Gathering Competetive Intelligence From Restaurant Menus Using Natural Language Processing Based Techniques

By: Diya Saha

Problem Statement:

Menu analytics is an emerging area where the menus from two or more restaurants are compared using natural language processing to gain competitive intelligence about the restaurants. As a demonstration of this we conducted a menu analysis on the menus of two prominent restaurants in Bangalore, India to compare dishes and the similarities between the items. To conduct the analysis we used the item names and the descriptions and ingredients provided in the menus accessible to us.

Setting up the data:

To first set up the data, we manually collected from the publicly available online menus of the restaurants Bercos and Mainland china.

The menu for Bercos was found from their website. Since this menu was posted as an image, we used the optical character recognition(OCR) implemented on the iphone camera to transcribe the text from the images of the menu into a .txt file.

The menu for Mainland was collected by downloading the pdf provided on their website and then manually cleaning the characters in other languages and eventually converting it to a .txt file.

Analysis:

To conduct the analysis we are mainly using the python dictionary data structures. By combing through all the item names we broke the entire menu into two dictionaries.
For example this is the first five items of both the dictionaries

```
edamame dumplings with truffle oil: ()
corn and water chestnuts dumplings: ()
basil flavoured vegetable dumplings: ()
steamed cottage cheese dumplings with truffle oil: ()
crispy corn cubes: (piping hot cubes of creamy crunchy lightly spiced corn)
```

Fig.1. First five item of the Mainland menu dictionary

```
vanilla ice-cream : (two scoops)
darsan : (crispy noodles sweetened with honey and served with vanilla ice cream)
brown zebra: (rich and velvety eggless chocolate brownie topped with hazelnut)
date and walnut roll : ()
chocolate spring roll with ice cream: ()
```

Fig.2.First five items of the Berco's menu's dictionary

By creating this dictionary it allows us to traverse through all the different items in the menu and access any item's ingredients using the name of the item. To start formulating a plan to compare both menus we need to figure out a general idea of the entire data set. To do this, we break down the entire data set into dictionaries according to the categories listed in the original menu and count how many different dictionaries have been formed. From that we get the statistics in Fig.3.

	Mainland	**Bercos**
Number of items:	154	154
Number of categories:	24	23

Fig.3. High Level Statistics for both menus

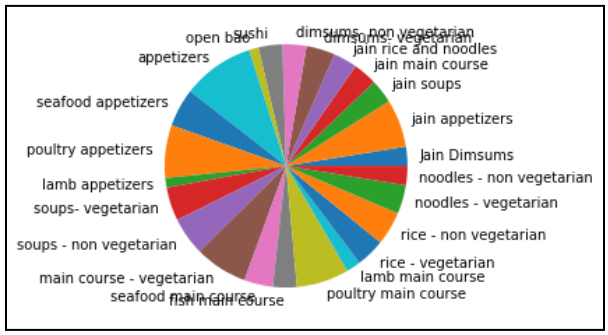

Fig.4. Item distribution from Mainland China's original menu

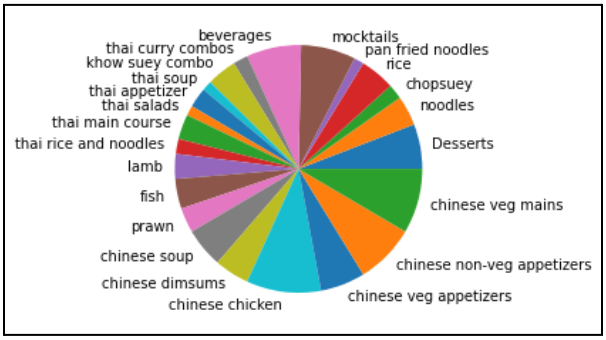

Fig.5. Item distribution from Berco's original menu

Fig.4 and Fig.5. are used to show the distribution of items in each of the categories listed. As we can see from these graphs there are too many categories to distinguish the number of items in each of them. Since each menu has over 20 categories we can reduce the categories by grouping them into bigger sections based on their similarities. This would make it more manageable to accurately understand the distribution.

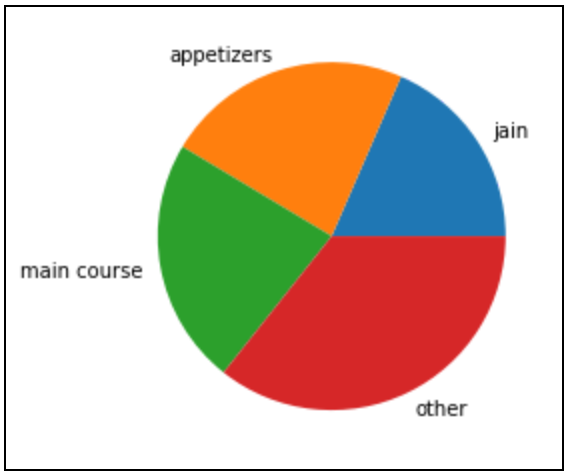

Fig.6. Mainland main category breakdown

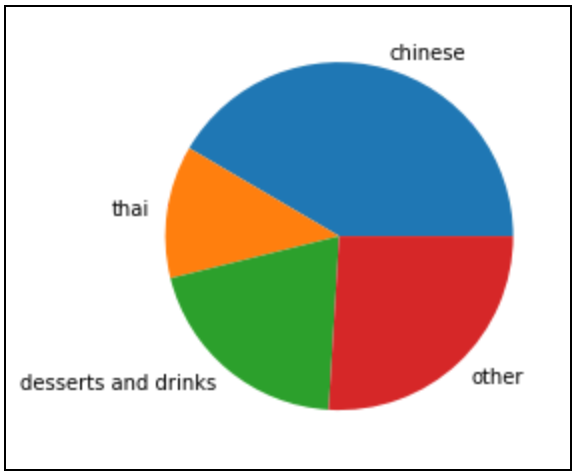

Fig.7. Berco main category breakdown

Manually looking at the different categories, we could see 4 very apparent groups. For Mainland, it was Appetizers, Jain items, Main course items that were not Jain and items that did not fit into any of those three: others. Alternatively, for Berco's menu, there were Chinese and Thai items, desserts and drinks and items that did not belong to the other three: other. This breakdown was more clear and made it easier to view the distribution a little better.

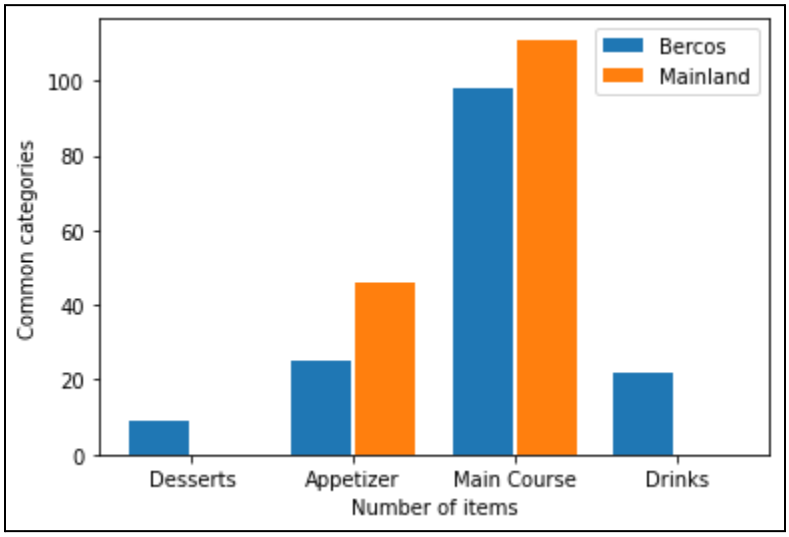

Fig.8. Comparing number of items in each category for each menu

One of the problems that we faced was that the categories from each of the restaurants were not the same, and therefore could not be compared with each other. We further separated the categories in Fig.6. And Fig.7. into more common groups like Desserts, Appetizers, Main Course items and Drinks. Since both menus have the same categories we can compare them side by side. Another problem faced in this stage was that one of the menu(Mainland) was missing a couple of the categories.

As seen in Fig.8. Mainland's menu did not have any items in the Desserts and Drinks category. One of the main reasons might be because the restaurant might have a separate dessert or drinks menu that we do not have access to.

For our purposes we decided to use the appetizer items and main course items and compare them further. To understand if there are any similar items or how to approach the comparison we used an excel sheet to manually compare them first.

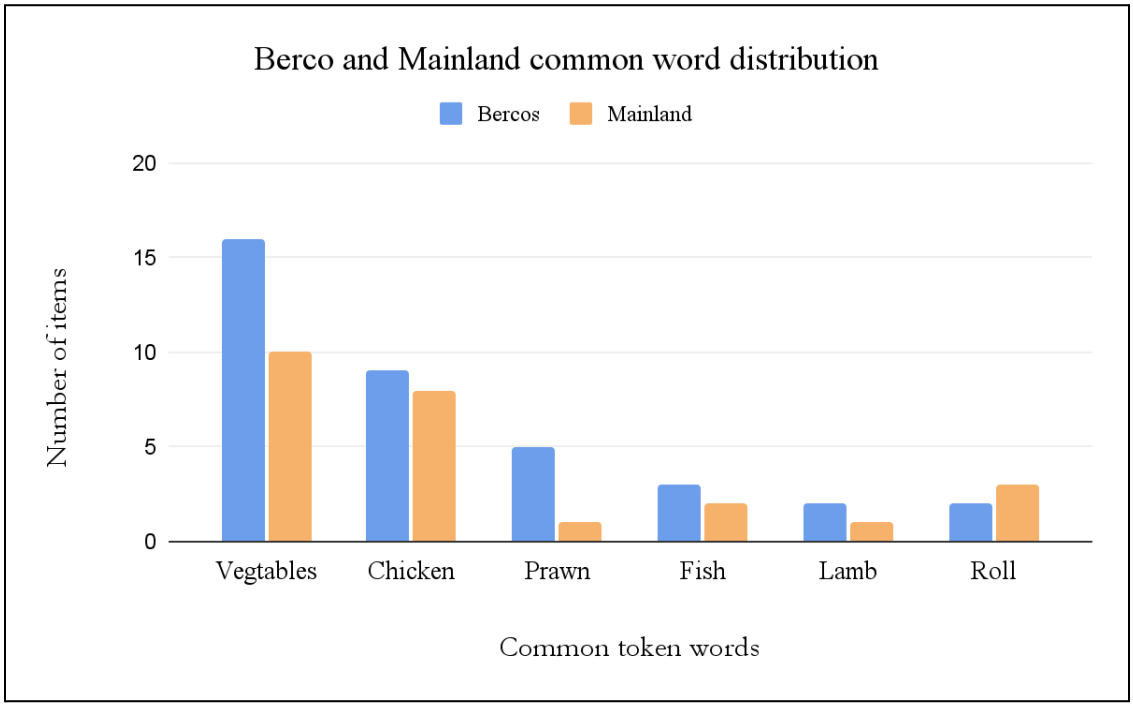

Fig.9. Manually finding the common token words in each item on Google Sheets

By manually comparing them we spotted some common tokens in all the items. These words were mainly roll, fish, prawn, chicken, lamb and vegetable. Since each token was present in both menus we could successfully use the Jaccard's Distance formula to an extent.

Jaccard Index:

Jaccard Index or Jaccard similarity Coefficient is a method of comparison that can be used to compare word tokens. This method treats the data objects like sets and is defined as the size of the intersection of two sets divided by the size of the union.

To calculate the Jaccard index, we first started by breaking down the description into tokens for each item in the dictionary. While processing these tokens, we got rid of punctuations so that the matching could be focused on just the words. When we first processed there were some issues that we came across.

```
get_item_info('Crispy Corn Cubes',mainland_app , berco_app, 0.1)

ITEM:  crispy corn cubes
ITEM DESCRIPTION:  (piping hot cubes of creamy, crunchy lightly spiced corn)
number of matches:  3
Threshold used:  0.1
------------ Items matched: ----------------
name:  crispy chilli potatoes
desc:  (in-house speciality of crispy potatoes tossed which spicy sauce)
jaccard index:  0.1111111111111111
----------------------------------------
name:  corn pepper salt
desc:  (juicy corn kernels spiked wich freshly crushed pepper)
jaccard index:  0.1111111111111111
----------------------------------------
name:  juicy chicken drumsticks
desc:  (popular dish of spicy chicken drumsticks seasoned with pepper and garlic)
jaccard index:  0.1111111111111111
----------------------------------------
```

Fig.10. First iteration of the Jaccardian index implementation

As we can see in Fig.10. the item we were matching was Crispy Corn Cubes and the function found 3 matches with a threshold of 0.1. This is a very low threshold, but for the purpose of this example we can use it to see a difference. When we closely look at the matches found we can see that the item's descriptions are not very similar. One is a potato dish, a corn dish and a chicken dish. This is caused because the function is matching words like "and", "with", "tossed", etc. This makes it a problem because these words are present in the description for most of the items. To overcome this problem by removing stop words from the descriptions. When the tokens were being processed, we added a list of stop words like "and", "with", "in", "for", etc that would be disregarded.

```
get_item_info('Crispy Corn Cubes',mainland_app , berco_app, 0.1)

ITEM:  crispy corn cubes
ITEM DESCRIPTION:  (piping hot cubes of creamy, crunchy lightly spiced corn)
number of matches:  1
Threshold used:  0.1
------------ Items matched: ----------------
name:  corn pepper salt
desc:  (juicy corn kernels spiked wich freshly crushed pepper)
jaccard index:  0.125
----------------------------------------
```

Fig.11. Second iteration of the Jaccardian index with stop words implemented

Now we can see in Fig.11. that the search is more refined after adding the stop words. The function only found 1 match with a threshold of 0.1 and the description matches a lot better to the item given.

Key Findings:

Initially, the menus were harder to compare as the categories from both the menus did not align with each other. To compare them correctly we needed to manually categorize them with our own categories and this improved our ability to compare the different items. To further improve the comparison using Jaccard Index, we implemented the use of stop words, this meant that those particular words like of, with, in, on, etc. would not be taken into consideration while comparing the description of the items.

Next Step:

To strengthen the comparison, we plan to implement sentence embedding algorithms, to implement semantic similarities between menus. This can help us prioritize certain words over others. For example, we can prioritize words like Chicken, Noodle, Fish, etc, which will help improve our comparing implementation. Grouping common terms together like Tossed and Stirred together would also improve the algorithm.

Finally, we plan to build a network connecting the items together through branches. We want to build an interactive web page where the client(Chefs) can enter certain keywords of different items and the restaurant's names to figure out the most popular food on the market. Senior chefs can use this to help restaurants to build their our menu or help them market their brand better.

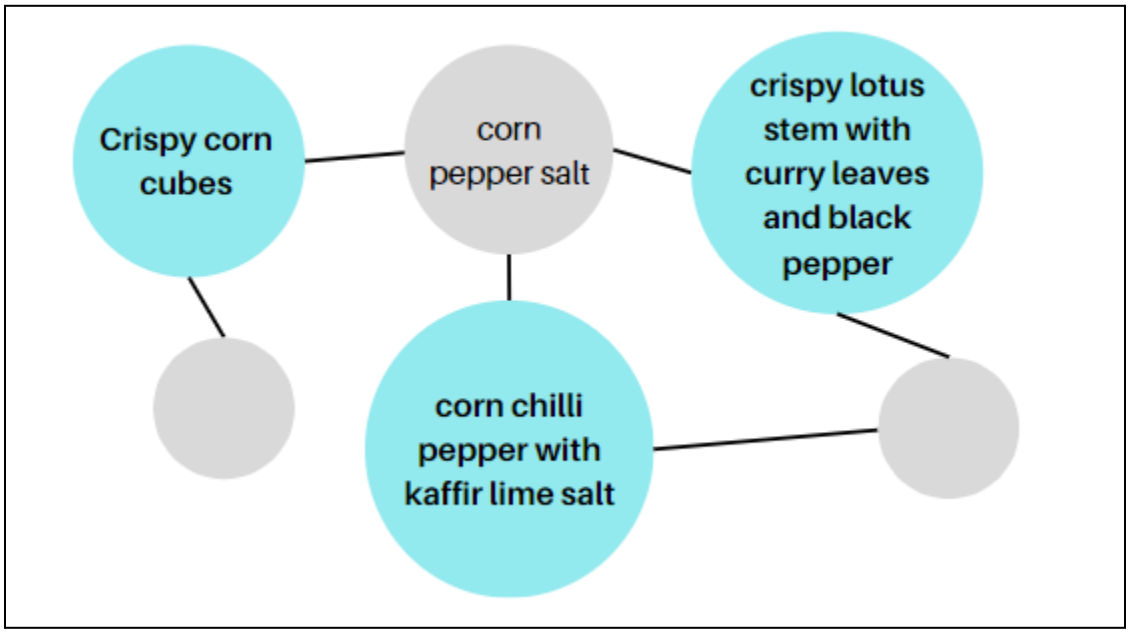

Fig.12. Prototype of the model

Acknowledgement:

I would like to thank DSights for providing the opportunity to conduct this research on this interesting problem of menu analytics. This project helped me understand the different issues faced during real life data collection and how to conduct data cleaning, (e.g. tokenization). I also learned and researched different natural language processing techniques like the Jaccard's Index and the Euclidean Distance. Overall this project taught me how to conduct research, gather and analyze publicly available data and develop my technical writing skills.

Reference:

1. Author: Fatih Karabiber Ph.D. in Computer Engineering, & Fatih Karabiber Ph.D. in Computer Engineering. (n.d.). Jaccard similarity. Learn Data Science - Tutorials, Books, Courses, and More. Retrieved September 17, 2022
2. Beltis, A. J. (2022, April 22). *Menu analysis: Why analyze your restaurant menu*. Menu Analysis: Why Analyze Your Restaurant Menu - On the Line | Toast POS. Retrieved September 17, 2022, from https://pos.toasttab.com/blog/on-the-line/menu-analysis
3. Berco's - menu - chinese & Thai restaurant, Delhi. Berco's - Menu - Chinese & Thai Restaurant, Delhi (n.d.). Retrieved September 17, 2022
4. Menu of Mainland China Bistro. Menu of mainland China Bistro. (n.d.). Retrieved September 17, 2022
5. PyShark. (2022, February 27). Jaccard similarity and Jaccard distance in python. PyShark. Retrieved September 17, 2022