_sauce', 'mayo_sauce', 'tzatiki', 'braising', 'cilantro_sauce', 'borracho', 'sriracha_sauce', 'padron', 'sriacha', 'sirracha', 'horsey', 'mirin', 'wall_places', 'hoisin_sauce', 'pepper_wings', 'aioli_sauce', 'suace', 'wilbur', 'proprietary', 'homemade_ranch',

'type_sauce', 'hoisen', 'soysauce', 'pastes', 'brimstone', 'pepper_shakers', 'kikkoman', 'pepper_squid', 'shisito']

44 ['cheese', 'soup', 'corn', 'calamari', 'spinach', 'mushroom', 'eggplant', 'asparagus', 'bisque', 'artichoke', 'chowder', 'zucchini', 'squash', 'cauliflower', 'polenta', 'brussel', 'mash', 'puree', 'brussels', 'butternut', 'fritters', 'skins', 'portobello', 'leek', 'croquettes', 'crisps', 'portabello', 'portabella', 'yukon', 'porcini', 'croquette', 'chard']

47 ['cartel', 'java', 'toddy', 'boba_tea', 'roaster', 'bingsoo', 'chai_latte', 'mothership', 'cappucino', 'roasters', 'commonplace', 'yoghurt', 'coffe', 'chai_tea', 'frappuccino', 'tango', 'intelligentsia', 'tea_latte', 'vanilla_latte', 'cortado', 'pitaya', 'ice_coffee', 'bobo', 'chocolate_milk', 'stumptown', 'kulfi', 'apple_juice', 'soymilk', 'nespresso', 'chemex', 'coffee_beans', 'breve', 'shave_ice', 'columbian', 'wheatgrass', 'raspado', 'cold_brew', 'lavazza', 'macha', 'cuppa', 'paleta', 'coffee_drink', 'caramel_macchiato', 'vera', 'pinapple', 'frappes', 'soy_milk', 'sago', 'gogo', 'chamomile', 'fortune_cookies', 'caramel_latte', 'peach_tea', 'frapp', 'mint_tea', 'chrysanthemum', 'cocomo', 'annihilator', 'jasmine_tea', 'gulab_jamun', 'fufu', 'clamato', 'trenta', 'superfood', 'siphon', 'hainan_chicken', 'malasada', 'oro', 'fountain_drinks', 'miel', 'capuccino', 'grapefruit_juice', 'chamoy', 'finders', 'sugarcane', 'rooibos', 'goji', 'gatorade', 'macchiatos', 'nutella_latte', 'drip_coffee', 'watermelon_juice', 'tea_boba', 'water_ice', 'chamango', 'coca_cola', 'longan', 'peets', 'moka', 'tea_selection', 'lavendar', 'tazza', 'lime_juice', 'iss', 'frappucino', 'sherbert', 'ful', 'pop_rocks', 'wintermelon', 'gratuit', 'soursop', 'honey_boba', 'buko', 'calamansi', 'aeropress', 'dreamsicle', 'mui', 'starburst', 'boba_drink', 'macapuno', 'lunchbox', 'carrot_juice', 'mango_smoothie', 'cake_batter', 'chica', 'roastery', 'powerade', 'snapple', 'fruity_pebbles', 'dirty_chai', 'protien', 'lassis', 'horchatas', 'lychees', 'paan', 'paradiso', 'tap_water', 'litchi', 'brew_coffee', 'fruit_juice', 'water_bottle', 'guatemalan', 'coke_products', 'soy_latte', 'butter_pecan', 'coolata', 'vanille', 'shasta', 'fluffy_snow', 'water_bottles', 'cochata', 'cortadito', 'caramelizer', 'jasmin', 'moshi', 'osmanthus', 'buttercup', 'banane', 'fountain_soda', 'mocha_latte', 'cadillac_margarita', 'coconut_juice', 'jimmies', 'ice_machine', 'lilikoi', 'house_coffee', 'sandia', 'lychee_martini', 'machiatto', 'earl_grey', 'limeades', 'energizer', 'fruit_smoothie', 'ranchers', 'mint_leaves', 'orchata', 'fresa', 'bingsu', 'patbingsu', 'coolatta', 'mano', 'guatemala', 'agar', 'fraps', 'coke_zero', 'bobba', 'watermelons', 'tea_refills', 'slushee', 'espresso_machine', 'brewski', 'pitaya_bowl',

'sumatra', 'agua_fresca', 'pepsi_products', 'malassadas', 'skittles', 'ibc', 'shrimp_cocktails', 'cherry_coke', 'penicillin', 'butterbeer', 'roca', 'strawberry_smoothie', 'cappucinos', 'homemade_lemonade', 'coffee_taste', 'vernors', 'muesli', 'macadamia_nuts', 'pikes', 'cherry_limeade', 'cup_o', 'voluto', 'cofee', 'ancora', 'cervezas', 'chocolate_hazelnut', 'cucumber_martini', 'watermelon_lemonade', 'coffee_machine', 'kirkland', 'fruit_punch', 'peach_bellini', 'kefir', 'egg_custard', 'gulaman', 'lassie', 'orange_slices', 'soda_refills', 'slurpees', 'mango_juice', 'greenie', 'mint_lemonade', 'coconut_water', 'flaxseed', 'specialty_drink', 'pourover', 'food_cold', 'avocado_smoothie', 'sarsaparilla', 'nama', 'cheesecake_bites', 'teapresso', 'caramel_apple', 'plum_wine', 'keurig', 'carmelizer', 'folgers', 'pog', 'butterfingers', 'dragonfruit', 'tazo', 'buttercrunch', 'mint_chocolate']

50 ['keg', 'pinot', 'chardonnay', 'blanc', 'cabernet', 'merlot', 'vino', 'malbec', 'cork', 'prosecco', 'chianti', 'sauvignon', 'winery', 'moscato', 'grigio', 'burgundy', 'zin', 'zinfandel', 'syrah', 'ros', 'bordeaux', 'chateau', 'decanter', 'shiraz', 'reisling', 'cava', 'cellars', 'pinot_noir', 'sangiovese', 'somm', 'wineries', 'barolo', 'steward', 'hennessy', 'varietal', 'vineyard', 'tempranillo', 'amarone', 'helix', 'magnum', 'rhone', 'macallan', 'gris', 'sauv', 'pinot_grigio', 'vineyards', 'pappy', 'brut', 'tawny', 'chablis', 'litre', 'barbera', 'rioja', 'grenache', 'sauvignon_blanc', 'sav', 'prisoner', 'meritage', 'woodford', 'muscat', 'tokaji', 'spectator', 'cakebread', 'kendall', 'sancerre', 'sauternes', 'albarino', 'franc', 'remy', 'stag', 'clicquot', 'viognier', 'montepulciano']

51 ['beer', 'wine', 'beers', 'tap', 'sample', 'wines', 'bottles', 'craft', 'samples', 'tour', 'draft', 'shots', 'brews', 'rounds', 'taps', 'pitchers', 'flights', 'drafts', 'cans', 'spirits', 'pints', 'tastings', 'drink_specials', 'craft_beer', 'craft_beers', 'microbrews', 'ciders', 'kegs', 'libations', 'draught', 'tequilas', 'liquors', 'imports', 'whiskeys', 'crafts', 'vodkas', 'microbrew', 'tours', 'draft_beers', 'harp', 'beer_selections', 'beer_choices', 'beer_specials', 'food_specials', 'bourbons', 'seasonals', 'lagers', 'porters', 'varietals', 'beer_options', 'belgians', 'craft_brews', 'scotches', 'micros', 'car_bombs']

53 ['garlic', 'truffle', 'olive', 'parmesan', 'jam', 'rosemary', 'vinaigrette', 'hollandaise', 'herb', 'sage', 'reduction', 'garnish', 'chutney', 'peppercorn', 'parsley', 'anchovies', 'compote', 'chimichurri', 'curd', 'crema', 'brandy', 'marmalade', 'paprika', 'sherry', 'demi', 'bearnaise', 'parmigiano', 'romano', 'remoulade', 'anchovy', 'cayenne', 'cognac', 'anise', 'bechamel', 'caper', 'cotija', 'parmesean', 'tarragon', 'evoo', 'nutmeg', 'shallot', 'tarter', 'bernaise', 'cloves',

'cardamom', 'glace', 'gelee', 'peppercorns', 'meyer', 'bordelaise', 'vinagrette', 'sprig', 'coulis', 'romesco', 'mignonette', 'vinegarette', 'emulsion', 'clove', 'mornay', 'gastrique', 'vinegrette', 'allspice']

60 ['tea', 'boba', 'juice', 'smoothie', 'snow', 'lemonade', 'teas', 'watermelon', 'taro', 'bubble', 'mochi', 'horchata', 'lychee', 'milk_tea', 'pomegranate', 'jasmine', 'grapefruit', 'lavender', 'melon', 'matcha', 'guava', 'slush', 'orange_juice', 'ice_tea', 'tapioca', 'kiwi', 'cantaloupe', 'durian', 'fresca', 'colada', 'pina', 'oolong', 'hibiscus', 'honeydew', 'limeade', 'frescas', 'pandan', 'melons']

71 ['wall', 'walls', 'lighting', 'lights', 'color', 'ceiling', 'colors', 'flowers', 'tree', 'candle', 'leather', 'trees', 'ceilings', 'paint', 'candles', 'carpet', 'cloth', 'plants', 'marble', 'accents', 'chandeliers', 'fixtures', 'tile', 'tablecloths', 'chandelier', 'cloths', 'shades', 'vinyl', 'tones', 'linen', 'wallpaper', 'tiles', 'graffiti', 'tablecloth', 'fabric', 'paneling', 'granite', 'hardwood', 'flags', 'linens', 'statues', 'brass', 'bulbs', 'beams', 'flooring', 'panels', 'lanterns', 'panel', 'bulb', 'drapes', 'carpets', 'chrome', 'columns', 'carpeting', 'pillars', 'upholstery']

74 ['dishes', 'things', 'items', 'options', 'stuff', 'choices', 'foods', 'selections', 'products', 'product', 'fare', 'offerings', 'goods', 'goodies', 'menu_items', 'staples', 'classics', 'specialties', 'brands', 'alternatives', 'food_items', 'food_options', 'menu_options', 'food_choices', 'suspects']

82 ['sugar', 'syrup', 'protein', 'power', 'caffeine', 'fuel', 'creamer', 'syrups', 'nutrition', 'supplements', 'boost', 'kombucha', 'medicine', 'sweetener', 'oats', 'fructose', 'yogurts', 'medication', 'surgery', 'vitamins', 'fats', 'splenda', 'flu', 'vitamin', 'fever', 'sugars', 'whey', 'stevia', 'meds', 'flavorings', 'alkaline', 'refresher', 'milks', 'wellness', 'additives', 'clover', 'pills', 'pregnancy', 'fiber', 'creamers', 'mineral', 'detox', 'sweeteners', 'elixir', 'powders', 'pill', 'nutrients', 'healing', 'cleanses']

85 ['pay', 'tip', 'charge', 'cost', 'bucks', 'dollars', 'dollar', 'tax', 'tips', 'cents', 'costs', 'gratuity', 'fee', 'charges', 'cent', 'taxes', 'fees']

88 ['shrimp', 'crab', 'lobster', 'salmon', 'seafood', 'tuna', 'oysters', 'scallops', 'mussels', 'mahi', 'octopus', 'clams', 'bass', 'cod', 'scallop', 'clam', 'squid', 'oyster', 'halibut', 'prawns', 'trout', 'shrimps', 'tilapia', 'snapper', 'sea_bass', 'shellfish', 'swordfish', 'seabass', 'urchin', 'prawn', 'muscles', 'grouper', 'mussel']

91 ['potatoes', 'potato', 'slaw', 'catfish', 'coleslaw', 'cornbread', 'cole', 'okra', 'potato_fries', 'plantains', 'puppies', 'potato_salad', 'collard', 'yams', 'yucca', 'plantain', 'potatoe', 'bangers', 'corn_bread', 'collards', 'potatos', 'hushpuppies', 'yuca', 'hush_puppies', 'cole_slaw', 'taters', 'collard_greens', 'sweet_potatoes', 'mish']

92 ['veggies', 'lettuce', 'mushrooms', 'vegetables', 'greens', 'broccoli', 'cabbage', 'carrots', 'peas', 'peanuts', 'sauteed', 'celery', 'saut', 'grass', 'cucumbers', 'iceberg', 'romaine', 'artichokes', 'radish', 'capers', 'pea', 'choy', 'bamboo', 'bok', 'scallions', 'pur', 'scallion', 'daikon', 'chives', 'leeks', 'chive', 'shoots', 'jicama', 'radishes', 'shallots', 'shiitake', 'enoki', 'shitake', 'leafy', 'chestnuts', 'iceburg', 'choi', 'iceberg_lettuce', 'wilty']

98 ['adovada', 'frybread', 'asada_fries', 'asada_tacos', 'fry_bread', 'chile_relleno', 'frijoles', 'picante', 'coconut_shrimp', 'gordita', 'vampiro', 'charro', 'birria', 'chicken_taco', 'frito', 'street_corn', 'tamal', 'sopapilla', 'gauc', 'tomate', 'sope', 'gran', 'bistec', 'reina', 'argentine', 'gambas', 'flauta', 'yucatan', 'carnita', 'papusas', 'tripa', 'adobado', 'fritos', 'nogada', 'basque', 'mexicana', 'taquito', 'buche', 'rojo', 'chili_relleno', 'tacu', 'california_burrito', 'huitlacoche', 'milanesa', 'churrasco', 'ceviches', 'guacomole', 'pabellon', 'queso_dip', 'pollo_fundido', 'zucca', 'nopales', 'shrimp_burrito', 'combo_plate', 'ensalada', 'anticuchos', 'elotes', 'sampler_platter', 'roja', 'suadero', 'quac', 'chili_verde', 'cerviche', 'stuffed_mushrooms', 'chile_verde', 'lobster_tacos', 'pulpo', 'quesa', 'spicy_salsa', 'plater', 'verdes', 'salsa_verde', 'brasa', 'limon', 'pepiada', 'carna', 'mango_salsa', 'pasteles', 'curtido', 'papusa', 'chicharrones', 'picadillo', 'breakfast_tacos', 'mula', 'rellena', 'huaraches', 'tierra', 'veracruz', 'tostado', 'fundito', 'huarache', 'rajas', 'pil', 'gandules', 'habla', 'cubana', 'loup', 'chiliquiles', 'queso_fundido', 'suizas', 'platanos', 'adovado', 'soyrizo', 'steak_quesadilla', 'street_taco', 'sofritas', 'quesidilla', 'mulas', 'lomo_saltado', 'carnitas_tacos', 'chalupa', 'crane', 'suiza', 'escabeche', 'combination_plate', 'huancaina', 'pork_taco', 'chilequiles', 'arbol', 'sofrito', 'vampiros', 'crab_dip', 'feijoada', 'quesedilla', 'bandeja', 'veggie_tacos', 'jalea', 'parilla', 'chicharon', 'quacamole', 'cachapa', 'mole_sauce', 'parrillada', 'loroco', 'empenadas', 'asada_nachos', 'paisa', 'cromesquis', 'corn_salsa', 'espanola', 'bacalao', 'con_pollo', 'meat_combo', 'pasole', 'shrimp_fajitas', 'salas', 'bellota', 'mixto', 'chicken_chimichanga', 'chicken_enchilada', 'morue', 'buritto', 'verde_sauce', 'picado', 'sopas', 'homemade_salsa', 'chaufa', 'enchilladas', 'motulenos', 'mulitas', 'asada_burritos', 'bean_burrito', 'con_queso', 'fresas', 'burritto', 'pork_carnitas', 'steak_taco', 'calabacitas',

'seco', 'combo_platter', 'valenciana', 'popusas', 'carne_adovada', 'naco', 'torts', 'mini_tacos', 'campechana', 'spinach_enchiladas', 'carnitas_burrito', 'cabra', 'huevo', 'cerdo', 'conchinita', 'parrilla', 'burito', 'rellenas', 'hongos', 'moros', 'coctel', 'chorizo_tacos', 'cachapas', 'asada_torta', 'boricua', 'tomatillo_salsa', 'veggie_bowl', 'arrachera', 'aguachile', 'egg_burrito', 'juevos', 'sesos', 'machacha', 'camerones', 'enchillada', 'taco_combo', 'champlain', 'chili_rellenos', 'wet_burrito', 'asada_quesadilla', 'causa', 'catalan', 'molcajetes', 'cruda', 'lingua', 'ranchera', 'pastelitos', 'mojarra', 'chorizos', 'revuelta', 'caille', 'alpastor', 'mixta', 'aguachiles', 'maiz', 'moles', 'quesodilla', 'nuevo', 'tampiquena', 'platano', 'tongue_tacos', 'seafood_enchiladas', 'asana', 'anticucho', 'brisket_tacos', 'steak_bowl', 'revueltas', 'quesidillas', 'guisada', 'ahogada', 'tostados', 'quesedillas', 'tomatillo_sauce', 'mini_chimis', 'dilla', 'veggie_quesadilla', 'dorados', 'culichi', 'barbacoa_tacos', 'chile_rellenos', 'vallarta', 'catalana', 'chicken_quesadillas', 'chicken_chimi', 'steak_nachos', 'alcapurrias', 'pierna', 'chimichunga']

104 ['sandwhich', 'chicken_sandwich', 'banh_mi', 'chicken_salad', 'caesar_salad', 'side_salad', 'bobbie', 'pork_chop', 'bahn_mi', 'pork_sandwich', 'montagu', 'chix', 'sammich', 'veggie_burger', 'seitan', 'huli', 'kiki', 'steak_sandwich', 'ronin', 'pork_chops', 'miyagi', 'jackfruit', 'sammie', 'jerk_chicken', 'chicken_breast', 'tempeh', 'gardein', 'house_salad', 'turkey_burger', 'kale_salad', 'ground_beef', 'bbq_chicken', 'cobb_salad', 'club_sandwich', 'capastrami', 'beet_salad', 'pasta_salad', 'bologna', 'chicken_pizza', 'pot_roast', 'lamb_chops', 'pastrami_sandwich', 'chicken_wrap', 'cuban_sandwich', 'spinach_salad', 'beef_brisket', 'tuna_melt', 'pork_tenderloin', 'wedge_salad', 'steak_salad', 'vege', 'bacon_cheeseburger', 'pork_tacos', 'chicago_dog', 'tuna_salad', 'skirt_steak', 'meat_sandwich', 'vegi', 'brisket_sandwich', 'montague', 'veggie_sandwich', 'bacon_burger', 'tuna_sandwich', 'chili_dog', 'cesar_salad', 'oggie', 'buta', 'salmon_salad', 'turkey_club', 'suey', 'kobe_burger', 'bison_burger', 'mushroom_burger', 'ceasar_salad', 'quinoa_salad', 'chx', 'capistrami', 'chicken_plate', 'pork_sliders', 'bbq_sandwich', 'bbq_burger', 'chicken_dinner', 'kitchen_burger', 'vegie', 'chicken_benedict', 'chiken', 'pork_sandwiches', 'meat_sandwiches', 'chicken_combo', 'pork_nachos', 'meat_poutine']

106 ['whiskey', 'root', 'rum', 'bourbon', 'blood', 'manhattan', 'malt', 'float', 'kool', 'blossom', 'agave', 'goose', 'absinthe', 'root_beer', 'whisky', 'moonshine', 'jameson', 'jamaica', 'gelati', 'rootbeer', 'nectar', 'liqueur', 'bitters', 'cadillac', 'tangerine', 'michelada',

'fireball', 'cosmos', 'passionfruit', 'daiquiri', 'sherbet', 'pom', 'fizz', 'malibu', 'shochu', 'gimlet', 'pisco', 'caipirinha', 'creamsicle', 'negroni', 'patch', 'chambord', 'vermouth', 'grasshopper', 'collins', 'tini', 'ritas', 'framboise', 'strawberry_lemonade', 'passion_fruit', 'moscow_mule', 'julep', 'slushie', 'midori', 'rasberry', 'house_margarita', 'mezcal', 'jager', 'slurpee', 'stoli', 'inca', 'sazerac', 'sidecar', 'tonics', 'grappa', 'elderflower', 'bacardi', 'koolaid', 'julius', 'greyhound', 'cran', 'paloma', 'campari', 'shandy', 'manhattans', 'cuervo', 'tamarindo', 'ketel', 'mai_tai', 'lemoncello', 'absolut', 'seltzer', 'cream_soda', 'redbull', 'belvedere', 'chaser', 'ginger_beer', 'fanta', 'schnapps', 'aviation', 'grenadine', 'chicha', 'hendricks', 'kool_aid', 'ginger_ale', 'saketini', 'smirnoff', 'aide', 'skyy', 'jarritos', 'tequilla', 'screwdriver', 'nigori', 'sambuca', 'blood_orange', 'ouzo', 'antioxidant', 'orenji', 'dirty_martini', 'pina_colada', 'cointreau', 'apple_cider', 'frangria', 'titos', 'morada', 'mango_margarita', 'strawberry_margarita', 'pomegranite', 'club_soda', 'everclear', 'kola', 'daiquiris', 'painkiller', 'floater', 'spritz', 'reposado', 'pear_margarita', 'aperol', 'irishman', 'spritzer', 'tanqueray', 'coladas', 'anejo', 'effen', 'puss', 'fernet', 'ciroc', 'vesper', 'cruzan', 'cachaca', 'chopin', 'bramble']

108 ['mein', 'bao', 'banh', 'cha', 'hue', 'tai', 'bahn', 'shu', 'siu', 'moo', 'nuong', 'har', 'mien', 'xiao', 'nem', 'biet', 'gio', 'dac', 'gow', 'bun_bo', 'vang', 'cuon', 'sui', 'jenni', 'goi', 'xeo', 'gau', 'khai', 'rieu', 'kho', 'canh', 'tieu', 'hoan', 'luc', 'lac', 'xao', 'vien', 'hoa', 'sua', 'quang', 'chua']

110 ['spring', 'caesar', 'cucumber', 'kale', 'arugula', 'caprese', 'seaweed', 'beet', 'beets', 'quinoa', 'wedge', 'papaya', 'grape', 'cous', 'fennel', 'cobb', 'couscous', 'heirloom', 'cesar', 'lentils', 'ceasar', 'orzo', 'chickpeas', 'endive', 'pureed', 'farro', 'watercress', 'frisee', 'ceaser', 'caeser']

111 ['orange', 'coconut', 'honey', 'strawberry', 'mango', 'apple', 'peanut', 'pineapple', 'soy', 'lime', 'ginger', 'cinnamon', 'pumpkin', 'mint', 'maple', 'sesame', 'cherry', 'raspberry', 'blueberry', 'peach', 'pear', 'pecan', 'jelly', 'berry', 'citrus', 'walnut', 'cranberry', 'yuzu', 'buttermilk', 'lemongrass', 'macadamia', 'cashew', 'blackberry', 'plum', 'apricot', 'cumin', 'tamarind', 'rhubarb', 'coriander']

112 ['curry', 'pad', 'teriyaki', 'jerk', 'pad_thai', 'drunken', 'panang', 'tao', 'satay', 'pao', 'tso', 'foo', 'orange_chicken', 'yakitori', 'karaage', 'larb', 'chow_mein', 'agedashi', 'wonton_soup', 'yaki', 'pow', 'stir_fry', 'massaman', 'sesame_chicken', 'penang',

'chicken_curry', 'mochiko', 'papaya_salad', 'consensus', 'panang_curry', 'chow_fun', 'laksa', 'kushi', 'hainan', 'pao_chicken', 'crab_rangoon', 'kung_pao', 'mapo', 'kara', 'terriyaki', 'karage', 'egg_foo', 'teriyaki_bowl', 'teryaki', 'tsao', 'hainanese', 'masaman', 'tsos', 'lomein', 'silken', 'tso_chicken', 'agadashi']

116 ['cap', 'waygu', 'snails', 'kobe_beef', 'fishes', 'stake', 'poultry', 'sardines', 'delicacy', 'beef_wellington', 'gizzards', 'vide', 'ribeyes', 'snail', 'sardine', 'intestines', 'livers', 'picanha', 'intestine', 'fowl', 'sea_food', 'cutlets', 'shanks', 'duck_confit', 'rotisserie_chicken', 'frittes', 'chateaubriand', 'bavette', 'wagu', 'swine', 'porkchop', 'frite', 'strip_steak', 'medallion', 'shoo', 'beef_carpaccio', 'riblets', 'beef_tongue', 'hanger_steak', 'kitfo', 'veal_chop', 'pork_shoulder', 'beef_burger', 'seafoods', 'eye_steak', 'minion', 'species', 'steak_frites', 'porkchops', 'flatiron', 'boyardee', 'offal', 'half_chicken', 'piglet', 'steak_tartare', 'tenderloins', 'crustacean', 'wagyu_beef', 'ribeye_steak', 'flank_steak', 'beef_tenderloin', 'pork_loin', 'meat_loaf', 'koro', 'crustaceans', 'lamb_shank', 'skinless', 'jumbalaya', 'beef_tartare', 'mignons', 'fillet_mignon', 'calf', 'beef_sliders', 'lambs', 'seafood_tower', 'babyback', 'sea_scallops', 'hamburg', 'burger_medium', 'meat_platter', 'striploin', 'drumettes', 'minon', 'rib_cap', 'medium_well', 'sirlion', 'sirloins', 'rib_steak', 'sirloin_steak', 'crib', 'mignon_sliders', 'tournedos', 'steak_medium']

119 ['tater', 'chicken_wings', 'chicken_fingers', 'shoe', 'shoestring', 'kettle', 'crinkle', 'chicken_tenders', 'mozzarella_sticks', 'kit', 'chicken_strips', 'boneless_wings', 'reubens', 'tator', 'xxx', 'hnj', 'beer_cheese', 'potato_pancakes', 'potato_skins', 'smashfries', 'doritos', 'buffalo_wings', 'dippers', 'cheese_curds', 'corn_dogs', 'grinders', 'faygo', 'potato_tots', 'chili_fries', 'stix', 'chicken_nachos', 'onion_strings', 'potato_hash', 'chicken_sandwiches', 'flats', 'pokey', 'fondu', 'zucchini_fries', 'corndogs', 'ore', 'coneys', 'wingstop', 'potato_wedges', 'tator_tots', 'smash_fries', 'feta_fries', 'zingers', 'bbq_wings', 'veggie_burgers', 'style_fries', 'fondues', 'twc', 'cravin', 'blasts', 'strombolis', 'onion_straws', 'potato_pie', 'blazin', 'ida', 'ruffles', 'jalapeno_poppers', 'crispy_fries', 'wing_sauce', 'cheese_fries', 'dings', 'tostitos', 'steak_sandwiches', 'potato_casserole', 'pretzel_bites', 'honey_bbq', 'lei', 'shoestring_fries', 'chicken_sliders', 'style_burger', 'crinkle_fries', 'tatter', 'potatoe_fries', 'armadillo', 'ragin', 'chicken_dip', 'daytona', 'chos', 'potatoes_fries', 'tooths', 'potato_mash', 'medium_wings', 'chicken_rolls', 'parm_fries', 'chicken_wraps']

121 ['beans', 'chili', 'onions', 'tomato', 'pepper', 'avocado', 'tomatoes', 'peppers', 'bean', 'basil', 'chile', 'pesto', 'cilantro', 'olives', 'jalapenos', 'chilli', 'chiles', 'chilis', 'chilies', 'jalepenos', 'bell_peppers', 'tomatos', 'shrooms', 'tomatoe', 'bean_sprouts', 'chillies']

125 ['tacos', 'taco', 'burrito', 'nachos', 'carne', 'enchiladas', 'burritos', 'quesadilla', 'enchilada', 'fajitas', 'carnitas', 'pastor', 'tamales', 'torta', 'chimichanga', 'quesadillas', 'burro', 'tostada', 'pupusas', 'fajita', 'arepas', 'menudo', 'adobada', 'barbacoa', 'lengua', 'taquitos', 'tortas', 'chimi', 'machaca', 'carne_asada', 'flautas', 'street_tacos', 'tostadas', 'pozole', 'cochinita', 'chicken_tacos', 'chimichangas', 'shrimp_tacos', 'posole', 'cabeza', 'sopes', 'tortilla_soup', 'pibil', 'burros', 'taco_salad', 'chicken_burrito', 'tinga', 'beef_tacos', 'fish_tacos', 'chimis', 'chicken_quesadilla', 'gorditas', 'chicken_enchiladas', 'beef_taco', 'pastor_tacos', 'steak_tacos', 'chicken_fajitas', 'burrito_bowl', 'veggie_burrito', 'steak_burrito', 'steak_fajitas', 'beef_burrito', 'mahi_tacos', 'beef_chimichanga', 'beef_enchiladas']

130 ['staff', 'owner', 'chef', 'servers', 'employees', 'owners', 'bartenders', 'waiters', 'chefs', 'waitresses', 'cooks', 'waitstaff', 'managers', 'hostesses', 'baristas', 'cashiers', 'hosts', 'wait_staff', 'staffs', 'sushi_chefs', 'bar_staff', 'service_staff', 'bar_tenders']

131 ['guy', 'man', 'girl', 'lady', 'cashier', 'employee', 'woman', 'workers', 'worker', 'gentleman', 'dude', 'barista', 'gal', 'clerk']

132 ['tar', 'dynamite', 'tako', 'sushis', 'butterfish', 'perch', 'spicy_tuna', 'tamago', 'saba', 'firecracker', 'crudo', 'whitefish', 'tuna_roll', 'california_roll', 'poki', 'sushi_roll', 'seaweed_salad', 'lomi', 'handroll', 'shrimp_tempura', 'lobster_rolls', 'pike', 'monk', 'shell_crab', 'sturgeon', 'king_crab', 'limu', 'rainbow_roll', 'jellyfish', 'butterfly', 'crayfish', 'yellowfin', 'bluefin', 'nigiris', 'abalone', 'dragon_roll', 'natto', 'ahi_tuna', 'stone_crab', 'pok', 'california_rolls', 'tnt', 'nigri', 'sword', 'hand_roll', 'escolar', 'handrolls', 'tempura_roll', 'lobster_tails', 'tuna_tartare', 'broil', 'amberjack', 'inari', 'seafood_salad', 'twister', 'tare', 'salmon_roll', 'crab_claws', 'kamikaze', 'crag', 'langoustines', 'stone_crabs', 'gobo', 'onigiri', 'crablegs', 'goma', 'mussles', 'hand_rolls', 'spider_roll', 'mentaiko', 'sunomono', 'seviche', 'tuna_tartar', 'negi', 'chawanmushi', 'shimp', 'kimbap', 'osetra', 'cuttlefish', 'crunch_roll', 'cape', 'shashimi', 'crab_leg', 'cavier', 'lobstah', 'climax', 'tuna_burger', 'tuna_tacos', 'otoro', 'kani', 'volcano_roll', 'wakame', 'hirame', 'tuna_rolls', 'tiradito', 'sashimi_salad', 'anago', 'crab_salad', 'lobster_sauce', 'makis', 'steelhead', 'harami', 'crab_roll', 'miso_cod', 'naruto',

'misoyaki', 'hotate', 'sashimis', 'freshwater', 'kelp', 'chowders', 'gra', 'shirmp', 'geoduck', 'tsunami', 'haru', 'crispy_rice', 'tempera', 'chirashi_bowl', 'salmon_sashimi', 'nasu', 'neptune', 'mushi', 'ahi_poke', 'loch', 'stacie', 'shima', 'banzai', 'squid_salad', 'catalina', 'tuna_sashimi', 'swai', 'thermidor', 'almondine', 'ink_pasta', 'spiny', 'hama', 'tuna_poke', 'sea_urchin', 'surimi', 'jonah', 'toban', 'kusshi', 'crawdads', 'futomaki', 'mackeral', 'yamato', 'pollock', 'tiger_roll', 'maki_rolls', 'carpacio', 'wowie', 'scotia', 'quail_egg', 'duart', 'senorita', 'cocktail_shrimp', 'amandine', 'pollack', 'nagiri', 'eggless', 'ahi_salad', 'spicy_salmon', 'gumbos', 'poke_bowl', 'poke_salad', 'scallop_roll', 'poboys', 'geso', 'tar_tar', 'sockeye', 'tuna_steak', 'ishiyaki', 'specialty_roll', 'seafood_gumbo', 'ume', 'beef_roll', 'steampot', 'jala', 'temaki', 'matilda', 'shrimp_roll', 'yellowtail_sashimi', 'ngiri', 'langostino', 'tomago', 'battera', 'mirugai', 'shrip', 'salmon_nigiri', 'philadelphia_roll', 'eat_shrimp', 'rock_n', 'monger', 'chowda', 'sushimi', 'beef_tataki', 'grais', 'tekka', 'kohada', 'hapu', 'bama', 'basa', 'suzuki', 'tavarua', 'salmon_sushi', 'tootsy', 'grois', 'shad', 'shiromi', 'littleneck', 'tuna_appetizer']

139 ['popcorn', 'chip', 'cone', 'macaron', 'monkey', 'cotton', 'cones', 'mores', 'macaroon', 'oreos', 'concrete', 'icecream', 'blizzard', 'biscotti', 'smores', 'churro', 'turtle', 'hotcakes', 'lollipops', 'lollipop', 'eclair', 'hershey', 'cream_pie', 'snickers', 'mouse', 'reese', 'cinnamon_roll', 'apple_pie', 'cotton_candy', 'snickerdoodle', 'bailey', 'gingerbread', 'twinkie', 'chip_cookies', 'chocolate_chip', 'chip_cookie', 'flapjack', 'silk', 'coffee_cake', 'twinkies', 'flapjacks', 'affogato', 'chocolate_croissant', 'snicker', 'apple_fritter', 'nut_muffin', 'croque_madame', 'fosters', 'banana_bread', 'doodle', 'butterfinger', 'heath', 'smore', 'cookie_dough', 'blueberry_pancakes', 'protein_pancakes', 'waffle_cone', 'blueberry_muffin', 'nutella_crepe', 'reeses', 'chip_pancakes', 'chocolatine', 'buttermilk_pancakes', 'banana_pancakes', 'creme_pie', 'nut_muffins']

140 ['ingredients', 'toppings', 'sauces', 'meats', 'spices', 'cheeses', 'condiments', 'herbs', 'seasonings', 'extras', 'fillings', 'oils', 'dressings', 'condiment', 'fixings', 'proteins', 'ons', 'grains', 'accompaniments', 'garnishes', 'fixins', 'sommes', 'vinegars', 'accoutrements', 'fixin', 'salts', 'trimmings']

144 ['dog', 'dogs', 'poutine', 'pot', 'piping', 'sausages', 'schnitzel', 'brat', 'links', 'pierogies', 'sauerkraut', 'link', 'brats', 'pierogi', 'bratwurst', 'beef_sandwich', 'pierogis', 'kielbasa', 'kraut', 'perogies', 'beefs', 'beef_sandwiches']

145 ['belly', 'roast', 'kobe', 'ground', 'wagyu', 'tartare', 'chops', 'carpaccio', 'tenderloin', 'tartar', 'jerky', 'wellington', 'angus', 'shoulder', 'loin', 'shank', 'kalua', 'cheek', 'tataki', 'stroganoff', 'cheeks', 'kahlua', 'rinds', 'bourguignon']

152 ['moon', 'fin', 'ribbon', 'collar', 'taj', 'cordon', 'hound', 'adobe', 'cheese_sandwich', 'cheez', 'cheddar_cheese', 'chz', 'cheeze', 'cheese_sauce', 'jack_cheese', 'chesse', 'cheese_salad', 'cheese_burger', 'chees', 'chese', 'cheese_dip']

159 ['pizza', 'crust', 'pepperoni', 'dough', 'meatball', 'flatbread', 'calzone', 'margherita', 'stromboli', 'style_pizza', 'pep', 'crust_pizza', 'pepperoni_pizza', 'peperoni', 'mushroom_pizza', 'sausage_pizza']

161 ['carrot', 'oxtail', 'stew', 'base', 'lentil', 'casserole', 'saffron', 'barley', 'congee', 'basmati', 'tomato_soup', 'matzo', 'buckwheat', 'broccolini', 'porridge', 'brussel_sprouts', 'minestrone', 'pilaf', 'yam', 'brussels_sprouts', 'chickpea', 'spaetzle', 'chestnut', 'matzoh', 'albondigas', 'mash_potatoes', 'turnip', 'fideo', 'rabe', 'dashi', 'goulash', 'parsnip', 'mashed_potatoes', 'ragout', 'chanterelle', 'succotash', 'blossoms', 'brocolli', 'brunswick', 'soybean', 'rind', 'borscht', 'turnips', 'potato_soup', 'saute', 'bouillon', 'ginseng', 'fungus', 'pods', 'haystack', 'ball_soup', 'spatzle', 'lardons', 'eggplants', 'fagioli', 'parsnips', 'manila', 'bread_bowl', 'turmeric', 'plantation', 'steamers', 'bhaji', 'ramps', 'bloomin', 'kabocha', 'brocoli', 'lipton', 'kaffir', 'guanciale', 'cream_corn', 'bok_choy', 'julienne', 'spinich', 'roughy', 'krispies', 'half_sandwich', 'basil_soup', 'fingerling_potatoes', 'string_beans', 'matzah', 'veloute', 'kreplach', 'claypot', 'dishwater', 'squash_ravioli', 'fingerlings', 'butternut_squash', 'cremini', 'clay_pot', 'crispies', 'segments', 'egg_drop', 'krispie', 'maitake', 'squash_soup', 'avgolemono', 'takana', 'chanterelles', 'brandade', 'roni', 'galanga', 'bisque_soup', 'menma', 'crimini', 'figgy', 'mousseline', 'rutabaga', 'kulcha', 'celeriac', 'sauteed_mushrooms', 'gratinee', 'mashers', 'cipollini', 'cheddar_soup', 'sautee', 'beansprouts', 'peels', 'soybeans', 'trax', 'vidalia', 'faro', 'sunchoke', 'sunchokes', 'matza', 'kugel', 'galangal', 'kikurage', 'fricassee', 'crab_puff', 'chicken_noodle', 'matzo_ball', 'fiddleheads', 'salsify', 'fishcake', 'bokchoy', 'shimeji', 'arborio']

166 ['smoothies', 'juices', 'milkshakes', 'coffees', 'omelets', 'lattes', 'breakfasts', 'omelettes', 'sundaes', 'breakfast_burritos', 'skillets', 'benedicts', 'brunches', 'breakfast_sandwiches', 'cappuccinos', 'lemonades', 'malts', 'blizzards', 'mochas', 'scrambles', 'bottomless_mimosas', 'coffee_drinks', 'slushies', 'bellinis', 'slushes', 'bobas',

'custards', 'espressos', 'bloody_marys', 'milk_shakes', 'omlettes', 'americanos', 'concretes', 'milk_teas', 'bloodies', 'bloodys']

167 ['bread', 'toast', 'bun', 'bagel', 'patty', 'buns', 'croissant', 'muffin', 'wheat', 'patties', 'baguette', 'sourdough', 'brioche', 'rye', 'lox', 'cream_cheese', 'grain', 'scone', 'ciabatta']

172 ['manicotti', 'porchetta', 'lasagne', 'chicken_parm', 'bianca', 'pomodoro', 'delicioso', 'excellente', 'mortadella', 'bucatini', 'meatball_sub', 'capicola', 'primo', 'primavera', 'funghi', 'quattro', 'wiseguy', 'chicken_parmesan', 'chicken_marsala', 'meatball_sandwich', 'shrimp_scampi', 'orecchiette', 'pasta_dish', 'rapini', 'diavola', 'riserva', 'lobster_ravioli', 'gnudi', 'rustica', 'puttanesca', 'capellini', 'caponata', 'tartufo', 'calabrese', 'biancoverde', 'margharita', 'merguez', 'vodka_sauce', 'chicken_alfredo', 'rollatini', 'salsiccia', 'eno', 'soppressata', 'seafood_pasta', 'spaghettini', 'sopressata', 'fungi', 'formaggi', 'tortelli', 'spagetti', 'ossobuco', 'mushroom_risotto', 'burratta', 'pici', 'meat_balls', 'arrabiata', 'genoa', 'vongole', 'halloumi', 'toscana', 'salame', 'tortelloni', 'chicken_piccata', 'zuppa', 'canneloni', 'cavatappi', 'lobster_gnocchi', 'blanca', 'penne_pasta', 'amatriciana', 'ches', 'papardelle', 'olio', 'osso_bucco', 'vita', 'fusilli', 'coppa', 'stagioni', 'bowtie', 'formaggio', 'farfalle', 'buratta', 'shrimp_pasta', 'homemade_pasta', 'clam_sauce', 'chicken_parmigiana', 'romana', 'pizzette', 'oso', 'mushroom_sauce', 'fettuccine_alfredo', 'cotto', 'pasta_sauce', 'gemelli', 'spiedini', 'secondi', 'primi', 'raviolini', 'don_juan', 'pesto_pasta', 'pizze', 'mushroom_ravioli', 'manzo', 'bistecca', 'lobster_risotto', 'braciole', 'capricciosa', 'patate', 'pesce', 'toscano', 'aubergine', 'pescatore', 'rossa', 'rigatony', 'garganelli', 'bolognaise', 'parpadelle', 'rotini', 'seafood_risotto', 'arrabbiata', 'classico', 'santi', 'prosciutto_pizza', 'bocconcini', 'italiana', 'involtini', 'bresaola', 'quatro', 'principe', 'vitello', 'buca_di', 'sugo', 'polpette', 'sorrento', 'salata', 'aglio', 'brasato', 'scoglio', 'strozzapreti', 'fruitti', 'tagliolini', 'prociutto', 'brunello', 'mostaccioli', 'gamberi', 'notte', 'verdure', 'ossobucco', 'rasta', 'giorno', 'rosso', 'diavlo', 'allo', 'posto', 'genovese', 'fetuccini', 'zany', 'fiorentina', 'rustico', 'filetto', 'fornaio', 'brioni', 'pasto']

173 ['cocktails', 'cocktail', 'margarita', 'margaritas', 'sangria', 'martini', 'mary', 'vodka', 'tequila', 'mojito', 'martinis', 'gin', 'mule', 'mojitos', 'tonic', 'moscow', 'marys', 'bellini', 'margs', 'marg', 'sangrias', 'mules', 'micheladas', 'house_margaritas']

174 ['cream', 'creme', 'creams', 'cream_sandwich', 'burg', 'cream_shop', 'cream_sandwiches', 'cream_cone', 'cream_cake', 'cream_flavors', 'cream_dessert', 'cream_place', 'berg', 'cream_machine', 'cream_cones', 'cream_sundae', 'cream_parlor']

175 ['chicken', 'meat', 'fish', 'beef', 'pork', 'ribs', 'rib', 'brisket']

182 ['napkins', 'fork', 'spoon', 'silverware', 'knife', 'utensils', 'gloves', 'napkin', 'straw', 'forks', 'chopsticks', 'spoons', 'lid', 'towel', 'knives', 'rag', 'lids', 'tongs', 'cutlery', 'bib', 'glassware', 'ware', 'utensil', 'flatware', 'bibs']

183 ['wings', 'sides', 'steaks', 'bone', 'sliders', 'buffalo', 'fingers', 'strips', 'wing', 'breast', 'cuts', 'skewers', 'tenders', 'rack', 'leg', 'boneless', 'nuggets', 'skewer', 'rotisserie', 'filets', 'thigh', 'breasts', 'medallions', 'cutlet', 'thighs', 'fillets', 'drumsticks', 'drumstick']

184 ['donuts', 'desserts', 'cupcakes', 'gelato', 'cookies', 'bagels', 'crepes', 'pastries', 'macarons', 'doughnuts', 'muffins', 'sweets', 'macaroons', 'deserts', 'croissants', 'brownies', 'scones', 'tarts', 'sorbets', 'cheesecakes', 'gelatos', 'ice_creams', 'eclairs', 'caramels', 'confections']

185 ['mac', 'goat', 'curds', 'feta', 'macaroni', 'bleu', 'pimento', 'lobster_mac', 'mac_n', 'truffle_mac']

190 ['menu', 'list', 'beer_selection', 'lunch_menu', 'wine_selection', 'dinner_menu', 'drink_menu', 'salad_bar', 'salsa_bar', 'kids_menu', 'wine_list', 'breakfast_menu', 'food_selection', 'dessert_menu', 'food_menu', 'brunch_menu', 'beer_menu', 'drink_selection', 'menu_selection', 'cocktail_menu', 'bar_menu', 'wine_menu', 'menue', 'appetizer_menu', 'tapas_menu', 'sushi_menu', 'ayce_menu', 'draft_selection']

191 ['fruit', 'banana', 'strawberries', 'nuts', 'bananas', 'fruits', 'apples', 'nut', 'berries', 'seeds', 'leaf', 'almonds', 'walnuts', 'blueberries', 'pecans', 'seed', 'grapes', 'cherries', 'avocados', 'raisins', 'limes', 'lemons', 'peaches', 'oranges', 'cranberries', 'pistachios', 'pears', 'raspberries', 'cashews', 'pineapples', 'hazelnuts', 'blackberries', 'mangoes', 'mangos']

194 ['israeli', 'souper', 'tagine', 'nicoise', 'chicken_shawarma', 'stetson', 'fattoush', 'kibbeh', 'felafel', 'wot', 'mutton', 'eden', 'chicken_pita', 'waldorf', 'rendang', 'tandoori_chicken', 'gomen', 'bibb', 'panzanella', 'spanikopita', 'chicken_kabob', 'brochette', 'kisra', 'chicken_gyro', 'tandoor', 'gyro_plate', 'tandori', 'athena', 'gyro_sandwich', 'tika', 'briyani', 'falafal', 'rogan', 'chicken_kabobs', 'mutter', 'chicken_tikka', 'tibs', 'doro', 'shahi', 'dolma', 'lamb_curry', 'lebni', 'rice_pilaf', 'barg', 'hummas', 'spelt', 'mista', 'kebobs',

'spinach_pie', 'boti', 'kofte', 'keema', 'lula', 'jasmine_rice', 'kababs', 'goat_curry', 'falaffel', 'gyro_salad', 'gryo', 'kubideh', 'gyro_platter', 'sag', 'pel', 'nihari', 'misir', 'beef_kabob', 'pastitsio', 'adana', 'meni', 'nann', 'taboule', 'alicha', 'matar', 'pita_sandwich', 'bharta', 'labneh', 'pulao', 'hummos', 'kufta', 'shawarmas', 'massala', 'sabzi', 'athenian', 'dahl', 'mediteranean', 'kibbi', 'basmati_rice', 'bulgur', 'cucumber_sauce', 'baingan', 'chicken_kebab', 'fatoush', 'sega', 'watt', 'dolmeh', 'kibbe', 'saffron_rice', 'dhal', 'tahini_sauce', 'borek', 'spanokopita', 'sambusa', 'lamb_gyro', 'rasam', 'murgh', 'navratan', 'cesear', 'laffa', 'makhni', 'tsatziki', 'hommos', 'bulgar', 'chicken_shwarma', 'panner', 'chicken_souvlaki', 'tabbouli', 'biryanis', 'chapati', 'fatoosh', 'tsaziki', 'chicken_schwarma', 'papadums', 'malai_kofta', 'rotis', 'beef_kabobs', 'gobhi', 'kibbee', 'soltani', 'mysore', 'beef_shawarma', 'dolmathes', 'bhatura', 'lamb_kabob', 'swarma', 'sultani', 'tikil', 'hommus', 'mattar', 'cottage_pie', 'masalas', 'tadka', 'tricolore', 'pongal', 'shirazi', 'uthappam', 'idlis', 'schawarma', 'babaghanoush', 'mirchi', 'tzatsiki', 'tabooli', 'pasticcio', 'somosas', 'saag_paneer', 'sambusas', 'tzaziki_sauce', 'meditteranean']

195 ['salsa', 'guacamole', 'tortilla', 'tortillas', 'guac', 'flour', 'salsas', 'pico', 'gallo', 'corn_tortillas', 'tortilla_chips']

199 ['okonomiyaki', 'bbq_pork', 'tofu_soup', 'soondubu', 'beef_ribs', 'oxtail_soup', 'pork_buns', 'spicy_chicken', 'meat_jun', 'spicy_pork', 'sundubu', 'loco_moco', 'oxtails', 'beef_jerky', 'cucumber_salad', 'spareribs', 'chicken_soup', 'chicken_biryani', 'tocino', 'roast_pork', 'pork_ribs', 'beef_stew', 'beef_salad', 'dinuguan', 'momo', 'shiro', 'donburi', 'kim_chee', 'potsticker', 'palabok', 'charsiu', 'ramen_noodles', 'spam_musubi', 'miso_ramen', 'longanisa', 'manapua', 'beef_pho', 'duck_curry', 'crispy_chicken', 'seafood_soup', 'chicken_skewers', 'noodle_soup', 'gogi', 'lemon_chicken', 'chicken_satay', 'chicken_pho', 'bbq_ribs', 'honey_chicken', 'gizzard', 'bbh', 'shrimp_salad', 'katsudon', 'kalua_pig', 'rice_soup', 'beef_bowl', 'drunken_noodles', 'pinakbet', 'kushiage', 'egg_noodles', 'schezuan', 'hakata', 'singapore_noodles', 'chowmein', 'pork_bun', 'spicy_beef', 'chop_suey', 'lau_lau', 'pho_soup', 'pork_dish', 'pho_broth', 'stirfry', 'tsukemen', 'basil_chicken', 'oden', 'bistek', 'laulau', 'beef_dish', 'rice_cake', 'zaru', 'spicy_miso', 'bul', 'corn_chowder', 'spicy_ramen', 'nitamago', 'padthai', 'egg_tarts', 'crispy_pork', 'jjambbong', 'pho_tai', 'raman', 'pumpkin_curry', 'siopao', 'kelaguen', 'beef_soup', 'pepper_steak', 'mussaman', 'rice_plate', 'shrimp_appetizer', 'bbq_beef',

'spicy_noodles', 'soup_base', 'chow_mien', 'char_siu', 'tapsilog', 'jook', 'nengmyun', 'tsukune', 'poi', 'coconut_soup', 'nabeyaki', 'rice_cakes', 'kalua_pork', 'tantan', 'orange_beef', 'chasiu', 'noddle', 'spicy_shrimp', 'saimen', 'bagoong', 'shrimp_rolls', 'bulgoki', 'jamburrito', 'kuro', 'naengmyun', 'lemongrass_chicken', 'liempo', 'vesuvio', 'pof', 'siamese', 'maw', 'ground_pork', 'coconut_curry', 'beef_broth', 'bihon', 'chili_chicken', 'lettuce_cups', 'kailua', 'beef_noodle', 'bimbimbap', 'mango_salad', 'caldereta', 'spicy_basil', 'curry_soup', 'mongo', 'crispy_noodles', 'jerkey', 'dak', 'jade_chicken', 'oyster_sauce', 'spicy_eggplant', 'goyza', 'soba_noodles', 'kaarage', 'xiaolongbao', 'kome', 'beef_curry', 'puto', 'beef_noodles', 'bossam', 'guisado', 'kakuni', 'squash_curry', 'popcorn_chicken', 'har_gow', 'mapo_tofu', 'teriaki', 'pansit', 'pho_ga', 'pork_cutlet', 'brochettes', 'seasame', 'kerala', 'mushu', 'mumbai', 'daeji', 'spicy_soup', 'mock_chicken', 'house_chicken', 'ssam', 'tail_soup', 'bulkogi', 'laab', 'beef_skewers', 'beef_teriyaki', 'protein_bowl', 'tajine', 'hor', 'omurice', 'mango_curry', 'egg_plant', 'schezwan', 'jajangmyun', 'pineapple_curry', 'inasal', 'francaise', 'crispy_pata', 'gyudon', 'baboy', 'yakatori', 'lua', 'pot_pies', 'sari', 'phad_thai', 'mabo', 'massaman_curry', 'crispy_beef', 'jjajangmyun', 'rice_noodle', 'coconut_rice', 'dumplins', 'broccoli_beef', 'lamb_vindaloo', 'taho', 'ramyun', 'beef_broccoli', 'szechaun', 'lemongrass_soup', 'marination', 'beef_dishes', 'kurma', 'malabon', 'bulalo', 'yassa', 'andhra', 'rosu', 'penang_curry', 'cheong', 'lumpiang', 'yellow_curry', 'lok', 'oyako', 'teriyaki_beef', 'chadol', 'ginger_chicken', 'chicken_korma', 'fishballs', 'siomai', 'manok', 'curry_bowl', 'seafood_pancake', 'kare_kare', 'shashlik', 'vermicelli_bowl', 'shanxi', 'veggie_pho', 'curry_goat', 'crab_ragoon', 'pan_dulce', 'shoyu_ramen', 'lanzhou', 'union_pig', 'cracklings', 'bulaklak', 'duck_soup', 'tempuras', 'steam_buns', 'lechon_kawali', 'bun_rieu', 'drunkin', 'palak_paneer', 'pad_tai', 'donkatsu', 'silog', 'hyderabadi', 'friend_rice', 'chicken_masala', 'chana_masala', 'agedashi_tofu', 'seafood_pho', 'chettinad', 'spicy_dishes', 'diniguan', 'beef_combo', 'pork_katsu', 'pajeon', 'chicken_vindaloo', 'crispy_tofu', 'ramen_bowl', 'hargow', 'chicken_adobo', 'gui', 'curry_flavor', 'chicken_broth', 'beef_bulgogi', 'laing', 'tikki_masala', 'handi', 'chuka', 'hae', 'beef_tendon', 'mame', 'panag', 'manapuas', 'bastilla']

202 ['legs', 'crawfish', 'crab_legs', 'tail', 'crabs', 'puffs', 'tails', 'rangoon', 'claws', 'rangoons', 'lobsters', 'dungeness', 'ragoons', 'softshell', 'ragoon', 'cake_sandwich']

205 ['fries', 'chips', 'onion', 'dip', 'rings', 'pickles', 'pretzel', 'tots', 'fondue', 'knots', 'pickle', 'pretzels', 'onion_rings', 'breadsticks', 'croutons', 'wedges', 'straws', 'strings',

'truffle_fries', 'tater_tots', 'spears', 'potato_chips', 'fires', 'bread_sticks', 'tot', 'cut_fries', 'cajun_fries', 'waffle_fries', 'steak_fries', 'spear']

210 ['sprouts', 'verde', 'relleno', 'string', 'poblano', 'tomatillo', 'rellenos', 'pinto', 'shishito', 'ancho', 'garbanzo', 'xtreme', 'rellano', 'piquillo', 'fava', 'mung', 'lima', 'escarole', 'spouts', 'cannellini']

211 ['sashimi', 'nigiri', 'uni', 'wasabi', 'yellowtail', 'eel', 'caviar', 'hamachi', 'toro', 'ponzu', 'albacore', 'unagi', 'maki', 'mackerel', 'roe', 'nori', 'kama', 'ikura', 'ebi', 'masago', 'bonito', 'maguro', 'tobiko', 'ika', 'shiso']

212 ['coffee', 'latte', 'espresso', 'chai', 'mocha', 'cappuccino', 'americano', 'decaf', 'macchiato', 'expresso', 'frappe', 'venti', 'frap', 'baileys']

216 ['appetite', 'buds', 'palate', 'taste_buds', 'palette', 'tastebuds', 'mouths', 'stomachs', 'senses', 'minds', 'pallet', 'appetites', 'bellies', 'palates', 'desires', 'tummies', 'palettes', 'pallets', 'horizons']

218 ['attitude', 'smile', 'personality', 'knowledge', 'smiles', 'hospitality', 'skills', 'treatment', 'behavior', 'humor', 'friendliness', 'attitudes', 'interaction', 'encounter', 'professionalism', 'demeanor', 'rudeness', 'manners', 'attentiveness', 'urgency', 'service_skills', 'personalities', 'actions', 'interactions']

223 ['cronut', 'cronuts', 'honey_toast', 'butter_cake', 'godiva', 'peach_cobbler', 'topper', 'pizookie', 'frrrozen', 'popsicles', 'cinnamon_rolls', 'vosges', 'spumoni', 'paletas', 'beignet', 'donut_holes', 'coconuts', 'canolis', 'chocolate_shake', 'cupcakery', 'anglaise', 'blondie', 'greenbush', 'tuxedo', 'chiffon', 'cake_pops', 'milk_shake', 'cinnabon', 'confetti', 'magnolia', 'nutter', 'mushroom_soup', 'halo_halo', 'slade', 'banana_split', 'crop', 'sorbetto', 'souffles', 'ices', 'puddings', 'kolaches', 'raspados', 'fortune_cookie', 'streusel', 'ghirardelli', 'creme_brule', 'samoa', 'freeds', 'pizza_cookie', 'coconut_bark', 'munchkins', 'zeppoles', 'madagascar', 'chocolate_souffle', 'eggnog', 'pink_box', 'marzipan', 'clair', 'blondies', 'pazookie', 'tiramasu', 'zeppoli', 'kronuts', 'ingrediants', 'cheescake', 'granita', 'nougat', 'chocolate_dessert', 'razzy', 'suzuya', 'pound_cake', 'stracciatella', 'cream_puffs', 'apple_fritters', 'mirabelle', 'fruit_tart', 'parting', 'zeppole', 'manon', 'shamrock', 'cruller', 'canolli', 'acai_bowls', 'suarez', 'cookie_butter', 'frostings', 'turnovers', 'monkey_bread', 'semifreddo', 'goodie', 'pekara', 'snowstorm', 'glazes', 'lesley', 'cinnamon_sugar', 'crossiant', 'cait', 'macrons', 'froth', 'confection', 'madeleines', 'birthday_dessert', 'waffle_cones',

'sunflour', 'devonshire', 'tsoynami', 'chocolate_croissants', 'bundt_cakes', 'homemade_tortillas', 'couture', 'vanilla_bean', 'crullers', 'tort', 'zeppolis', 'brookie', 'gellato', 'speculoos', 'sugar_cookie', 'vanilla_custard', 'puff_pastry', 'funfetti', 'chocolate_bar', 'raspberry_sauce', 'biscoff', 'bananas_foster', 'mixtures', 'natas', 'gelatto', 'liege', 'sugar_cookies', 'brownie_sundae', 'snowball', 'chocolate_fondue', 'bomba', 'barbs', 'marshmellow', 'cassava', 'chocolate_brownie', 'cream_puff', 'mudslide', 'loafs', 'rosewater', 'cup_cakes', 'chocolate_souffl', 'brulees', 'pizzookie', 'pasteries', 'mcflurry', 'crape', 'tko', 'ringo', 'carib', 'epi', 'bonbons', 'bundtinis', 'bruleed', 'cinammon', 'napolean', 'coconut_cake', 'pinkbox', 'veil', 'snickerdoodles', 'cobblers', 'ollie', 'apple_pancake', 'babka', 'pavlova', 'quarts', 'rocher', 'chicory', 'hotcake', 'ferrero', 'bouchons', 'sparkler', 'tirimisu', 'clafoutis', 'kronut', 'sfogliatelle', 'nibs', 'chocolate_cupcake', 'ladyfingers', 'pazooki', 'cake_pop', 'yogurt_flavors', 'meringues', 'butter_cream', 'cake_donut', 'duncan', 'candy_bar', 'wildberry', 'caramel_brownie', 'almond_croissants', 'dozen_donuts', 'honey_bread', 'financier', 'mousses', 'butter_cup', 'blackout', 'madeline', 'lingonberry', 'chocolate_gelato', 'nestle', 'bundt_cake', 'napoleons', 'cannolli', 'chocolate_lava', 'cajeta', 'apple_cobbler', 'alfajores', 'pralines', 'choclate', 'chocolate_ganache', 'soynami', 'loganberry', 'chocolate_martini', 'fortunes', 'vanilla_shake', 'coconut_cream', 'frangelico', 'allegro', 'chocolate_raspberry', 'jinju', 'chocolate_tart', 'butter_cookies', 'chocolate_bread', 'thingie', 'mini_donuts', 'santiago', 'suzette', 'sweeties', 'chocolate_malt', 'hines', 'fruit_smoothies', 'marshmellows', 'babcock', 'baklavas', 'cake_donuts', 'tuile', 'valencia', 'mattress', 'butter_pie', 'brioches', 'fruit_toppings', 'cadbury', 'hazel', 'mini_cupcakes', 'rugelach', 'cinnamon_bun', 'bombolini', 'profiterole', 'cracker_crust', 'winchells', 'chocolate_cookie', 'cheese_soup', 'chocolate_milkshake', 'snowballs', 'pumpkin_pie', 'munchkin', 'gateau', 'vanilla_gelato', 'apple_tart', 'valrhona', 'irons', 'chocolate_pizza', 'strawberry_sauce', 'poundcake', 'pannacotta', 'almond_cookies', 'burlee', 'bacio', 'hummingbird', 'nilla', 'mud_pie', 'panacotta', 'chocolate_torte', 'cookie_sandwich', 'graham_crackers', 'showboy', 'moistest', 'cerreta', 'butterfat', 'gianduja', 'mousse_cake', 'apple_crisp', 'madelines', 'tortes', 'cookie_dessert', 'palmier', 'pistachio_gelato', 'cardamon', 'cinnabun', 'crossaint', 'layer_cake', 'chocolate_bars', 'decedent', 'butter_cookie', 'frangipane', 'sugarlips', 'sparklers', 'custom_cakes', 'fruit_tarts', 'brownie_dessert', 'salt_caramel', 'strawberry_shortcake', 'doughnut_holes', 'puddin', 'fruit_juices', 'sesame_balls', 'ghiradelli', 'cassata', 'peridot', 'palmiers', 'breton', 'flans', 'bacon_donut', 'tartelette', 'chocolate_cheesecake', 'aracelli', 'banana_foster', 'tammie_coe', 'canele', 'bombe', 'boysenberry', 'nib', 'cake_shake', 'rainbow_cookies', 'peanutbutter', 'fruit_drinks', 'confiture', 'tira', 'chocolate_cupcakes', 'bundtlet', 'chocolate_cookies', 'dessert_pizza', 'banana_crepe', 'zano', 'mango_sorbet', 'aguas_frescas', 'banana_smoothie', 'pumkin', 'bunt', 'misu', 'gummy_bears', 'lava_cake', 'banana_cream', 'petit_fours', 'nubs', 'chocolate_molten', 'chocolate_shavings', 'cake_bites', 'canollis', 'rollover', 'bundtlets', 'chomeur', 'apple_strudel', 'bundlets', 'raspberry_sorbet', 'macadamia_nut', 'daz', 'bundts', 'kaimaki', 'chocolate_truffles']

225 ['atmosphere', 'decor', 'ambiance', 'view', 'vibe', 'environment', 'interior', 'ambience', 'views']

232 ['spice', 'level', 'heat', 'kick', 'scale', 'spiciness', 'levels', 'spicy_food', 'spice_level', 'hotness', 'spicyness', 'spicy_level', 'spice_levels', 'heat_level']

235 ['cake', 'dessert', 'pie', 'cheesecake', 'cakes', 'desert', 'pudding', 'tiramisu', 'halo', 'sorbet', 'flan', 'cannoli', 'baklava', 'beignets', 'souffle', 'cobbler', 'tres', 'churros', 'lava', 'creme_brulee', 'shortcake', 'souffl', 'panna', 'cotta', 'leche', 'fritter', 'leches', 'molten', 'cannolis', 'dulce', 'torte', 'carrot_cake', 'flourless', 'ube', 'creme_br', 'sopapillas', 'amaretto', 'tres_leches', 'limoncello', 'strudel', 'profiteroles', 'malasadas', 'canoli', 'lime_pie', 'cleanser', 'chocolate_mousse', 'panna_cotta', 'budino', 'turon', 'haupia', 'cake_dessert']

236 ['rice', 'roll', 'rolls', 'noodles', 'broth', 'tofu', 'noodle', 'dumplings', 'tempura', 'miso', 'edamame', 'wonton', 'gyoza', 'wontons', 'spring_rolls', 'egg_rolls', 'potstickers', 'eggrolls', 'egg_roll', 'eggroll']

237 ['burgers', 'sandwiches', 'salads', 'pizzas', 'soups', 'pies', 'subs', 'pastas', 'hamburgers', 'calzones', 'hotdogs', 'cheesesteaks', 'cheeseburgers', 'hoagies']

239 ['drinks', 'drink', 'water', 'glasses', 'refills', 'champagne', 'mimosas', 'beverage', 'beverages', 'waters', 'mimosa', 'water_glasses']

242 ['lamb', 'duck', 'gras', 'sea', 'veal', 'dates', 'marrow', 'confit', 'escargot', 'quail', 'liver', 'rabbit', 'pate', 'foie_gras', 'crostini', 'venison', 'sweetbreads', 'bone_marrow', 'toasts', 'branzino', 'terrine', 'iberico', 'langoustine', 'escargots', 'popovers', 'diver', 'ratatouille', 'kurobuta', 'jamon', 'monkfish', 'shortrib', 'dover', 'cassoulet', 'shortribs', 'rillette', 'sweetbread', 'squab', 'rillettes', 'gravlax', 'berkshire', 'turbot', 'magret', 'roulade', 'pheasant']

243 ['pasta', 'meatballs', 'spaghetti', 'ravioli', 'risotto', 'gnocchi', 'lasagna', 'marinara', 'penne', 'bolognese', 'linguine', 'scampi', 'fettuccine', 'linguini', 'carbonara', 'rigatoni', 'tortellini', 'ziti', 'bucco', 'ragu', 'raviolis', 'pappardelle', 'cannelloni', 'mare', 'gnocci', 'fra', 'buco', 'fettuccini', 'tagliatelle', 'agnolotti', 'fettucine', 'fettucini', 'cavatelli', 'diavolo']

245 ['ones', 'curries', 'banchan', 'side_dishes', 'sushi_rolls', 'takoyaki', 'shumai', 'lettuce_wraps', 'broths', 'pot_stickers', 'crab_puffs', 'baos', 'xlb', 'calimari', 'roles', 'specialty_rolls', 'stews', 'edemame', 'soup_dumplings', 'momos', 'ramens', 'wanton', 'crab_rangoons', 'meat_dishes', 'gyozas', 'rices', 'bento_boxes', 'rice_dishes', 'summer_rolls', 'bentos', 'chicken_dishes', 'wantons', 'rice_bowls', 'seafood_dishes', 'pork_dumplings', 'curry_dishes', 'springrolls', 'phos', 'thai_dishes', 'rice_plates', 'currys']

246 ['sauce', 'butter', 'salt', 'oil', 'ranch', 'mayo', 'seasoning', 'mustard', 'ketchup', 'vinegar', 'aioli', 'horseradish', 'sriracha', 'rub', 'relish', 'dill', 'mayonnaise', 'tabasco', 'hoisin', 'soy_sauce', 'dijon', 'siracha', 'cholula', 'tapatio']

255 ['server', 'waitress', 'waiter', 'manager', 'bartender', 'hostess', 'host', 'sommelier']

264 ['cuban', 'tamale', 'mole', 'empanadas', 'pollo', 'loco', 'arepa', 'empanada', 'fundido', 'diablo', 'elote', 'cubano', 'rican', 'rico', 'pupusa', 'camarones', 'ranchero', 'papas', 'tostones', 'saltado', 'croquetas', 'pernil', 'bravas', 'mariscos', 'chicharron', 'patatas', 'ropa', 'mofongo', 'molcajete', 'pescado', 'caldo', 'camaron', 'vieja', 'diabla', 'maduros', 'frita', 'sopa', 'puerco', 'plancha', 'fritas', 'gallina']

265 ['pasties', 'basics', 'paninis', 'flatbreads', 'poutines', 'pasta_dishes', 'sandwhiches', 'odds', 'examples', 'vegan_options', 'haves', 'quiches', 'possibilities', 'breakfast_items', 'sammies', 'bases', 'taster', 'entries', 'sandwichs', 'stuffs', 'essentials', 'samplers', 'sammiches', 'tasters', 'bets', 'dipping_sauces', 'bruschettas', 'menu_choices', 'specialty_drinks', 'upgrades', 'amenities', 'comparisons', 'breakfast_options', 'ensemble', 'entres', 'usuals', 'necessities', 'mignardises', 'similarities', 'pickings', 'nibbles', 'characteristics', 'bakery_items', 'sake_bombs', 'bbq_sauces', 'yummies', 'dessert_options', 'lunch_options', 'pastys', 'originals', 'priorities', 'veggie_options', 'food_selections', 'specialities', 'tidbits', 'mileage', 'samplings', 'appitizers', 'grades', 'drink_options', 'vegetarian_options', 'appetisers', 'assortments', 'dinner_specials', 'breakfast_foods', 'lunch_items', 'specialty_cocktails', 'tiers', 'strengths', 'appies', 'savories', 'attributes', 'renditions', 'quartet', 'menu_selections', 'rums', 'specialty_pizzas', 'meat_options', 'verity',

'schnitzels', 'foodstuffs', 'alcohols', 'specialty_items', 'fares', 'dinner_options', 'food_offerings', 'mainstays', 'pates', 'interpretations', 'contenders', 'indulgences', 'dinner_items', 'dinner_entrees', 'aps', 'charcuteries', 'ketchups', 'seafood_options', 'accouterments', 'weaknesses', 'hams', 'accoutrement', 'temptations', 'liqueurs', 'infusions', 'flavor_profiles', 'novelties', 'signature_dishes', 'salad_options', 'gravies', 'breakfast_dishes', 'edibles', 'iterations', 'food_stations', 'oddities', 'comfort_foods', 'flavor_options', 'breakfast_specials', 'menu_offerings', 'feasts', 'healthier_options', 'delectables', 'variants', 'notables', 'batters', 'incarnations', 'sauce_options', 'breakfast_choices', 'grocery_items', 'stuffings', 'cubans', 'drink_choices', 'purees', 'brunch_options', 'sandwich_options', 'meat_choices', 'vintages', 'burger_options', 'signature_drinks', 'rieslings', 'signature_cocktails', 'chocolatiers', 'style_dishes', 'topings', 'draughts', 'buffet_items', 'style_restaurants', 'veg_options', 'doughs', 'side_options', 'meat_selections', 'tap_beers', 'flavor_combos', 'salamis', 'champagnes', 'hitters', 'icings', 'gelato_flavors', 'varities']

267 ['poke', 'kimchi', 'bulgogi', 'udon', 'kalbi', 'spam', 'tonkatsu', 'tonkotsu', 'adobo', 'vermicelli', 'lechon', 'shoyu', 'soba', 'bibimbap', 'kimchee', 'ono', 'chi', 'kare', 'chashu', 'lumpia', 'musubi', 'lau', 'jun', 'miso_soup', 'sisig', 'chee', 'pancit', 'sukiyaki', 'yakisoba', 'saimin', 'furikake', 'hotpot', 'galbi', 'chirashi', 'spring_roll', 'macaroni_salad', 'chasu', 'sinigang', 'ban', 'bim', 'teriyaki_chicken', 'mac_salad', 'bento_box', 'chan', 'chicken_katsu', 'shio', 'rice_bowl', 'bap', 'pata', 'rice_noodles', 'chicken_teriyaki', 'chicken_bowl', 'bangus', 'bulgolgi', 'bop', 'dolsot', 'jjigae', 'japchae', 'noddles', 'mandu', 'myun', 'masubi', 'jap', 'jang', 'bibimbop', 'musubis', 'kal', 'chap', 'bibim', 'dubu', 'mandoo', 'dol', 'jigae', 'sot', 'gae', 'chigae', 'bim_bap']

278 ['parm', 'marsala', 'osso', 'parmigiana', 'pot_pie', 'piccata', 'salad_sandwich', 'dish_pizza', 'saltimbocca', 'picatta', 'jidori', 'madeira', 'cacciatore', 'broaster', 'francese', 'pesto_pizza', 'scallopini', 'parmagiana', 'generals', 'katsu_curry', 'crispers', 'kiev', 'limone', 'parm_sandwich', 'satays', 'paillard', 'pesto_sandwich', 'picata', 'scaloppine', 'scaloppini', 'shawarma_plate', 'cordon_bleu', 'piccatta', 'scarpariello']

283 ['masala', 'tikka', 'paneer', 'tandoori', 'biryani', 'samosas', 'dosa', 'roti', 'vindaloo', 'samosa', 'korma', 'lassi', 'saag', 'dal', 'tikka_masala', 'kofta', 'chaat', 'gobi', 'palak', 'butter_chicken', 'biriyani', 'pakora', 'chutneys', 'makhani', 'malai', 'dosas', 'raita', 'sambar', 'kheer', 'daal', 'chana', 'mango_lassi', 'gulab', 'jamun', 'manchurian', 'pakoras', 'tikki', 'puri',

'paratha', 'vada', 'pav', 'naans', 'poori', 'channa', 'papadum', 'halwa', 'chole', 'idli', 'dosai', 'pani', 'sambhar', 'bhindi', 'dahi', 'bhel']

289 ['burger', 'salad', 'sandwich', 'turkey', 'wrap', 'pastrami', 'cheeseburger', 'hamburger', 'panini', 'cheesesteak', 'meatloaf', 'blt', 'slider', 'reuben', 'hotdog', 'ruben', 'roast_beef', 'turkey_sandwich', 'rueben']

290 ['masa', 'yeast', 'maple_syrup', 'hominy', 'sizzle', 'substitutes', 'egg_whites', 'brine', 'veges', 'margarine', 'roux', 'canola', 'cornstarch', 'starches', 'gram', 'yoke', 'vegies', 'flour_tortillas', 'cripsy', 'tyson', 'drippings', 'pinto_beans', 'spicy_kick', 'flavourless', 'waxy', 'flava', 'cheesiness', 'vegtables', 'corn_tortilla', 'sushi_rice', 'leathery', 'slimey', 'panchan', 'banchans', 'ghee', 'bead', 'dorito', 'heft', 'cinnamony', 'flavorfull', 'husk', 'carby', 'fishiness', 'stickiness', 'legumes', 'chewey', 'saltine', 'saucey', 'consistancy', 'digiorno', 'rice_paper', 'taco_shells', 'imitation_crab', 'icey', 'pintos', 'outsides', 'smoke_flavor', 'vegs', 'underside', 'lather', 'taco_shell', 'spec', 'tender_meat', 'gluey', 'crisco', 'ground_meat', 'orangey', 'bouncy', 'zingy', 'bacony', 'blan', 'coffee_flavor', 'oiliness', 'greesy', 'kernel', 'tasy', 'side_item', 'hamburger_bun', 'twang', 'meatiness', 'peanutty', 'gaminess', 'pizza_box', 'cheesy_goodness', 'manjoo', 'pillsbury', 'chocolate_flavor', 'thinness', 'bleck', 'smoky_flavor', 'toughness', 'wow_factor', 'flours', 'lunchmeat', 'potatoey', 'crumby', 'frig', 'fuzz', 'freezer_section', 'bread_crumbs', 'lunch_meat', 'egg_yolk', 'reheats', 'coconutty', 'wonderbread', 'gob', 'mystery_meat', 'fingernail', 'droopy', 'yolky', 'baloney', 'vegis', 'headcheese', 'flour_tortilla', 'cornflakes', 'saltines', 'dinner_roll', 'bony', 'pizzaz', 'cobs', 'cold_cuts', 'overdid', 'heartiness', 'juicy_meat', 'vegatables', 'overtones', 'coconut_flavor', 'imparts', 'zilch', 'burger_meat', 'gooeyness', 'juicey', 'circumference', 'greasey', 'juicy_chicken', 'stouffer', 'tombstone', 'stouffers', 'chef_boyardee', 'smudge', 'bisquick', 'spicy_flavor', 'counterbalance', 'smokey_flavor', 'tender_chicken', 'creamcheese', 'granules', 'pie_crust', 'kronos', 'viscosity', 'crispyness', 'totino', 'tomato_paste', 'mushiness', 'redolent', 'crunches', 'garnishments', 'garnishment', 'cat_food', 'tender_beef', 'cornflake', 'prefab', 'rectangles', 'stone_cold', 'briney', 'wads', 'baby_food', 'twig', 'jasons']

292 ['portions', 'size', 'amount', 'portion', 'sizes', 'servings', 'amounts', 'portion_size', 'portion_sizes', 'proportions', 'quantities', 'food_portions', 'helpings', 'serving_sizes', 'potions']

298 ['breads', 'oatmeal', 'granola', 'parfait', 'dips', 'poppy', 'raisin', 'cereal', 'baguettes', 'spreads', 'sunflower', 'jams', 'challah', 'butters', 'loaves', 'chia', 'jellies', 'bran', 'pumpernickel', 'multigrain', 'poppyseed', 'hemp', 'parfaits', 'danishes', 'mustards', 'blintzes', 'flax', 'oat', 'cereals']

301 ['sushi', 'buffet', 'pho', 'ramen', 'ayce', 'shabu', 'dim_sum', 'thai_food', 'kbbq', 'dimsum']

304 ['ceviche', 'gumbo', 'paella', 'jambalaya', 'pan_roast', 'maine', 'alligator', 'gazpacho', 'crab_cakes', 'lobster_roll', 'cioppino', 'clam_chowder', 'trifecta', 'shooters', 'walleye', 'lobster_bisque', 'crabcake', 'bouillabaisse', 'haddock', 'crab_cake', 'etouffee', 'crabcakes', 'rockefeller', 'fish_sandwich', 'shooter', 'shrimp_cocktail', 'gulf', 'mahi_mahi', 'craw', 'gator', 'poboy', 'etoufee', 'conch', 'rattlesnake', 'kumamoto', 'shrimp_taco', 'seafood_platter', 'lobsicle', 'chicken_pasta', 'fish_taco', 'barramundi', 'whiting', 'half_shell', 'redfish', 'talapia', 'etouffe', 'shuck']

307 ['hummus', 'gyro', 'pita', 'naan', 'falafel', 'gyros', 'shawarma', 'kabob', 'tzatziki', 'baba', 'kabobs', 'pitas', 'kebab', 'nan', 'schwarma', 'tahini', 'souvlaki', 'shwarma', 'shish', 'humus', 'falafels', 'tabouli', 'tabbouleh', 'kebabs', 'dolmas', 'ganoush', 'spanakopita', 'kafta', 'dolmades', 'kabab', 'lentil_soup', 'haji', 'koobideh', 'moussaka', 'shawerma', 'babaganoush', 'ghanoush', 'kebob', 'kefta', 'tabouleh', 'doner', 'grape_leaves', 'ghanouj', 'taouk', 'ghannouj', 'ganoosh', 'ganouj']

APPENDIX B. SELECTED CLUSTERS BASED ON LDA

0 ['food', ' place', ' time', ' service', ' pizza', ' chicken', ' order', ' restaurant', ' menu', ' love', ' burger', ' cheese', ' salad', ' bar', ' sushi', ' table', ' fries', ' lunch', ' wait', ' minutes', ' staff', ' vegas', ' server', ' night', ' dinner', ' steak', ' way', ' rice', ' experience', ' everything', ' beer', ' bit', ' drinks', ' dish', ' location', ' side', ' flavor', ' buffet', ' fish', ' stars', ' taste', ' beef', ' drink', ' times', ' something', ' shrimp', ' price', ' pork', ' hour', ' waitress', ' tacos', ' tea', ' husband', ' quality', ' waiter', ' roll', ' atmosphere', ' prices', ' friends', ' wings', ' top', ' check', ' half', ' wife', ' things', ' selection', ' items', ' tables', ' rolls', ' spot', ' burgers', ' town', ' dining', ' everyone', ' years', ' flavors', ' garlic', ' pasta', ' house', ' fun', ' look', ' party', ' room', ' salmon', ' owner', ' brunch', ' beans', ' options', ' chef', ' group', ' portions', ' kids', ' sandwiches', ' week', ' style', ' patio', ' point', ' dog', ' money', ' mac', ' name', ' potato', ' business', ' music', ' average', ' guy', ' vegan', ' drive', ' boyfriend', ' today', ' tuna', ' wish', ' salty', ' birthday', ' lots', ' amount', ' medium', ' portion', ' beers', ' toast', ' hours', ' variety', ' orders', ' servers', ' tip', ' offer', ' year', ' bartender', ' corn', ' yum', ' ingredients', ' chili', ' plates', ' tender', ' cut', ' onion', ' duck', ' pick', ' ambiance', ' vegetarian', ' boba', ' delivery', ' hit', ' salads', ' days', ' salt', ' tasting', ' man', ' gluten', ' girl', ' looks', ' guys', ' hotel', ' specials', ' friday', ' lettuce', ' view', ' lady', ' bun', ' fruit', ' date', ' customer_service', ' waffles', ' gelato', ' space', ' option', ' nachos', ' weekend', ' world', ' pepper', ' spinach', ' dip', ' juicy', ' hands', ' hope', ' son', ' belly', ' que', ' veggie', ' stay', ' grab', ' texture', ' city', ' foie', ' fat', ' une', ' thanks', ' guacamole', ' charlotte', ' juice', ' carne', ' dogs', ' value', ' buffets', ' valley', ' coconut', ' grill', ' dim', ' establishment', ' pittsburgh', ' kinda', ' vietnamese', ' mention', ' club', ' dumplings', ' pas', ' girlfriend', ' attention', ' months', ' wrap', ' crowd', ' heat', ' shake', ' favorites', ' management', ' york', ' spring', ' desert', ' ham', ' yeah', ' cash', ' boy', ' expectations', ' cuisine', ' month', ' omelet', ' tap', ' platter', ' par', ' locations', ' kid', ' stick', ' smile', ' fry', ' mayo', ' smell', ' dollars', ' beat', ' greens', ' margaritas', ' miso', ' california', ' biscuits', ' madison', ' pineapple', ' notch', ' word', ' plan', ' rings', ' weeks', ' tortilla', ' smoothie', ' bartenders', ' omg', ' hair', ' play', ' caesar', ' environment', ' pepperoni', ' kick', ' selections', ' pickles', ' meh', ' interior', ' center', ' round', ' building', ' bellagio', ' poke', ' workers', ' candy', ' truck', ' nights', '

heaven', ' enchiladas', ' monday', ' future', ' oven', ' refills', ' smoke', ' crawfish', ' tuesday', ' sangria', ' airport', ' mess', ' tait', ' patty', ' omelette', ' nigiri', ' naan', ' lines', ' break', ' details', ' steakhouse', ' offers', ' giant', ' mais', ' request', ' complaints', ' mary', ' spots', ' quesadilla', ' stomach', ' anniversary', ' pastrami', ' tempe', ' walls', ' doubt', ' dollar', ' hype', ' recommendation', ' difference', ' wood', ' art', ' discount', ' plastic', ' thursday', ' ladies', ' leftovers', ' board', ' meatball', ' caf', ' fare', ' rate', ' crunch', ' kinds', ' las_vegas', ' fingers', ' tots', ' signature', ' weather', ' strips', ' craft', ' texas', ' skin', ' cheeseburger', ' tom', ' smoothies', ' ceviche', ' movie', ' leaves', ' situation', ' occasion', ' wing', ' grease', ' teas', ' jam', ' boring', ' protein', ' guest', ' points', ' cornbread', ' mon', ' suggestions', ' support', ' napkins', ' offering', ' ass', ' carnitas', ' kale', ' artichoke', ' joke', ' upstairs', ' sizes', ' chowder', ' pig', ' spoon', ' macarons', ' curds', ' container', ' christmas', ' ive', ' sooo', ' occasions', ' dive', ' dans', ' regulars', ' whiskey', ' juices', ' waitstaff', ' clam', ' doors', ' hurry', ' peach', ' meeting', ' patient', ' mill', ' edamame', ' effort', ' calories', ' health', ' cards', ' example', ' dates', ' temperature', ' highlight', ' valet', ' bien', ' changes', ' rocks', ' reasons', ' fruits', ' standards', ' liquor', ' thumbs', ' oyster', ' beverage', ' theme', ' sont', ' opportunity', ' bistro', ' match', ' subway', ' appetite', ' joints', ' pecan', ' plaza', ' pack', ' masala', ' pleasure', ' fav', ' soul', ' detail', ' eater', ' child', ' search', ' potential', ' vacation', ' flan', ' pretzels', ' cauliflower', ' foodie', ' btw', ' heck', ' exception', ' boys', ' everytime', ' hopes', ' caprese', ' pico', ' quiche', ' trouble', ' count', ' display', ' distance', ' weekday', ' cleaning', ' chinatown', ' torta', ' layer', ' banh', ' hipster', ' worry', ' layout', ' matt', ' seaweed', ' bubble', ' killer', ' muffins', ' photo', ' gilbert', ' trash', ' personality', ' beware', ' earth', ' sam', ' penny', ' smoky', ' manner', ' comparison', ' winter', ' managers', ' everybody', ' horchata', ' reuben', ' news', ' patties', ' yay', ' handful', ' boss', ' crew', ' members', ' veg', ' okra', ' cappuccino', ' mimosa', ' wallet', ' energy', ' summerlin', ' brick', ' char', ' sausages', ' soo', ' lovers', ' polish', ' factor', ' period', ' sodas', ' bones', ' rubbery', ' quinoa', ' smokey', ' peak', ' screen', ' blow', ' kill', ' citrus', ' grub', ' runs', ' oreo', ' deliciousness', ' spam', ' pricy', ' jar', ' respect', ' mama', ' mcdonald', ' panda', ' shack', ' brussels', ' gallo', ' district', ' tater', ' knowledge', ' test', ' promise', ' fyi', ' cracker', ' straw', ' improvement', ' blast', ' pics', ' array', ' sight', ' fridge', ' freezer', ' degrees', ' max', ' power', ' sriracha', ' cross', ' dutch', ' creek', ' milk_tea', ' quesadillas', ' bison', ' ideal', ' boot', ' fruity', ' furniture', ' brownies', ' edges', ' food_quality', ' lackluster', ' skimp', ' success', ' charges', ' granola', ' lattes', ' train', ' concern', ' jasmine', ' habit', ' hangout', ' bowling', ' breath', ' degree', ' kinks', ' kbbq', ' feedback', ' suspect', ' hunt', ' disaster', ' brothers', ' rotation', ' pale', ' cubes', ' yolk', ' breakfast_burrito', ' amber', ' lavender', ' mule', ' matcha', ' flow', ' signage', ' side_dishes', ' neighbors', ' mason', ' moms', ' portillo', ' blaze', ' trifecta', ' nitro', ' potter', ' restaurant_namestore', ' customer', ' customers', ' shop', ' employees', ' help', ' cashier', ' employee', ' register', ' popcorn', ' self', ' product', ' mileslot', ' front', ' parking', ' door', ' walk', ' car', ' corner', ' park', ' moment', ' gas', ' drunken', ' hits', ' fed', ' carsbreakfast', ' bacon', ' eggs', ' egg', ' potatoes', ' burrito', ' pancakes', ' crepe', ' crepes', ' diner', ' benedict', ' venuecool', ' show', ' game', ' watch', ' vibe', ' games', ' sports', ' pretzel', ' pumpkin', ' team', ' season', ' football', ' wont', ' volume', ' executiontaco', ' street', ' dad', ' bell', ' women', ' repeat', ' passion', ' mex', ' fremont', ' soho', ' genreview', ' yelp', ' reviews', ' reading', ' croissant', ' hey', ' story', ' decision', ' dude', ' description', ' edgepie', ' wall', ' ribeye', ' hole', ' mocha', ' michael', ' quarter', ' coffee_shop', ' girlfriends', ' leaf', ' shape', ' lesson', ' praisesauce', ' spicy', ' bbq', ' ribs', ' spice', ' ranch', ' buffalo', ' baby', ' photos', ' businesses', ' challengedishes', ' ramen', ' noodles', ' noodle', ' casino', ' west', ' chow', ' coast', ' squid', ' fake']

3 ['sandwich', ' bread', ' turkey', ' avocado', ' deli', ' roast', ' coleslaw', ' fatty', ' wheat', ' pickle', ' bet']

5 ['wine', ' glass', ' list', ' bottle', ' poutine', ' glasses', ' wines', ' bruschetta', ' bottles', ' flight']

14 ['oil', ' olive', ' ravioli', ' visits', ' vinegar', ' fondue', ' cod', ' buddy', ' watermelon', ' rosemary', ' yuck', ' goodies', ' containers', ' almonds', ' lentil', ' flakes']

15 ['chips', ' salsa', ' curry', ' wow', ' owners', ' tortillas', ' tons', ' enchilada', ' balls', ' guac', ' tamales']

29 ['cake', ' donuts', ' cupcakes', ' needs', ' sugar', ' donut', ' bakery', ' lime', ' burritos', ' key', ' doughnuts', ' jelly', ' icing']

36 ['dessert', ' desserts', ' cheesecake', ' margarita', ' legs', ' creme', ' tiramisu', ' sweetness', ' balance', ' squash', ' brulee', ' deserts', ' fig', ' reach']

41 ['sauces', ' pad', ' mein', ' chris', ' tenders', ' croissants', ' potato_fries', ' penne', ' creations', ' depth', ' friendliness']

55 ['cream', ' ice', ' treat', ' vanilla', ' cups', ' strawberries', ' blueberry', ' smooth', ' cone', ' setup']

65 ['pho', ' tofu', ' craving', ' bean', ' mango', ' basil', ' arizona', ' sprouts', ' mimosas', ' grass']

APPENDIX C. UNSELECTED CLUSTERS BASED ON WORD2VEC

78 ['places', 'restaurants', 'buffets', 'locations', 'spots', 'bars', 'shops', 'stores', 'joints', 'businesses', 'chains', 'establishments', 'steakhouses', 'houses', 'eateries', 'breweries', 'cities', 'bakeries', 'gems', 'cafes', 'pubs', 'coffee_shops', 'venues', 'sushi_places', 'pizza_places', 'thai_restaurants', 'delis', 'sushi_restaurants', 'breakfast_places', 'delicacies', 'pizzerias', 'steak_houses', 'burger_joints', 'burger_places', 'treasures', 'pizza_joints']

93 ['hours', 'noon', 'midnight', 'peak', 'tomorrow', 'til', 'lunchtime', 'hrs', 'holidays', 'clock', 'closes', 'restaurant_week', 'lunch_time', 'closing_time', 'dinner_time', 'lunch_hours', 'business_hours', 'peak_hours', 'lunch_hour', 'dinner_hours']

101 ['part', 'secret', 'breakfast_place', 'burger_joint', 'lunch_spot', 'pizza_place', 'breakfast_spot', 'worlds', 'sushi_place', 'pizza_joint', 'sandwich_shop', 'burger_place', 'secrets', 'sushi_spot', 'donut_shop', 'seller', 'breakfast_joint', 'sushi_joint', 'coffee_place', 'bbq_joint', 'bbq_place', 'brunch_spot', 'sub_shop', 'pizza_shop', 'pizza_spot', 'sellers', 'bar_none', 'kept_secret', 'coffee_spot', 'food_spot', 'bbq_spot', 'delivery_place']

123 ['friend', 'husband', 'wife', 'boyfriend', 'mom', 'daughter', 'son', 'girlfriend', 'sister', 'mother', 'dad', 'brother', 'fiance', 'partner', 'father', 'cousin', 'companion', 'companions', 'partners']

147 ['dress', 'wear', 'shirt', 'pants', 'clothes', 'shirts', 'hat', 'jeans', 'shorts', 'shoes', 'clothing', 'uniform', 'jacket', 'heels', 'outfits', 'tattoos', 'uniforms', 'nurse', 'gun', 'booty', 'dresses', 'gear', 'flops', 'boots', 'coats', 'sweater', 'toe', 'outfit', 'nurses', 'makeup', 'sunglasses', 'apron', 'pony', 'horn', 'pajamas', 'bra', 'sleeve', 'boobs', 'guns', 'glove', 'jackets', 'sleeves', 'sandals', 'costume', 'costumes', 'plaid', 'gown', 'underwear', 'sneakers', 'piercings', 'slacks', 'cleavage', 'gowns', 'panties']

164 ['minutes', 'min', 'mins', 'minute']

168 ['plate', 'plates', 'cups', 'basket', 'boxes', 'tray', 'bags', 'jar', 'package', 'containers', 'trays', 'mug', 'pots', 'baskets', 'jars', 'mason', 'mugs', 'packets', 'packaging', 'packages', 'pans', 'batches', 'shapes', 'tubs', 'tins']

179 ['chicago', 'california', 'san', 'nyc', 'state', 'hawaii', 'mexico', 'los', 'japan', 'states', 'wisconsin', 'cali', 'jersey', 'boston', 'tex', 'seattle', 'france', 'brooklyn', 'india',

'houston', 'thailand', 'detroit', 'tucson', 'nevada', 'cal', 'miami', 'canada', 'florida', 'korea', 'austin', 'philadelphia', 'germany', 'denver', 'europe', 'usa', 'toronto', 'michigan', 'portland', 'spain', 'dallas', 'london', 'puerto', 'ohio', 'atlanta', 'quebec', 'oregon', 'vietnam', 'taiwan', 'sedona', 'illinois', 'peru', 'ireland', 'nation', 'rome', 'utah', 'greece', 'vancouver', 'iowa', 'virginia', 'milwaukee', 'minnesota', 'philippines', 'reno', 'cleveland', 'indiana', 'australia', 'flagstaff', 'munich', 'suburbs', 'orlando', 'carolinas', 'connecticut', 'honolulu', 'baltimore', 'argentina', 'nebraska', 'hawai', 'venice', 'pennsylvania', 'florence', 'asheville', 'dublin', 'oahu', 'raleigh', 'idaho', 'los_angeles', 'sacramento', 'albuquerque', 'minneapolis', 'columbus', 'ontario', 'burbs', 'missouri', 'columbia', 'oklahoma', 'tampa', 'arkansas', 'chicagoland', 'prescott', 'huntersville']

196 ['john', 'matt', 'dave', 'george', 'jim', 'bob', 'tommy', 'andy', 'johnny', 'ben', 'maria', 'juan', 'todd', 'mrs', 'eddie', 'captain', 'carlos', 'rachel', 'rick', 'peter', 'jerry', 'chelsea', 'charlie', 'christopher', 'ken', 'randy', 'ann', 'ron', 'bryan', 'casey', 'tiffany', 'paco', 'marie', 'gang', 'wayne', 'dean', 'jon', 'marco', 'martin', 'stacy', 'keith', 'phil', 'duke', 'jackson', 'cooper', 'pedro', 'julia', 'luis', 'ross', 'leo', 'bahama', 'piper', 'kat', 'william', 'ricardo', 'shane', 'nancy', 'eva', 'mel', 'oliver', 'coach', 'miguel', 'marty', 'heidi', 'javier', 'ricky', 'johnson', 'bruce', 'andre', 'bond', 'robbie', 'zack', 'president', 'julio', 'neil', 'cathy', 'sergio', 'mandy', 'marilyn', 'beatles', 'ana', 'brady', 'lyrics', 'ralph', 'kristin', 'lupe', 'ernie', 'britney', 'howard', 'gump', 'mayor', 'bush', 'ronnie', 'williams', 'anne', 'ogden', 'moore', 'cameron', 'donnie', 'wright', 'stewart', 'edgar', 'marley', 'lloyd', 'obama', 'dion', 'oprah', 'martha', 'carmen', 'willie', 'richman', 'bradley', 'fernando', 'hughes', 'celine', 'kumar', 'fina', 'dona', 'ruths', 'horton', 'luger', 'danko', 'hortons']

203 ['chris', 'david', 'mike', 'michael', 'steve', 'jason', 'scott', 'james', 'anthony', 'paul', 'kevin', 'nick', 'brian', 'ryan', 'jeff', 'lisa', 'justin', 'daniel', 'josh', 'robert', 'jessica', 'michelle', 'tim', 'sarah', 'nicole', 'richard', 'jennifer', 'sean', 'andrew', 'taylor', 'heather', 'sweetheart', 'brandon', 'rob', 'christina', 'jen', 'gary', 'katie', 'aaron', 'larry', 'jay', 'joey', 'jenny', 'greg', 'tina', 'danny', 'brittany', 'doug', 'jamie', 'angela', 'kyle', 'megan', 'donna', 'laura', 'kathy', 'melissa', 'anna', 'julie', 'dennis', 'jesse', 'gina', 'andrea', 'brad', 'jake', 'erin', 'tyler', 'angie', 'lauren', 'jordan', 'jackie', 'chad', 'sara', 'charles', 'kenny', 'karen', 'kate', 'doll', 'danielle', 'shannon', 'jeremy', 'tammy', 'vanessa', 'liz', 'tracy', 'joseph', 'shawn', 'victor',

'steven', 'jessie', 'cindy', 'matthew', 'carol', 'nate', 'roger', 'nikki', 'christine', 'jonathan', 'natalie', 'courtney', 'denise', 'rebecca', 'terry', 'lindsey', 'manny', 'erica', 'ivan', 'veronica']

209 ['favorites', 'fav', 'fave', 'faves', 'favs', 'favourites']

224 ['game', 'football', 'network', 'hockey', 'channel', 'baseball', 'episode', 'soccer', 'ufc', 'nfl', 'basketball', 'television', 'tournament', 'ddd', 'suns', 'youtube', 'giants', 'bourdain', 'espn', 'pirates', 'nba', 'cardinals', 'badger', 'food_network', 'channels', 'football_game', 'panthers', 'playoffs', 'finals', 'playoff', 'softball', 'championship', 'dbacks', 'cubs', 'ncaa', 'football_games', 'coyotes', 'diamondbacks']

241 ['july', 'march', 'june', 'april', 'december', 'january', 'august', 'october', 'november', 'february', 'september', 'jan', 'dec', 'feb', 'sept', 'oct', 'nov']

251 ['yelpers', 'reviewers', 'reviewer', 'yelper']

270 ['problem', 'issue', 'issues', 'problems', 'delay', 'confusion', 'barrier', 'errors', 'flaw', 'miscommunication', 'mishap', 'delays', 'hiccup', 'glitch', 'snafu', 'discrepancy', 'mixup']

REFERENCES

Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, vldb* (Vol. 1215, pp. 487–499).

Andrzejewski, D., Zhu, X., & Craven, M. (2009). Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th annual international conference on machine learning* (pp. 25–32).

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Acl (1)* (pp. 238–247).

Beineke, P., Hastie, T., Manning, C., & Vaithyanathan, S. (2004). Exploring sentiment summarization. In *Proceedings of the aaai spring symposium on exploring attitude and affect in text: Theories and applications* (Vol. 39).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

Chen, Z., & Liu, B. (2014). Mining topics in documents: Standing on the shoulders of big data. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1116–1125).

Choi, Y., Cardie, C., Riloff, E., & Patwardhan, S. (2005). Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 355–362).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493-2537.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on world wide web* (pp. 519–528).

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955.

Grabmeier, J., & Rudolph, A. (2002). Techniques of cluster algorithms in data mining. *Data Mining and Knowledge Discovery*, *6*(4), 303–360.

Guha, S., Rastogi, R., & Shim, K. (1999). Rock: A robust clustering algorithm for categorical attributes. In *Data engineering, 1999. proceedings., 15th international conference on* (pp. 512–521).

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146–162.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100–108.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In Proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval (pp. 50–57).

Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, *5*(Nov), 1457–1469.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining (pp. 168–177).

Hu, Y., Boyd-Graber, J., Satinoff, B., & Smith, A. (2014). Interactive topic modeling. *Machine learning*, *95*(3), 423–469.

Huang, J., Rogers, S., & Joo, E. (2014). Improving restaurants by extracting subtopics from Yelp reviews. *iConference 2014 (Social Media Expo)*.

Jagarlamudi, J., Daum´e III, H., & Udupa, R. (2012). Incorporating lexical priors into topic models. In *Proceedings of the 13th conference of the european chapter of the association for computational linguistics* (pp. 204–213).

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

Jakob, N., & Gurevych, I. (2010). Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 1035–1045).

Kowalski, G. (1998). Information retrieval systems: Theory and implementation. *Computers and Mathematics with Applications*, *5*(35), 133.

Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth acm sigkdd international conference on knowledge discovery and data mining* (pp. 16–22).

Lerman, K., Blair-Goldensohn, S., & McDonald, R. (2009). Sentiment summarization: Evaluating and learning user preferences. In *Proceedings of the 12th conference of the european chapter of the association for computational linguistics* (pp. 514–522).

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, *5*(1), 1–167.

Liu, B., Blasch, E., Chen, Y., Shen, D., & Chen, G. (2013). Scalable sentiment classification for big data analysis using naive bayes classifier. In *Big data, 2013 ieee international conference on* (pp. 99–104).

Lu, Y., Zhai, C., & Sundaresan, N. (2009). Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on world wide web* (pp. 131–140).

Luca, M. (2011). Reviews, reputation, and revenue: The case of Yelp. com. *Com*
(September 16, 2011). Harvard Business School NOM Unit Working Paper (12-016).

Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(Nov), 2579-2605.

McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th acm conference on recommender systems* (pp. 165–172).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Hlt-naacl* (Vol. 13, pp. 746–751).

Moghaddam, S., & Ester, M. (2011). Ilda: Interdependent lda model for learning latent aspects and their ratings from online product reviews. In Proceedings of the 34th international acm sigir conference on research and development in information retrieval (pp. 665–674).

Mukherjee, A., & Liu, B. (2012). Aspect extraction through semi-supervised modeling. In Proceedings of the 50th annual meeting of the association for computational linguistics: Long papers-volume 1 (pp. 339–348).

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, *2*(1-2), 1–135.

Popescu, A.-M., & Etzioni, O. (2007). Extracting product features and opinions from reviews. In *Natural language processing and text mining* (pp. 9–28). Springer.

Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, *108*, 42–49.

Potamias, M. (2012). The warm-start bias of Yelp ratings. *arXiv preprint arXiv:1202.5713*.

Qiu, G., Liu, B., Bu, J., & Chen, C. (2011). Opinion word expansion and target extraction through double propagation. *Computational linguistics*, *37*(1), 9–27.

Sajnani, H., Saini, V., Kumar, K., Gabrielova, E., Choudary, P., & Lopes, C. (2012). The Yelp dataset challenge - Multilabel Classification of Yelp reviews into relevant categories. Retrieved from http://www.ics.uci.edu/~vpsaini/

Sauper, C., Haghighi, A., & Barzilay, R. (2011). Content models with attitude. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1 (pp. 350–358).

Snyder, B., & Barzilay, R. (2007). Multiple aspect ranking using the good grief algorithm. In *Hlt-naacl* (pp. 300–307).

Steinbach, M., Karypis, G., Kumar, V., et al. (2000). A comparison of document clustering techniques. In *Kdd workshop on text mining* (Vol. 400, pp. 525–526).

Titov, I., & McDonald, R. (2008). Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on world wide web* (pp. 111–120).

Toh, Z., & Wang, W. (2014). Dlirec: Aspect term extraction and term polarity classification system. In *Proceedings of the 8th international workshop on semantic evaluation (semeval 2014)* (pp. 235–240).

Wang, H., Lu, Y., & Zhai, C. (2010). Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 16th acm sigkdd international conference on knowledge discovery and data mining* (pp. 783–792).

Wang, T., Cai, Y., Leung, H.-f., Lau, R. Y., Li, Q., & Min, H. (2014). Product aspect extraction supervised with online domain knowledge. *Knowledge-Based Systems*, *71*, 86–100.

Yelp Dataset Challenge [Online]. (2016). Retrieved from https://www.yelp.com/dataset_challenge

Zhang, L., Liu, B., Lim, S. H., & O'Brien-Strain, E. (2010). Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd international conference on computational linguistics: Posters* (pp. 1462–1470).

Zhuang, L., Jing, F., & Zhu, X.-Y. (2006). Movie review mining and summarization. In *Proceedings of the 15th acm international conference on information and knowledge management* (pp. 43–50).