DietChef

AN INTELLIGENT DIET SYSTEM

Intelligent Reasoning Systems
Practice Module
Semester II 2021/2022

Prepared by:

Chiu Man Shan
Zhao Lutong
Kuch Swee Cheng
Chen Hao

CONTENTS

1. EXECUTIVE SUMMARY

DietChef seeks to provide all users a simple, reliable and transparent recipe cookbook platform. Users can access and interact with DietChef easily via Telegram. DietChef would score the recipe from the database and provide the meal calories, ingredients, cooking instruction for the user. DietChef also seek to help you achieve your health goals by providing your calorie intake making your nutritional logging easier. DietChef would take your previous recipe preference and item inventory to recommend your next cooking experience, saving you time and effort to visiting multiple recipe website that fits your taste preference and inventory.

Project DietChef will provide a viable product for user to access and interact on Telegram, a widely popular Messaging System platform. The key highlights of DietChef are using Elastic Search for storing data and providing real-time search, recommend a recipe based on similarity score and YOLOv5 for object detection. Using YOLOv5 object detection, users are able to quickly onboard their ingredients to search for a recipe which best matches what they have.

We also realised the world produces enough food to nourish every man, woman and child on the planet. However, there is still roughly USD$1 trillion dollars of food that are wasted and lost because of edible food that is thrown away by consumers. According to USDA's Economic Research Service[1], roughly 30 to 40 percent are wasted in the United States alone. At the consumer level, people buy more than they need and then throw out unused food because they are unsure what to cook with the unused food. This amounts to more than 240 pounds of annual food wastage per person. We hope Dietchef can be the platform to rise the people awareness and provide a bridge to allow them to step out. We believe this small step can be a big step tomorrow.

[1] https://www.usda.gov/foodlossandwaste/faqs

2. PROBLEM DESCRIPTION

2.1 Statement

With the rise of more people turning to home cooking in the COVID-19 pandemic lockdown, research survey from participants has reported a decline in their diet quality, mainly caused by poorer dietary choices leading to other non-COVID health issues during the pandemic. Part of this is caused by cooking fatigue, where people run out of ideas on what to cook with their ingredients and turned to easy meals such as fast food or highly processed food.

Survey participant also reported that current market recipe applications struggle to yield desirable recipe result when their input are vague and ambiguous. Internet source recipe usually provide ingredient matching algorithm to match the user query. However, these recipe lacks clarity in listing its nutritional value and help health-conscious user monitor their calorie intake for monitoring and achieving their health goals.

As a result, in order to address this issue, we want to develop an intelligent recipe recommendation system which is health focused and positively impact users dietary pattern for a healthy post-pandemic lifestyle.

2.2 Objective

Unify recipe searches for a one stop experience- Offers the user a multitude of cuisine recipe, saving the user time and effort to search for cooking recipe on the internet.

Promote healthy dietary pattern- Helps user to positively impact their health through dietary means by calculating their calorie and nutritional intake. Which allow the user to achieve their health goal by more accurately determine and track their calorie and nutritional intake.

Reduction in food wastage- Offers the user our ingredient-based search engine to best utilise their current food inventory.

2.3 Proposed Solution

Our proposed solution to tackle the food insecurity issue is to use an AI chatbot on Telegram to gather users food consumption pattern through initial ingredients detection using computer vision and cooking recipe recommendation system as a potential way to combat food wastage.

The system will focus on learning and recommending recipes sourced from online recipe cookbook and allow user to submit their personal recipe to the community and maintained and vetted by our community of cooking enthusiast volunteer in an open collaboration manner before listing the recipe for the community.

3. MARKET RESEARCH

Since the beginning of the Covid-19 pandemic, access to food mall has been restricted, and people are spending more time staying at home (which includes work from home, digital education and limitation of outdoors physical activity). According to The Straits Times, nearly half of the Singaporeans (50%) polled in an online survey of 1,000 respondents have cut back on physical activities, and 73 per cent dine out less often. Similarly, a study conducted by Bloomberg News and Morning Consult revealed that almost a third of respondents (31%) claimed they will increase their home cooking frequency even the stay-at-home regulation is repealed, while only 7% said they will cook less once the economy reopens[2]. These market feedbacks indicate a potential demand for cooking at home services and we target to devote more efforts to transforming cook into a more intelligent action with the help of technology.

Based on our analysis of the present market for recipe application (e.g. YAZIO, Keto Diet App, etc), it can be found that the food recommendation function of these products are either searching the recipe in their database or searching online. It just involves the ingredient matching algorithm to simply provide the possible results, implying that only comprehensive user input could lead to the ideal result but if the user input is vague, the app is hard to predict the optimal recipe. Therefore, in order to tackle this pain point, we target to provide an intelligent solution for the recipe recommendation which will provide the best value for people's post-Covid lifestyle.

Besides, according to ADM's research (i.e. one of the world's largest nutrition companies), the pandemic has made more people interested in foods that benefit their immunity and metabolism[3]. The survey found that 57% of global consumers are more concerned about their diet nutrition balance because of the pandemic. Also around worldwide, authorities and healthcare professional's recommendations on how to stay healthy during the COVID-19 pandemic, are related to healthy diet-style measures such as assuring eat plenty of fresh fruits and vegetables, ensure sufficient protein intake and reduce to ultra-processed food. Therefore, the calory/nutrition tracking shall be a goal feature for us to implement.

With the purpose of better supporting our product design and future promotion, we also conducted our own market research by carrying out user survey in Singapore and interviewing the domain experts in nutrition science and Internet of Things (IoT) Engineering.

---

[2] https://www.bloomberg.com/news/articles/2020-07-07/newly-minted-home-chefs-mark-another-blow-to-u-s-restaurants
[3] https://www.fooddive.com/news/consumer-trends-shifting-toward-health-and-wellness-adm-finds/584388/

3.1 User Survey Result

Based on our survey with 165 respondents, we were able to obtain a general feeling on people's dieting habits change after the Covid-19 pandemic and their preference about the recipe recommendation features. The survey sample consists mainly of people aged between 25-45 years old (i.e., 69%) with approximate same gender distribution and 72% of respondents expressed that they would cook at home more often after the pandemic. Hence, from the potential user analysis, it can be inferred that most user groups are the young adults who lack of cooking skills and will need cooking guidance from the recipe. Such target users have a high consumption mindset and capacity, which could empower our recipe recommendation products development.

Regarding people's dieting habits under current post-Covid environment, 41% of respondents will rotate between similar food and 34% of respondents eat similar foods with some variation. However, there are around 75% of respondents are open, willing or strongly willing to intelligent recipe recommendation system based on their dietary habits/ preference, instead of searching online every time. Therefore, the demand for the recipe recommendation will be considerable and significant.

For the main features of this recommendation systems, 23% of respondents are willing to have recipe recommendation functions according to their diet preference and 28% of respondents would like to have all our intended design features from easier data input to recommendation output. Besides, there are several other ideas provided by the respondents which include incorporating this system into the Intelligent home appliances (e.g., smart fridge), Create online community to interact with other users and providing cooking lessons

3.2 Dietitian Interview

Based on our interview with an accredited dietitian Mr Jeffrey Lew, he shared his team's research on Singaporeans' dieting habits and the results suggested an association between customers' diets and unhealthier eating habits during the COVID-19 lockdown/restriction. The food contained vegetables decreased by 15%, while the probability of food is barbecue/fried food or beverage category increased by 11% and 4%, respectively. Also, most food ordering/community APP only provide the basic data for one certain recipe after user requesting for it, hence, people will mostly ignore to check for their daily meal.

According to his working experience, he suggested we shall consider the similarity between different food and make the intelligent recommends based on user's preference so that to catch users' long-term interest. The determining factor of the similarity between different food will be the ingredients combination, which means if there will be more overlapping ingredients, these two recipes will be more similar. We accepted Mr. Jeffery's advice and will be more focused on the similarity study on the recipe database.

3.3 AI Engineer (IoT) Interview

We also had a chance to interview Mr Huang Weijie, who works as an AI engineer in the Technology Company. He shared that after the epidemic, the "cooking fever" and "stocking tide" have caused people's demand for smart appliances such as refrigerators. The concept of health priority has received continuous attention, bringing new opportunities for product and service innovation to industries such as diet and life services. The intelligence of the home appliance industry is an important development trend for enterprises, and this incremental market is a blue ocean for traditional home appliance enterprises.

Regarding the intelligent recipe recommendation system incorporation into the smart fridge, he highlighted that it is necessary to systematically improve the user's refrigerator experience in combination with the specific user's preference which means the recommendation system should be "intelligent" enough to recommend the food as per user's history data. On this basis, it would also provide the necessary food data for caring user's health condition. We are agreeable with Mr Huang's suggestion and the user's history data shall be one of the key factors considered in our model.

4. SYSTEM MODELLING

4.1 Data Pre-processing

4.1.1 Information extraction from original database

There are two datasets used in this project. One is the recipe dataset which is sourced form Forkify API and another is the common food ingredient nutrition database from Google research Database (nutrition5k ingredient metadata). For the recipe dataset, it contains 2256 recipes and 7704 original ingredients. However, in order to have a more comprehensive corresponding ingredient's nutrition information, we used fuzzy matching method to create the "nutrition key" which stores the keywords connected the recipe dataset and standard nutrition dataset (i.e. nutrition5k ingredient metadata). Also due to the complex frames of original recipe database, we reorganized our databases into a structured knowledge representation which could carry more information in an organized manner. The Figure 1 shows the main attributes of the structured recipe dataset.

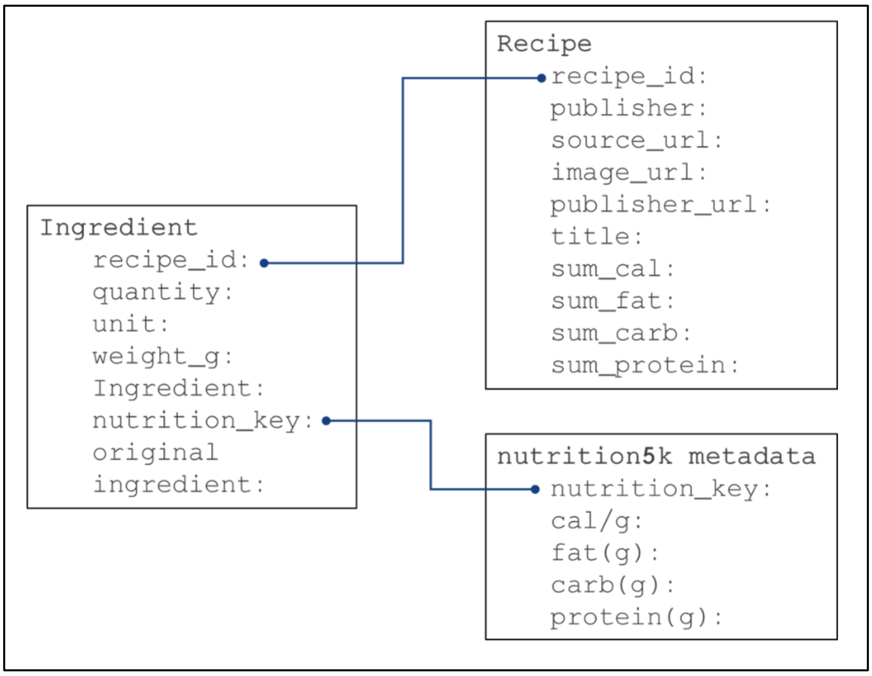


Figure 1 The attributes of the structured recipe dataset

4.1.2 Data cleaning

This system also accepts the user input channel to get the ingredient information. Since the user input is in text format, hence we would use the Spacy package to pre-process the text input. Once the clean and spelling corrected words are obtained, we will extract the key words on the ingredient and the calories number from these pre-processed words. Hence all the key words will be passed to the next step for further processing.

4.2 Object Detection Using YOLOv5

YOLOv5('You Only Look Once') is a popular single shot real-time detector known for its fast inference speed and reasonable accuracy. The reason for using YOLOv5 object detection function in this project is to reduce the hassle for users to manually input their ingredient into the search recipe. To offer the best user experience and a less time-consuming way in DietChef, user may take a picture or a photo and upload it to our chatbot in Telegram for ease of onboarding ingredients in their fridge.

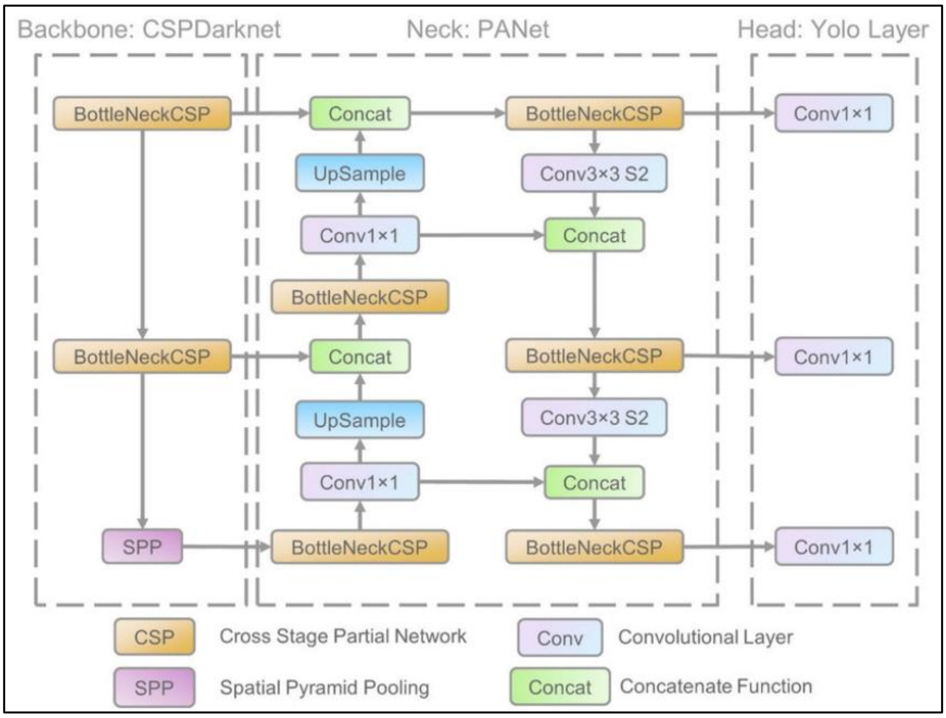


Figure 2 YOLOv5 Model Architecture

YOLOv5 was chosen as our model for the following reason:

1. Model Backbone - Extract key features using of an object using CSP(Cross Stage Partial Networks), where useful characteristic are extracted from a feature rich image especially in the use case where our refrigerator are usually stacked and crowded. This allows for feature propagation to combat vanishing gradient problem, by reusing the features found in the network.

2. Model Neck – To create the feature pyramid and help the model to generalize in terms of object scaling. This helps in the identification of object of various size and scale, couple with augmented training photo where images are rotated. This allows us to mimic the way refrigerator are stacked, where food is placed in an overlapping group setting with ingredients in different augmented position.

4.2.1 Training on the custom dataset using OpenImagev6

Using YOLOv5, we are able to pre-train our model using custom dataset images from OpenImagesv6. OpenImagesv6 contains a vast number of well annotated images with up to 600 boxable object classes. Through our generated ingredient classes “nutritional_key” attained during our recipe pre-processing, we are able to extract and match the necessary ingredient object classes found on OpenImagev6 required for object detection.

Out of the 430 variety of ingredient pre-processed from the recipe dataset, we could matched up to 65 explicit ingredient found on the OpenImagesv6 database to be used for training on the object detection model. These data are parse as dataset.yaml to the model for custom dataset training.

```
import fiftyone as fo
import fiftyone.zoo as foz
from difflib import get_close_matches
import numpy as np

# List of approved labels in the fo dataset
# fo_list=fo.utils.openimages.getclasses()
fo_list=openimages.get_classes()
avaliable_fo =list()
i = 0
while i < len(ingridentList):

    matches=get_close_matches(ingridentList[i],fo_list,n=2,cutoff=0.75)

    avaliable_fo.extend(matches)
    i += 1
    avaliable_fo.sort()
print("Number of ingredient classes avaliable on OpenImageV6: ",len(avaliable_fo))
print(avaliable_fo)
```

```
Number of ingredient classes avaliable on OpenImageV6:  65
['Apple', 'Artichoke', 'Banana', 'Beer', 'Bell pepper', 'Bread', 'Broccoli', 'Cabbage', 'Cantaloupe', 'Carrot', 'Cheese', 'Chicken', 'Coat', 'Coconut', 'Coffee', 'Cookie', 'Crab', 'Cream',
'Croissant', 'Cucumber', 'Deer', 'Dice', 'Drum', 'Duck', 'Fish', 'Fork', 'French fries', 'Goat', 'Grapefruit', 'Guacamole', 'Hamburger', 'Hamburger', 'Hot dog', 'Ice cream', 'Lemon',
'Lobster', 'Maple', 'Milk', 'Muffin', 'Mushroom', 'Orange', 'Oyster', 'Pancake', 'Parrot', 'Pasta', 'Peach', 'Pineapple', 'Pizza', 'Plate', 'Platter', 'Pomegranate', 'Popcorn', 'Poster',
'Poster', 'Pretzel', 'Pumpkin', 'Seafood', 'Shrimp', 'Squid', 'Sushi', 'Turkey', 'Watermelon', 'Wheel', 'Wine', 'Zucchini']
```

Figure 3 Matched ingredient list on OpenImagev6

4.2.2 Inference speed using YOLOv5

Inference speed is one of the identified key factor which influence to user’s experience in many of the object detection application leading to design success and failure. Therefore, our custom dataset from OpenImagesv6 are pre-trained in YOLOv5, which predicts both the class labels and bounding boxes simultaneously. This is far superior than Faster R-CNN using a sequential two stage object detection model cutting down inference time.

	YOLO v5	**Faster RCNN**
Inference Speed	✓	
Detection of small or far away objects	✓	
Little to no overlapping boxes	✓	
Missed Objects	✗	✗
Detection of Crowded objects	✓	✓

Figure 4 YOLOv5 and Faster RCNN comparison

4.3 Word Embedding with TF-IDF (Method 1)

4.3.1 Ingredient level embedding

After pre-processing, there are 430 categorized ingredients in our recipes. To capture the semantic similarity on the recipe level, we used Word2Vec to encode each recipe as one embedding. For example, in our processed ingredients:

```
In  [312]: word_vectors.most_similar("cheese")

Out[312]: [('ham', 0.9882979393005371),
            ('italian dressing', 0.9874359965324402),
            ('cheddar cheese', 0.9868790507316589),
            ('goat cheese', 0.9852006435394287),
            ('pepperoni', 0.9849198460578918),
            ('asparagus', 0.9847971796989441),
            ('egg whites', 0.9846328496932983),
            ('provolone cheese', 0.9835293292999268),
            ('bacon', 0.9824573397636414),
            ('broccoli', 0.9823499917984009)]
```

Figure 5 The test results for the most similar/matching ingredients of "cheese"

We can see that the model can capture the similarity between "cheese" and "cheddar cheese" and "provolone cheese", recipe embeddings with these words will be closer in the space. We are providing both CBOW/Skip-Gram models for ingredient embedding, since both of them are trained by looking at the context (neighbours), it can handle different naming conventions for same ingredients (like eggplant and brinjal). It also provides the user with more flexibility: if the user does not have the same type of cheese, we can still recommend recipes than use other cheese alternatives.

However, the contextually trained model indeed has drawbacks. As illustrated in Figure 5, the ingredient identified as "most similar" to cheese is "ham" and this is mainly due to their high probability of co-occurrence in recipe corpus. However, "cheese" is not semantically close to "ham". This is a common drawback we have seen for word embedding training on a small corpus. Therefore, a future improvement in semantic capturing will be including more recipes and more ingredients in our training datasets.

4.3.2 TF-IDF on ingredients

For recipe recommendations, it is common to see "oil", and "salt" on the ingredient list which may appear very often but is not very useful information. We want to reduce the impact of these common items on embedding similarity calculation by introducing TF-IDF.

As defined in formula of IDF (Inverse Document Frequency) (details in section 4.6), IDF will measure the rareness of each ingredient and assign a smaller weight for common words.

Based on our corpus, we trained a TfidfVectorizer from the sklearn library and got an idf_weight matrix for each ingredient. Then multiply it with the original word embedding from CBOW/Skip-gram.

```
ingredients.append(
    self.word_model.wv.get_vector(word) * self.idf_weight[word])
```

In our recipe corpus:

```
In [347]: sorted(tfidf_vec_tr.word_idf_weight.items(), key=lambda x: x[1], reverse=False)[:10

Out[347]: [('oil', 1.8170410068556975),
           ('zzseasoning', 1.874848637232156),
           ('salt', 1.8844589621319445),
           ('pepper', 1.9238513638426054),
           ('onions', 2.075401261969806),
           ('sugar', 2.130118044808878),
           ('cheese', 2.193833859194985),
           ('butter', 2.21750360328089),
           ('olive', 2.2220047361616815),
           ('garlic', 2.2295519417970646)]
```

Figure 6 The test results for ingredients with lowest IDF value

```
In [348]: sorted(tfidf_vec_tr.word_idf_weight.items(), key=lambda x: x[1], reverse=True)[:10]

Out[348]: [('dumplings', 8.02864459625759),
           ('stews', 8.02864459625759),
           ('veggie', 8.02864459625759),
           ('alfalfa', 8.02864459625759),
           ('spreads', 8.02864459625759),
           ('gouda', 8.02864459625759),
           ('focaccia', 8.02864459625759),
           ('tostadas', 8.02864459625759),
           ('collards', 8.02864459625759),
           ('oysters', 8.02864459625759)]
```

Figure 7 The test results for ingredients with highest IDF value

We can see for ingredients like oil, salt, and seasoning, the IDF values are the lowest and less commonly seen ingredients like “alfalfa” and “dumplings” have much higher IDF value. Therefore, by multiplying the IDF value by original word vector, we amplify the weight for more rare items and weigh less on the more commonly used ingredients (e.g. seasoning, salt, oil).

4.4 Weighted Ingredient Vector (Method 2)

Besides the drawbacks of word2vec embedding method trained on small corpus (aforementioned in section 4.3.1), we can see that this ingredient level embedding is using the mean value of each ingredient embedding multiplied by the IDF weight. There is another issue when we are encoding the ingredient embedding: the simple word2Vec model is unigram and when we pass through multi-word ingredients like "veggie burger", it will be treated as "veggie" and "burger" separately and multiplied by their separate IDF weights.

To solve these two issues, we designed a more intuitive but powerful way to represent each recipe. Instead of using the 50/100 latent features trained form 2-layer CBOW/Skip-gram, in case of recipe recommendation where there is no unknown value for each ingredient (they are just 0), we will use the ingredients (could be bi-gram or N-gram) as the explicit features and the quantity (in gram) is a perfect weight for this model. The rows are recipe_ids and columns are different ingredients, the value/weight is the quantity of the ingredients in gram.

4.5 Recipe Similarity Modelling

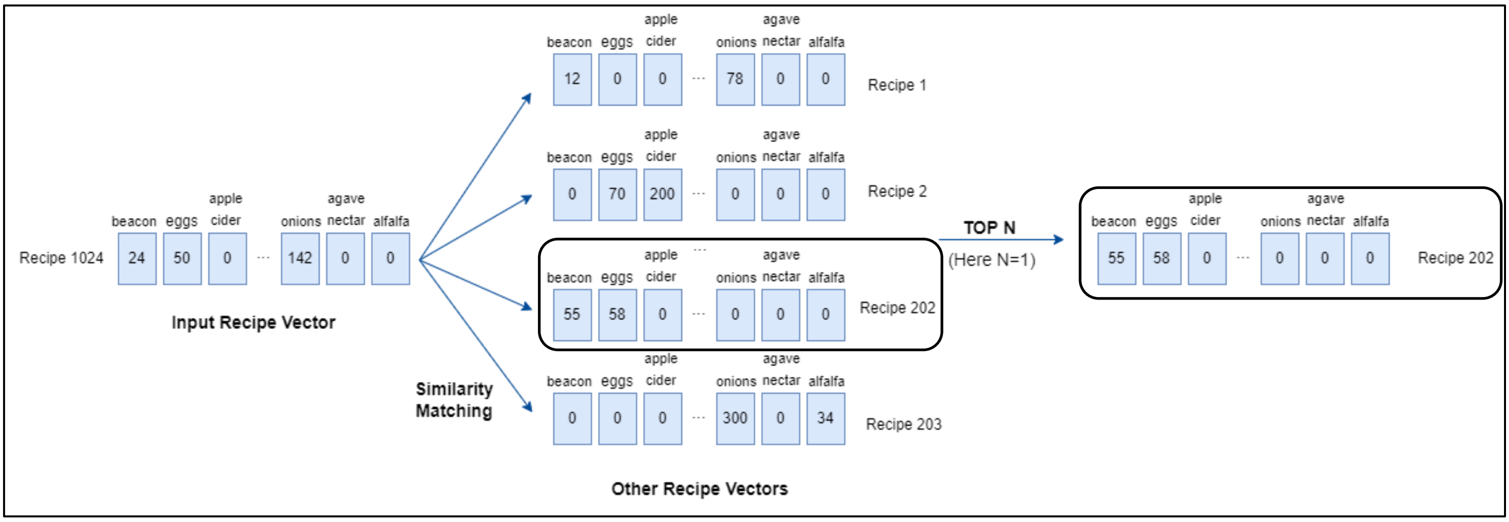


Figure 8 Working flow of recipe similarity calculation

4.5.1 Item-Item similarity

For recipe-to-recipe recommendation, we used sklearn's version of cosine_similarity and euclidean_distances to calculate the similarity scores. As illustrated in the aboce diagram, for both weight-based and word vector embedding model, it will calculate the similarity score based on different metrics and sort top N recipes with highest scores.

For current recipe vectors V_c and all other recipe vector V_i in the database, with k latent features(Method 1) /number of ingredients (Method 2):

The formula for cosine_similarity metric is

$$score(V_C, V_i) = Cos(\theta) = \frac{V_C \cdot V_i}{\| V_C \| \| V_i \|} = \frac{\sum_{n=1}^{k} V_{c,n} V_{i,n}}{\sqrt{\sum_{n=1}^{k} {V_{c,n}}^2} \sqrt{\sum_{n=1}^{k} {V_{i,n}}^2}}$$

The formula for euclidean_distances metric is

$$d(V_C, V_i) = \sqrt{\sum_{n=1}^{k} (V_C - V_i)^2}$$

$$score(V_C, V_i) = \frac{1}{1 + d(V_C, V_i)}$$

4.5.2 Different similarity metrics comparison

We have tested both metrics on both methods. In Method 1, from Figure 9 and 10, we could see that, for recommendation both on “Buffalo Chicken Grilled Cheese Sandwich”, the results have two common results “Bacon Wrapped Hamburgers” and “Bacon Double Cheeseburger Soup”, while euclidean distance scores range slightly lower than cosine similarities scores.

```
Current Recipe: Buffalo Chicken Grilled Cheese Sandwich
['barbecue sauce', 'blue cheese', 'bread', 'butter', 'carrot', 'celery', 'cheddar cheese', 'chicken', 'mayonnaise', 'onions']
similarity function used: cosine_similarity
[1483, 20, 514, 1347, 1434]
```

Out[76]:

	score	recipe_id	recipe	ingredients	sum_cal	sum_fat	sum_carb	sum_protein	url
0	0.99912	2296	Bacon Wrapped Hamburgers	[bacon, barbecue sauce, buns, cheddar cheese, ...	2085.652361	NaN	66.372425	92.816117	http://allrecipes.com/Recipe/Bacon-Wrapped-Ham...
1	0.99906	35109	Bacon Double Cheeseburger Soup	[bacon, barbecue sauce, beer, broth, buns, car...	2786.469743	NaN	169.503424	117.066765	http://www.closetcooking.com/2012/09/bacon-dou...
2	0.99885	35069	Ale and Cheddar Soup	[bacon, barbecue sauce, bell peppers, butter, ...	1913.284979	NaN	63.446049	105.137771	http://www.closetcooking.com/2012/03/ale-and-c...
3	0.99885	a723e8	Barbecue Chicken Pizza	[barbecue sauce, cheese, chicken, onions, oran...	1227.25558	NaN	62.899807	70.039665	http://www.mybakingaddiction.com/barbecue-chic...
4	0.99871	e40950	Grilled Mac & Cheese with BBQ Pulled Pork	[barbecue sauce, butter, cheddar cheese, macar...	878.544	NaN	108.4792	37.0436	http://paninihappy.com/grilled-mac-cheese-with...

Figure 9 The test results for recommendation using Method 1 (cos_simi)

```
Current Recipe: Buffalo Chicken Grilled Cheese Sandwich
['barbecue sauce', 'blue cheese', 'bread', 'butter', 'carrot', 'celery', 'cheddar cheese', 'chicken', 'mayonnaise', 'onions']
similarity function used: euclidean_distances
[1484, 1483, 20, 1486, 184]
```

Out[77]:

	score	recipe_id	recipe	ingredients	sum_cal	sum_fat	sum_carb	sum_protein	url
0	0.7916	30009	Spicy Chipotle Turkey Burgers	[barbecue sauce, buns, cilantro, garlic, groun...	1947.829115	NaN	157.55585	132.449093	http://allrecipes.com/Recipe/Spicy-Chipotle-Tu...
1	0.77783	2296	Bacon Wrapped Hamburgers	[bacon, barbecue sauce, buns, cheddar cheese, ...	2085.652361	NaN	66.372425	92.816117	http://allrecipes.com/Recipe/Bacon-Wrapped-Ham...
2	0.77772	35109	Bacon Double Cheeseburger Soup	[bacon, barbecue sauce, beer, broth, buns, car...	2786.469743	NaN	169.503424	117.066765	http://www.closetcooking.com/2012/09/bacon-dou...
3	0.77532	35108	Bacon Double Cheese Burger Salad	[bacon, barbecue sauce, buns, cheddar cheese, ...	2074.168611	NaN	116.459173	104.607952	http://www.closetcooking.com/2012/04/bacon-dou...
4	0.76058	694798	Leftover Easter Sandwich	[avocado, barbecue sauce, bread, ham, hard boi...	2076.4057	NaN	199.788	72.4403	http://thepioneerwoman.com/cooking/2013/04/eas...

Figure 10 The test results for recommendation using Method 1 (eucl_simi)

In Method 2, from Figure 11 and 12, we could see that for recommendation on "Cowboy Quiche", the recommended results are almost the same (4/5 are the same) for different similarity metrics. Besides, although two metrics are resulting in different scoring range, it does not affect the order of scores, therefore the recommendation results are not affected.

```
Generating similar recipes for recipe:Cowboy Quiche
['cheddar cheese', 'bacon', 'eggs', 'cream', 'butter', 'pepper', 'pie crust', 'onions']
similarity function used: euclidean_distances
[460, 131, 1122, 76, 348]
```

Out[46]:

	score	recipe_id	recipe	ingredients	sum_cal	sum_fat	sum_carb	sum_protein	url
0	0.01101	35533	Roast Turkey Sandwich with Artichoke Tapenade,...	[roast turkey, large onion, bread or 2 buns cu...	275.004	NaN	33.3776	10.4456	http://www.closetcooking.com/2012/03/roast-tur...
1	0.01049	36561	Mango Salsa	[medium red onion, fresh lime juice, ripe mang...	107.7172	NaN	24.7478	4.3981	http://www.simplyrecipes.com/recipes/mango_salsa/
2	0.01024	5993	Caribbean Fruity Salsa	[medium red onion, diced fresh pineapple, dice...	166.802	NaN	29.5005	4.3563	http://allrecipes.com/Recipe/Caribbean-Fruity-...
3	0.00972	35540	Roasted Cauliflower and Aged White Cheddar Soup	[garlic, aged white cheddar, medium onion, sma...	496.7676	NaN	45.38	16.776	http://www.closetcooking.com/2012/02/roasted-c...
4	0.00905	983c2a	Loaded Veggie and Prosciutto Pizza	[artichoke hearts, cherry tomatoes, sharp ched...	703.258457	NaN	103.39437	20.749837	http://whatsgabycooking.com/loaded-veggie-and-...

Figure 11 The test results for recommendation using Method 2 (cos_simi)

```
Generating similar recipes for recipe:Cowboy Quiche
['cheddar cheese', 'bacon', 'eggs', 'cream', 'butter', 'pepper', 'pie crust', 'onions']
similarity function used: cosine_similarity
[460, 1567, 131, 1122, 76]
```

Out[47]:

	score	recipe_id	recipe	ingredients	sum_cal	sum_fat	sum_carb	sum_protein	url
0	0.9115	35533	Roast Turkey Sandwich with Artichoke Tapenade,...	[roast turkey, large onion, bread or 2 buns cu...	275.004	NaN	33.3776	10.4456	http://www.closetcooking.com/2012/03/roast-tur...
1	0.9065	38403	Lamb Kebabs With Lima Bean Salad	[garlic, medium red onion, red wine vinegar, l...	631.367611	NaN	86.826781	26.709664	http://www.realsimple.com/food-recipes/browse-...
2	0.90614	36561	Mango Salsa	[medium red onion, fresh lime juice, ripe mang...	107.7172	NaN	24.7478	4.3981	http://www.simplyrecipes.com/recipes/mango_salsa/
3	0.89978	5993	Caribbean Fruity Salsa	[medium red onion, diced fresh pineapple, dice...	166.802	NaN	29.5005	4.3563	http://allrecipes.com/Recipe/Caribbean-Fruity-...
4	0.8843	35540	Roasted Cauliflower and Aged White Cheddar Soup	[garlic, aged white cheddar, medium onion, sma...	496.7676	NaN	45.38	16.776	http://www.closetcooking.com/2012/02/roasted-c...

Figure 12 The test results for recommendation using Method 2 (eucl_simi)

It is also interesting to note that this could also be another differentiator for these two models from Method 1 and Method 2: The evaluation metrics should give similar / comparable results when using different similarity metrics. Therefore, from this perspective Method 2 are better than Method 1 on recommendation results.

4.6 Text Search Scoring Algorithm

We evaluated text search recommendation and similarity function in two areas below and decided to go with text search**.**

1) Most of the users wouldn't know what they want exactly, and the similarity function wouldn't work without the full list of ingredients.
2) We must use exact feature to train the similarity model like "blue cheese" so the search engine would miss those recipes if users search "cheese" but vice-versa the text search would consider both "blue cheese" and "cheese".

Also, we would like to consider the how important the ingredients are in a recipe based on the weights Instead of doing simple text search on the ingredients as it can give more accurate and specific search experience to users. For example, "whole chicken" contains ingredients "chicken(500g) salt(5g)" and "chicken rice" contains ingredient "chicken(100g) rice(100g) egg(50g)", so the two recipes would have the same score when users search "chicken" if we just consider the ingredient, but the "whole chicken" should be more relevant. Therefore, we parse the into a paragraph and repeating it based on the weight. For example, considering the "chicken rice" recipe above, the lightest portion here is 50g then we would divide each ingredient by 50g, and the paragraph would be "chicken chicken rice rice egg". After that we use TF-IDF scoring algorithm to calculate the relevance between the search term (ingredient) and the existing recipes using this field.

$$repeat\ times = \frac{Weight\ of\ the\ ingredient}{Weight\ of\ lightest\ ingredient}$$

TF-IDF is a fair technique to quantify the importance of a word in a document by considering the frequency and the popularity of the word. We would do information extraction from the user input to make sure the search term only contains the ingredients we have in our database then split the list into single word, calculate the respective TF-IDF score and get the sum of score as total.

$$TF - IDF(score) = \sum_{wi} tf * idf$$

Term Frequency (TF) is the frequency of a word appears in the ingredient list.

$$tf(score) = \frac{f}{Total\ number\ of\ ingredients\ inthe\ recipe}$$

Inverse Document Frequency (IDF) is the to take the most common/rare ingredients into account. If the ingredient appears many times which means it is less important vice-versa it would be relatively important if it only appears in few recipes.

$$idf(score) = \log\left(\frac{N}{df}\right)$$

f = Number of occurrences of the ingredient in recipe
N = Total number of recipes
df = Number of recipes containing the ingredient

4.7 Rule Based Information Extraction

We first identify the user action by looking at the start word which is in the format "/xxx". The action must be in our action list then we would ask again if the user request is not found. After that we apply NLP to remove stop words and punctuations then apply following pre-defined rules to and extract the information.

No.	Rule	Description
1.	Check if the word token exists in our unit database If yes -> if previous token is a number then combine two token into single one. If no -> remove this token	To group unit and number together for later process.
2.	Check If the word exists in our ingredient database If yes -> then we try to combine this token and previous token and repeat the check again	We need to check with the previous token as for some ingredients like "blue cheese", it will be spitted into two token and so the recursive method would help to get exact ingredient.
3.	If current word token is ingredient then and the previous token contains "unit" in our database. If yes -> extract number and unit If no -> ignore	To identity the portion of ingredient.

4.	Check if the token contains “cal,kcal,calorie” If yes -> this is the calorie limit	To identity calorie limit.
5.	If the unit is not in “g” then we try to convert it into “g” based on the conversion list we have. If nonconvertible -> then remove it	Our system is using “gram” as our golden unit and we have to convert everything into it.

5. SOLUTION

5.1 System Design

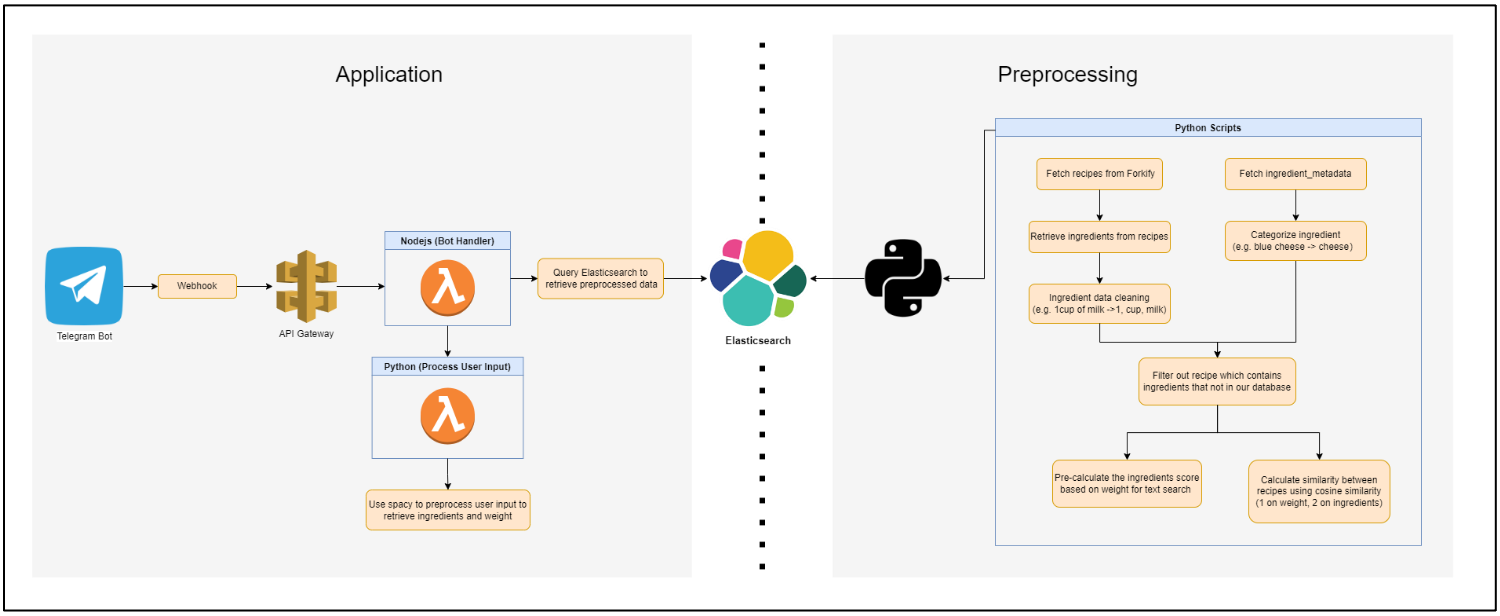


Figure 13 The System Architecture

5.1.1 Serverless Architecture

We are using AWS Lambda which is an event driven system, and it provides high availability and scalability. Our application can be easily scale up or down depending on the number of active users we have. We can also save the cost as we don't need 24 hours long running server to just listen to the telegram web hook which is more expensive.

5.1.2 Elasticsearch

Elasticsearch is a distributed document-oriented search engine, designed to store and retrieve unstructured data quickly. Our recipe application data doesn't have much relationship and we pre-processed the into well oriented structure and quick response time is the most important part to us and so Elasticsearch is the best fit in this case.

5.1.3 Telegram bot

Telegram bot is one of the famous communication platforms in the world and it can provide easy access to the users to kick start and try out our application. Telegram backend will send a webhook to our backend when there is a new message send to our bot.

5.2 Process Flow

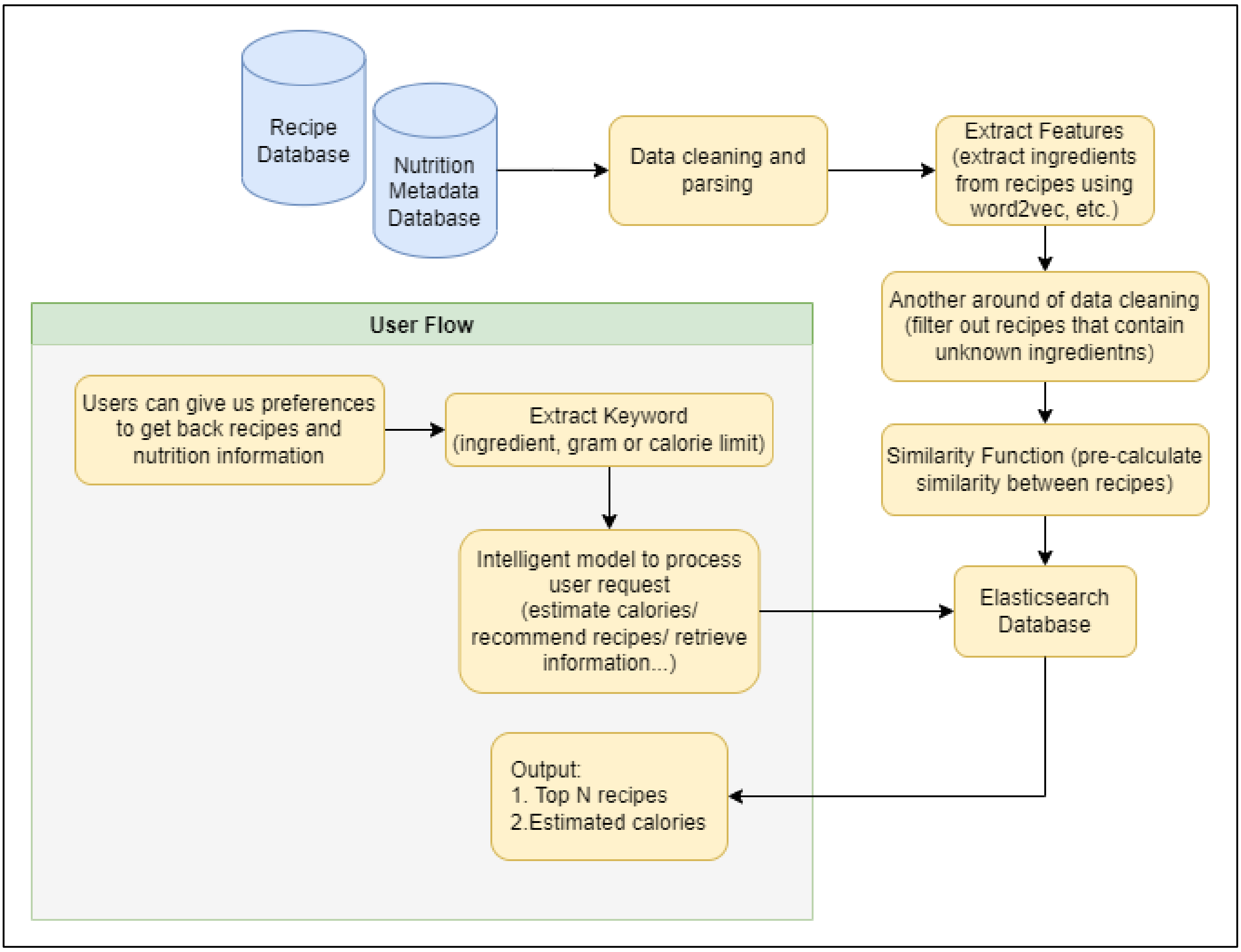


Figure 14 The Overall Process Workflow

5.2.1 Data cleaning

Firstly, we fetch the data from data source then do the data cleaning like eliminating some recipes without ingredients detail. After that we would extract all the features from the data like ingredients and its weight. Thirdly, we would do another around of data cleaning to filter out all the recipes with unknown ingredients in our database.

Detail steps

- Fetch recipes from Forkify
- Fetch ingredients metadata
- Retrieve ingredients from recipes
- Ingredient data cleaning

- Category ingredients
- Filter out recipe which contains ingredients that not in nutrition database
- Ingredients object detection
- Calculate popularity of ingredients in a recipe using naive bayes
- Project ingredients into a paragraph based on weight of text search
- Calculate similarity between recipes using Cosine similarity

5.2.2 Data modelling

Once we have the well-structured data, we would perform the modelling logic to calculate the similarity between recipes and prepare the data to be ready for text search later.

5.2.3 Data storage

We then push all the processed data to Elasticsearch, and it is ready for the bot to use.

5.2.4 User interaction

When a user talks to the bot, telegram backend would send a webhook to our backend then we would extract the information and query Elasticsearch to get back the related recipes.

5.3 Future Enhancement

Inventory management system – Incorporate new inventory management system to allow user to update product information such as date of expiry of their ingredient, and sauces into our system. This could allow us to inform you when you have up and coming ingredient that needs to be consumed by its best before date.

Community engagement – To further foster DietChef community in their passion for cooking. We would be inviting top star chef in the community to do a life telecast of cooking show for everyone to follow along. Shopping links of the ingredients, sauces and cooking utensils used by our star chef during the live telecast, would also be provided to the community as a tie-in with our sponsors.

Dietary education – In collaboration with Health Promotional Board, we would be rolling out the "NutriStar" series to educate our community. Guest nutritionist would join the show to share with our community how our dietary plans and choices could impact your health goals.

Integration into Health app - To further integrate with the iOS platform in their Health App, by allow ease of uploading nutritional intake into the Health App.

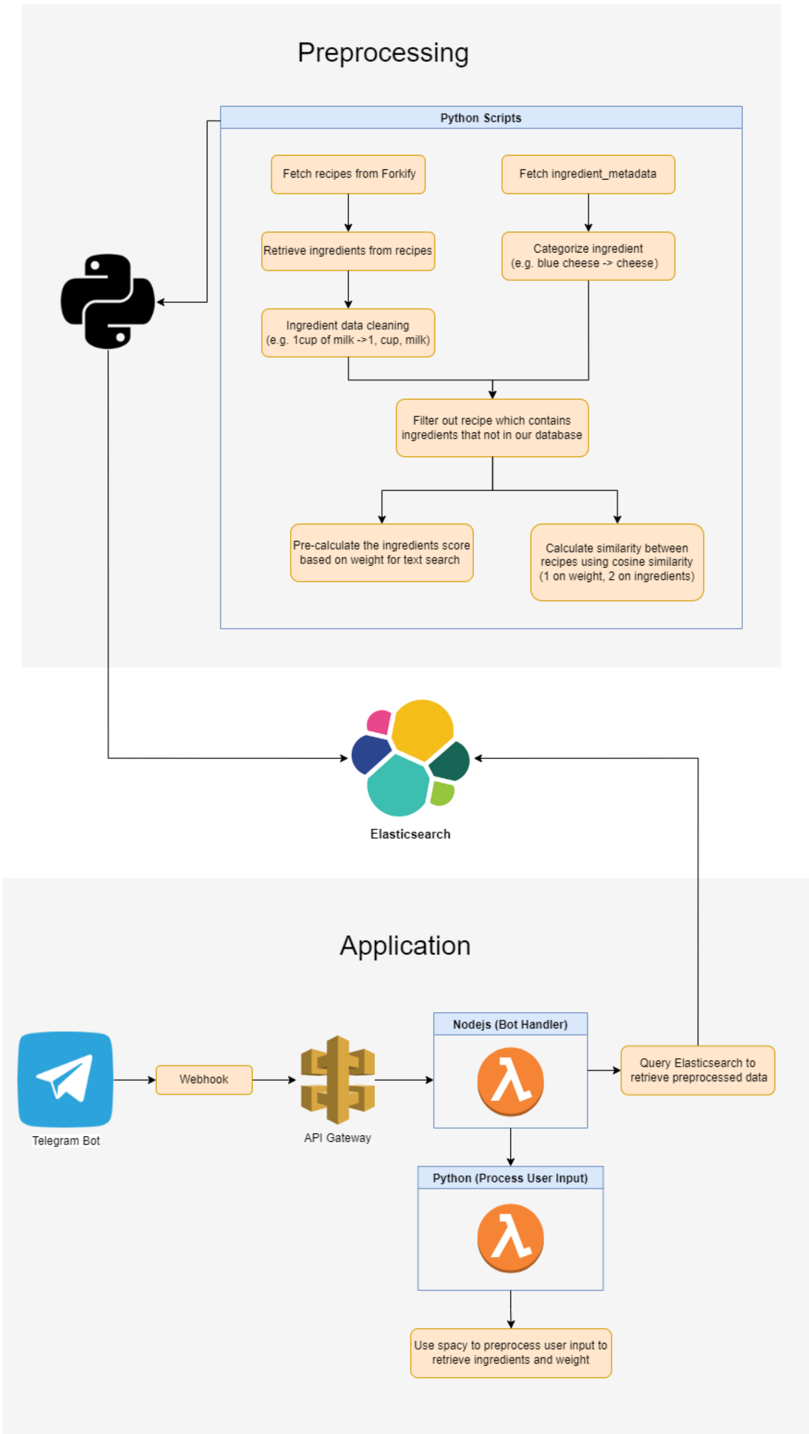
APPENDIX A

Project Proposal

Date of proposal: 28/04/2022
Project Title: DietChef - An Intelligent Diet System
Group ID (As Enrolled in LumiNUS Class Groups): Group 5 **Group Members (Name, Student ID):** Chen Hao (A0146967L) Chiu Man Shan (A0249252A) Kuch Swee Cheng (A0249246X) Zhao Lutong (A0249279L)
Sponsor/Client: None
Background/Aims/Objectives: Unify recipe searches for a one stop experience: • offers the user a multitude of cuisine recipe • saving the user time and effort to search for cooking recipe on the internet. Promote healthy dietary pattern: • Helps user to positively impact their health through dietary means by calculating their calorie and nutritional intake. • allow the user to achieve their health goal by more accurately determine and track their calorie and nutritional intake. Reduction in food wastage: • offers the user our ingredient based search engine to best utilise their current food inventory.

Project Description:

We would consider the recipes at “Ingredients” level as this is the necessary data. We need to calculate the nutrition value. We would also use weighted ingredients to search as well as calculating the similarity between recipes. Here is our first draft of solution.



APPENDIX B

Mapped System Functionalities against knowledge, techniques and skills of module

Module Cources	Knowledge
Machine Reasoning	1. Knowledge representation 2. Rule-based system 3. Word2vec 4. Text pre-processing with Spacy
Reasoning Systems	1. Cosine similarity 2. Item-item based similarity 3. Content based similarity
Cognitive Systems	1. TF-IDF 2. Structured knowledge representation

APPENDIX C

User and Installation Guide

Github User and Installation Guide

APPENDIX D: INTERVIEWS

Interview with Mr Jeffrey Lew, an accredited dietitian

1. Could you please introduce about your job scope as a dietitian, since for us, this kind of job seems like very newly emerging?

   Working as a dietitian in a local hospital after returning to Singapore made me realized that nutrition and dietetic services were more concentrated at the hospitals and polyclinics. But the health literacy level of overall Singaporeans is relatively low compared to other developed county. Hence, my job is to help the public equip with the knowledge and power to make healthful decisions. I can help my clients lead a healthier lifestyle through food choices and behaviour modification.

2. According to your recent work experience, do you find any change on Singaporeans' dieting habits after the COVID-19 pandemic?

   Yes, actually we did research on this issue and the results suggested an association between customers' diets and unhealthier eating habits during the COVID-19 lockdown/restriction. The food contained vegetables decreased by 15%, while the probability of food is barbecue/fried food or beverage category increased by 11% and 4%, respectively.

3. Do you have any idea on the reason of this phenomenon?

   I would like to say various stressors brought about by the pandemic, including increased workload, retrenchment and longer work hours at home, as well as a surge in the availability of food delivery options will overwhelm individuals and cause negative emotions. So, when they are browsing the food recommendation/order APP, they are more likely to choose unhealthy food to destress. More importantly, there is no alert function highlighting the healthy food for people and asking them to make the correct choice.

4. So as you mentioned, in the current market, there are vacancies for a food recommendation system equipped with both nutrition contents and users' dietary habits?

Yes, as we can see in our daily life, most APP only provide the basic data for one certain recipe after user requesting for it. If that's the case, people will mostly ignore to check for their daily meal.

5. If we would like to create a recipe recommendation system including its basic statistic, what advice you may want to give?

   My daily job includes one important part which is to let my clients know their diet. Our dietetic consultation helps people understand their dietary habits and preferences and offer alternatives and portion advice to optimize your diet for your desired weight and nutrition goals. Hence I would suggest you could consider the similarity between different food and make the intelligent recommends based on user's preference.

6. From your point of view, what is the key component of the similarity between different food as you mentioned just now, like ingredients or flavour?

   I reckon the determining factor of the similarity between different food will be the ingredients combination, which means if there will be more overlapping ingredients, these two recipes will be more similar.

7. If AI is utilised in the daily food recommendation system, what concern will you have?

   First, I would suggest having a database verified by domain experts such as dietitian or nutritionist. Second, if the machine recommendation does not match user's expectation, I would advise to have a feedback channel to revise your system algorism. Third, I would wish this technology is cheap and accessible, so the world will benefit from it.

Interview with Mr Huang Weijie,
an AI engineer in the Technology Company

1. Could you please make a general update for us on the AI and IoT development in household appliance industry?

   Today, technology has evolved to such an extent that there's a possibility to design meaningful collaboration between humans and machines, primarily due to advancements in AI. Moreover, digital assistants, also called virtual assistants, work on voice-controlled AI, which can do functions like searching the internet, making calls, and connecting to other devices. These assisted devices can be embedded into smartphones or can also be used as a standalone device. This is a great approach to enhance and automate with working of home appliances. IoT is one of the key focus areas of digital transformation projects in the consumer electronics and home appliances industry.

2. What is the current market demand for the intelligent household appliance after the COVID-19 pandemic?

   After the epidemic, the "kitchen fever" and "stocking tide" have caused people's demand for smart appliances such as refrigerators. The concept of health priority has received continuous attention, bringing new opportunities for product and service innovation to industries such as diet and life services. The intelligence of the home appliance industry is an important development trend for enterprises, and this incremental market is a blue ocean for traditional home appliance enterprises. With the gradual improvement of the national consumption level, the differentiation and personalization of user groups are more obvious. Refrigerator companies urgently need to further subdivide the market to make products meet the differentiated needs of users.

3. If we would like to incorporate an intelligent recipe recommendation system into the fridge, what will be suggestion on the critical selling point for this system?

   Refrigerator products in the existing market are seriously homogenized. Refrigerators need targeted innovation, and it is necessary to systematically improve the user's refrigerator experience in combination with the specific life habit of users. Therefore, research on the characteristics of user situations has become an important way to innovate smart refrigerator product design. Taking the recipe recommendation system as an example, it shall be "intelligent" enough to recommend the food as per user's history data. On this basis, it would also provide the necessary food data for caring user's health condition.

4. Will this function work well in the household IoT system?

   Yes, in the process of using this system, the relevant information generated by the user and the corresponding influencing factors between user and the system interaction will achieve better target for understanding the user needs/preference from a context-aware perspective.

APPENDIX E: INDIVIDUAL REPORT

Chiu Man Shan

Personal Contribution	1) System Architecture 2) Text Search Algorithm 3) Rule based information extraction 4) Telegram Chatbot 5) Project deployment
Lessons Learned	For text search, I tried to use naive bayes classifier as a scoring algorithm however it doesn't work well in our use case as we have to consider weight of the ingredients and some heavy recipe's ingredients are dominating. I learnt that naïve bayes cannot be used to compare different documents as they are in different weight. It can only be used in a single document to extract the features. I also tested using "term frequency" **only** scoring algorithm to calculate the score on the user preference. Again, as we consider the weight of ingredients, we would always get the same result as some ingredients are dominating and it's frequency is very high. Therefore, I went to do more research on the text search scoring and understand how important the IDF is. However term frequency can work well if your contents are evenly distributed. For information extraction part, I learnt that spacy is a good tool to help us to pre-process the text however we cannot 100% rely on it and we have to use it case by case. I tried to use spacy to retrieve ingredients and it would split the sentence into tokens which is word by word. The problem here is the ingredient may not be always a single word like "blue cheese", the spacy result will be ["blue","cheese"] In order to get the full ingredient name, we have to some rules like it looks like ingredient then check with the previous token. The last part is we explored different ways to find the similarity between recipes and user preferences. I understand more when should we use item-item, user-item as well as the content based recommendation. I realised that the data play the most important part here and i think most of the time, the real life scenario would be restricted by the data they have but not the way they want to go with.
Future	For my company, there are lots questions come from slack every day, I can extract the features from their questions then categorise it into few big areas using the text-pre-processing technique. Then use similarity function to find similar wiki pages to my team to check if it is relevant. It can help us to reduce sometime to process user

	questions and also finding documents in wiki. If it is a technical problem, I can also use the bot to get back who asked the similar questions before and recommend the people who solved the problem before to them.

Zhao Lutong

Personal Contribution	1) Implement Content-based Recommendation Engine - Word Embedding with TF-IDF model - Gram weight-based recipe vector model 2) Implement similarity matching evaluation using two distance metrics 3) Retrieve the top N highest scoring recipes, aggregate their nutrition metadata information 4) Pre-compute most top-N recommended recipe for all existing recipes in the database on a regular basis and store it as JSON in Elastic Search
Lessons Learned	1) We can use IDF weights on word embedding to reduce the impact of common tokens (less important tokens). Since word embedding is good at capturing semantic similarity between different words by looking at the semantic context of the words, it may be common to see common ingredients like “oil”, “salt”, “seasoning” have very high term frequency and have correlation with many ingredients. But these ingredients are less important when we want to compare between different recipes. Therefore, we want to reduce the impact of these too commonly seen words without dropping all of them. (Since it is still a factor to consider for recipe) By computing the IDF weights, we are able to see how often this ingredient also appear in other recipes. Therefore, we can use a IDF-weighted word-embedding by assign higher weights to the more rare ingredients. 2) Neural Network/more complex model does not always outperform simple models: To encode recipe according to their ingredients into a vector, I firstly tried the Word2vec model as it is the most popular model to deal with text inputs. It could capture the semantic similarity between ingredients well. However, it is actually not **considering the quantity** of each ingredient, which is crucial to differentiate recipes. Therefore, we proposed a simpler model which is actually more powerful at capturing difference particularly to recipe encoding task. In the word2vec model, we are using predefined K (we used 50 and 100) number of latent features to capture the semantic meaning on this k space. In the second model, instead of using latent features, we used each ingredient as explicit features, and the weights as the value. This model does not require training at

	all, but the model build based on it can deliver even better recommendation result than the first model. From this I realize, to fully understand what is the most suitable for the use case is the important. Complex model is not always the better solution.
Future	There are lot of application for the recommendation knowledge that I have learnt in my company. There are many spending data which includes spend amount, spend categories, shop names and spend date. We could recommend user to the shops that users who share similar consumer portfolio like. To build up the consumer portfolio, we could use very similar way in this project: we can pass the text input of shops' name in a fixed period of time for each user. Then build a word2Vec model to encode similarity between each shops. Then also times its IDF weights to reduce the impact of shops that everybody will go to. Then use two different similarity metrics to calculate the pair-wise similarity scores. Lastly, select top-N most similar user and recommend their visited shops as the recommendation for the current user.

Chen Hao

Personal Contribution	1) Market Research and Business Analysis 2) Recipe Database Collection 3) Database Pre-processing and Structure Arrangement 4) User Input Keyword Extraction 5) Report Writing and Video Preparation
Useful Learning	The first thing I learned was how to use Python. I started the IRS module with only a basic understanding of Python, but thanks to this project, I was able to enhance my scripting abilities and write more sophisticated programs like keyword extraction (NLTK), API calls, and database restructuring. Secondly, this project enabled me to know the actual application of machine learning technique. Since our designed system is more specific on Data Processing as well as Rule-Based system (RBS), it brought me through the whole process from business analysis to knowledge representation and finally we can work together to implement and test the system. It provided me with a better understanding on the techniques used in the real-life system development. For the data processing collected from Forkify API, I pre-processed the text into standard ingredient words and used the fuzzy matching method to make the "nutrition_key" to connect the ingredient words with the standard keywords in the nutrition database. After that, I tried to reorganize the original data structure into 3 different parts using "recipe id" and "nutrition_key" as the reference index. Hence in the later stage, my group mates can collect the data easily from my modified database. In order to achieve the purpose of user keyword extraction, RBS concept has also been applied into this function. I managed to introduce the rules based on the dataset and coded the rule engine to process the user input. With this approach we can have a more comprehensive coverage to tackle various user input. Finally, I attempted to stay in touch with all of my teammates throughout the process, so that I could learn new things from them and have a better grasp of the typical system design working flow, which included market research, brainstorming, collecting field data, and final review and implementation. In addition, in the role of a coordinator, I kept the track of progress to ensure each objective can be met.
Future	Since currently I am working in a research laboratory, my colleague and I need to handle many research materials in the form of scientific articles and journals. Before carrying out our own research, we are required to study sufficient published papers. After taking this semester's courses, I came up with the idea to apply the topic

	modelling to categorize these literatures. So, I coded up a script in Python to test my design concept. The test results proved to work well and the model separated the documents into 4 topics which is generally correct based on my personal judgement. However, the model still needs to be modified specifically on different topics since the frequency of some certain words varies in different topic. I plan to continue working and studying on NLP in the future since human language is a hugely complicated subject and training a computer to comprehend text and voice is a fascinating subject to me. And in the cognitive system courses, I understanded that NLP is also widely used in speech recognition algorithms and Chatbots. Hence, I plan to use what I will learn to design a basic smart city interaction platform to understand the people's real needs (especially on urban resilience topic, which is my current research topic) for government's further management.

Kuch Swee Cheng

Personal Contribution	1) Implement ingredient object detection using YOLOv5 2) Trained custom image dataset extracted from matched ingredient list from pre-processed recipe using images from Google OpenImagev6 and fiftyone for image dataset exploration. 3) Explored using Nvidia Jetson Nano SBC, edge server image inference and object detection.
Lessons Learned	Through this journey, I have greatly benefitted from the opportunity to learning computer vision technique such as object detection and utilise it in actual project deployment with my teammates. This has allowed me a more in-depth understanding of its working principle and driven me to read up more outside of current course curriculum such as the evaluation metrics such as (mAP). The environment factors such as the light, resolution, or objects cascaded within objects also plays a big part in affecting the outcome of the detection. Real world images are always in flux and often not in
Future	I am currently working in a manufacturing industry, and I strongly believe that there is immense value in delving deeper into computer vision technique such as segmentation and classification. This would help in areas such as defect detection, which improves quality assurance and the first step to reduce the manpower reliance to observe and classify defects. I am also looking for opportunities to deploy video analytics solutions (security and compliance) to my workplace with the upcoming implementation of 5G infrastructure in the work place.