A TF-IDF Based Algorithm for Regional Food's Features Extraction and Its Application

Trung Duc Nguyen

Faculty of Environment and Information Studies

Keio University

5322 Endo Fujisawa Kanagawa 252-8520 JAPAN

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Art

Advisors:

Professor Kiyoki Yasushi
Diep Nguyen-Thi Ngoc

Abstract of Bachelor's Thesis

A TF-IDF Based Algorithm for Regional Food's Features Extraction and Its Application

Automatically detecting food's taste is a non-trivial part. However, we realize that the taste of food can be extracted by directly analyzing recipes by the ingredients and the amount of them in the recipes. In this paper, we present a food analysis system to discover the taste of foods and to better understand the featured ingredients in each specific geographical region. The main features of this system are (1) to extract dominant ingredients and tastes in a region by analyzing the ingredients' frequency and its uniqueness, and (2) to transform user's existing materials or original recipe to a new recipe according to a targeted taste. To examine the feasibility and applicability of the algorithm, we have developed a web-based application with a recipe database collected from approximately 200 recipes in over 8 regions of Japan: Hokkaido-Tohoku, Kanto, Kansai, Shikoku, Tyubu, Kyusyu-Okinawa and Tyugoku.

Trung Duc Nguyen
Faculty of Environment and Information Studies, Keio University

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We can observe that in geographical regions that are far apart from each other often have different features and tastes. For example, the Kanto region, which is located in the East of Japan, often has dense taste in its foods, while the foods in Kansai region, which lies in the southern-central region of Japan's main island Honshu, often has a diluted taste. The reason is because each region has its own special materials for foods and people in these regions have different habits in cooking food.

In this thesis, we present a food analysis system to discover the taste of food and understand the featured ingredients in a specific geographical region.

This chapter describes the goals of our research, the overview of the problem and also the structure of this thesis.

1.1 Challenges and Research Goals

To understand each region's featured taste we need to answer the following questions: "How can we understand the different features of each region's food?" and "What effects change the region's food taste?". Among the many factors that affect a food's taste, the combination of materials is a direct and important factor. Each recipe has a list of its own ingredients together with their amount. This leads us to the idea that we could automatically achieve the features of a region's foods by analyzing the materials. In this research, we use the idea of TF-IDF method, which is originally used for weighting words and documents, to propose an algorithm to extract the featured ingredients of regional foods.

We also realize that understanding the region's featured taste and the preferred materials has an application in supporting cooking activities. For example, imagine someone living in Kanto region who wants to eat some traditional foods in the Kansai region. They know the original recipe but there are some tastes in Kansai region that are not favoured. They would prefer that traditional foods with replaced ingredients that are easy for Kanto people to eat. Conversely, someone living in Kanto region might want to try Kanto foods with Kansai taste. Solving this kind of problem means we can build up a system which can help people satisfy their taste. The recipes, which are made by the system, would be flexible and diverse.

However, cooking is an sophisticated art, there is not a common formula for all recipes. In this research, we just propose a method, an new approach to evaluate food's taste through analyzing recipe's ingredients.

1.2 Structure of Thesis

The outline of this thesis is as following. The background of the TF-IDF method is discussed in chapter 2 while the related work to this research is discussed in chapter 3. The proposed algorithm and the experimental results are introduced in chapter 4 and 5 respectively. In chapter 6 we describes the web-based application using the proposed algorithm and how to design the food database. Chapter 7 concludes and discusses the remaining problems and future works.

Chapter 2

Background

2.1 Computer aid cooking activities

Cooking and eating have been the most fundamental activities of humankind since the days of trading caravans in the ancient days until now, which affect various aspects of human life such as health, dietary, human communication, safety of food, entertainment, culinary art, welfare, and so on. However, many people who cook at home require supports for cooking because it requires experience and knowledge. They may also need support for food-logging and menu planning for the health of their family. Needless to say, support for a good and enjoyable dinner would improve the quality of life. On the other hand, systematic cooking and eating support for the elderly and/or physically challenged people are significantly important.

Together with the development of technology and the availability of equipment in cooking, many supporting systems are introduced. For example, the cooking support system utilizing built-in cameras and projectors [1], the cooking support system by using ubiquitous sensors [2], the calorie measurement system by image processing [3] or the system which helps inexperienced users in understanding non-professional recipe descriptions [4], etc.

However, by using analysis job we can discover the dominant ingredients and tastes in foods and understand how to alter the taste from one to another.

2.2 The TF-IDF numerical statistic

In the existing cooking support systems, the methods vary such as image processing, text retrieval, sensing, etc. We use the text processing approach to directly analyze the recipes with their ingredients and amount of ingredients. In this research we use a famous method named TF-IDF, which is originally used for weighting word and documents.

In this chapter, we introduce background of TF-IDF method, the idea, applications, mathematical definition and problems in section 2.1, 2.2, 2.3 respectively.

2.2.1 Overview of TF-IDF Algorithm and Its Application

One of the earliest and most popular ways to create weighting vectors is the TF-IDF family of weighting schemes.

In 1972, Karen Sparck Jones published in the Journal of Documentation a paper called "A statistical interpretation of term specificity and its application in retrieval" [5]. The measure of term specificity first proposed in that paper later became known as inverse document frequency, or IDF; it is based on counting the number of documents in the collection being searched which contain (or are indexed by) the term in question. The intuition was that a query term which occurs in many documents is not a good discriminator, and should be given less weight than one which occurs in few documents, and the measure was an heuristic implementation of this intuition. The intuition, and the measure associated with it, proved to be a giant leap in the field of information retrieval. Coupled with TF (the frequency of the term in the document itself, in this case, the more the better), it found its way into almost every term weighting scheme. The class of weighting schemes known generically as TF*IDF, which involve multiplying the IDF measure (possibly one of a number of variants) by a TF measure (again possibly one of a number of variants, not just the raw count) have proved extraordinarily robust and difficult to beat, even by much more carefully worked out models and theories. It has even made its way outside of text retrieval into methods for retrieval of other media, and into language processing techniques for other purposes.

For example, say that we have a set of English text documents and wish to determine which document is most relevant to the query "a good man". A simple way to start out is by eliminating documents that do not contain all three words "a", "good", and "man", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document and sum them all together; the number of times a

term occurs in a document is called its term frequency.

However, because the term "a" is so common, this will tend to incorrectly emphasize documents which happen to use the word "a" more frequently, without giving enough weight to the more meaningful terms "good" and "man". The term "a" is not a good keyword to distinguish relevant and non-relevant documents and terms, unlike the less common words "good" and "man". Hence an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

Variations of the TF-IDF weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. TF-IDF can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

2.2.2 Mathematical Details

The inverse document frequency is a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

where $|D|$ is cardinality of D, or the total number of documents in the corpus and $|\{d \in D : t \in d\}|$ is number of documents where the term t appears (i.e., $\mathrm{tf}(t, d) \neq 0$).

If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the formula to $1 + |\{d \in D : t \in d\}|$.

Mathematically the base of the log function does not matter and constitutes a constant multiplicative factor towards the overall result.

Then TFIDF is calculated as

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

The formula above is the simplest way to implement TF-IDF weighting scheme. Different schemes are used in specific case depends on the problems the system solves. Our research also use the idea of TF-IDF scheme but specific problem of analyzing regional food's taste.

2.2.3 Related Problems to TF-IDF

Though TF-IDF is a robust weighting scheme, for different systems it is adapted in different ways and there are also according problems.

For the system in which the database of documents is often updated, typically new documents are received over time. In this case, the TF value of old documents are fixed, there is no need to recalculate but the IDF value certainly changed. Recalculation is necessary but choosing which mechanics

One option is to keep using the existing TF-IDF until a certain number of new documents have been received, and the recalculate it. But there are systems that require update instantly, these system will encounter the problem of massive calculation. Especially in our regional food's featured ingredient system, because of the way we apply the TF-IDF algorithm, the recalculation on updating data became more expensive. We will propose a method to solve this problems in chapter 4.

Chapter 3

Related Work

In our research, we propose a system that can allow user to register their available ingredients in their home and then recommend suitable recipes based on these ingredients. The system can also replace some ingredients in original recipe to get a new recipe that has targeted region's tastes.

In this chapter, we introduce some related works to these functions and also the related work to TF-IDF algorithm.

3.1 Recommending Recipe by Ingredients

Recipe recommendation and retrieval has been the subject of cooking related research. One of the earlier works is Kalas, a social navigation system for food recipe, developed by Svensson et al. [6]. Xie et al. [7] proposed a hybrid semantic item model for recipe search by example. The hybrid semantic item model represents different kinds of features of recipe data.

Another branch of research has focused on the recipe recommendation for healthy food. Mino et al. investigated the recommendation of cooking recipes for a diet in which the evaluation value of intake or consumption of calorie is considered in the events of a user's schedule during the period of a diet [8]. Linear programming approach is utilized with the constraints of carbohydrate, lipid, protein, salt, and increasing the amount of vegetable intake. Karikome and Fujii propose a system to help users for planning nutritionally balanced menus [9]. Considerations of recipes that correct the users nutritional imbalance are incorporated into the recipe retrieval process. Visualization of dietary habits are also provided by this system.

3.2 Replacing ingredients

Shidochi et al. proposed an approach to extract replaceable ingredients from recipes in to satisfy users' various demands, such as calorie constraints and food availability [10].

In order to develop a strategy for changing users eating and cooking behaviors, Pinxteren et al. proposed a user-centered similarity measure for recommendation of healthier alternatives which are perceived to be similar to users commonly selected meals [11]. The similarity measure can be used to promote new recipes that fit users lifestyle.

By considering the users cooking competence, Wagner et al. presented a context-aware recipe retrieval and recommendation system to motivate users for healthy food preparation [12]. The system tracks the users cooking activities with sensors in kitchen utensils and

recommends healthy recipes that may increase the users cooking competence.

Chapter 4

Food's Feature-Ingredient Extraction Algorithm

In this section, we propose an algorithm for analyzing the dominant materials which are often used in a region. We define a material in a region to be a featured one if it appears many times with a large amount and be unique among recipes in that region. To evaluate whether it is featured or not, we suppose that the following questions should be answered: "How often, how much, and how unique the material is?". Respectively, we propose three kind of functions to answer these questions. They have the key role of the metrics for the featured ingredient's evaluation.

4.1 Ingredient Frequency

The first function named IF (Ingredient Frequency) is used to treat the question "How often does the material appear in a region?". The higher frequency an ingredient appears in a region, the higher possibility it is the region's featured ingredient. In each recipe, an ingredient only appears one time. Thus, the time that ingredient appears in the region is the number of the recipes in the region has it as ingredient. Because the database we have from the Internet are often unbalanced, there are some regions that have more recipes than others. Thus to make it indenpendent from the database, we prefer to use the ingredient's

frequency rather than its appearance times. This function is formed by the number of times the ingredient appears in the region's recipes over the number of total recipes in that region. Let R be the set of all recipes (r) in a region and i be an ingredient which appears in the region. The function is formed as follows:

$$IF(i, R) = \frac{|\{i|i \in r, r \in R\}|}{|R|}$$

Becaue the IF value is the ingredient's frequency, it takes the value between 0 and 1.

4.2 Ingredient Amount

The ingredient's frequency has little meaning if there is a small amount of it in the recipes. Thus, the taste of a food not only depends on the ingredients, but also the amount of the ingredients. Even when an ingredient has a high value of IF, it might not be the region's featured ingredient. Thus, the second function, IA, is proposed for the question "How much?"

Let r be a recipe in the set of recipes S and ingredient i is in r. We define the mean function $M(i, S)$ be the mean amount of i in S as follows:

$$M(i, S) = \frac{\sum^{i \in r, r \in S} amount(i, r)}{|\{i|i \in r, r \in S\}|}$$

in which $amount(i, r)$ is the amount of ingredient i in recipe r.

We also assume that AR is the set of all recipes in the country regardless of the region it belongs to, while R is the set of all recipes just in a specific region. Thus, $M(i, R)$ calculates the mean amount of ingredient i in the region's recipes (R) while $M(i, AR)$ calculate the mean amout of ingredient i in all the country's recipes (AR). We have the IA function as follows:

$$IA(i, R) = \frac{M(i, R)}{M(i, AR)}$$

Because the IA function calculates the mean of ingredient's amount, it is independent to the frequency of that ingredient. The higher IA value is, the higher possibility it is the region's featured ingredient. Because both numerator and denominator in the formula have the same unit, the IA value is non-unit. Therefore, regardless to the variety of the ingredient's unit, we have a stable metric for evaluating the ingredient's amount.

4.3 Ingredient Unique

The IF and IA functions above might tell us how often an ingredient appears in the region, but this ingredient can often appear in many regions. To be a featured ingredient of a region, the ingredient must satisfy the condition that it appears in the region but doesn't appear in many other regions. We propose the third function IU as follows:

$$IU(i, A) = \log\left(1 + \frac{|A|}{|\{i|i \in a, a \in A\}|}\right)$$

in which i is the ingredient in region a and A is the set of regions.

This function calculates the uniqueness of an ingredient among all the regions. The more often an ingredient appears in different regions the less unique it is. In other words, it is not the featured ingredient of the region. The higher IU value corresponds to higher possibility it is the region's featured ingredient. We use the log scale to make sure the IU values are not too big.

4.4 Featured Index

Featured Index, which is denoted by FI, is the index used to rank ingredients in a region in term of featured ingredient. We realize that these three functions are all proportional to

the rank of the featured ingredient, thus we proposed FI to be the production of these three function's values as follows.

$$FI(i, R) = IF(i, R) \times IA(i, R) \times IU(i, A)$$

The FI function returns the featured index of ingredient i in a region which has a set of recipes R. A is the set of all regions in the country. The ingredients which have the highest FI would be the featured ingredients. On the other hands, the ingredients which have the lowest FI would be considered as the common ingredients for every region.

4.5 Meta data Recalculation on Update

To evaluate the Feature Ingredient we need to calculate the meta data of recipe database, which are IF, IU, IA values.

In chapter 6 We propose a system that integrate a function allowing user to register their own recipes to the system. And every time a new recipe is added to the database, we have to recalculate the meta data. There is a fact that registering new recipe is a frequent action of users and each single action we still have to calculate many kind of functions. This might cause massive calculation for the system. But we aware of useful information in the old meta data can be used to calculate the new meta data. This means we don't need to gather all the data in database and recalculate new meta data. Instead, we just need to use the old meta data. In this subsection we will discuss about how to accomplish this task.

4.5.1 IF recalculation on update

The IF value of ingredient i in region R is calculated by this formula.

$$IF(i, R) = \frac{|\{i | i \in r, r \in R\}|}{|R|}$$

When we add one more recipe, we also add some more ingredients. These ingredients make the frequencies of them in the region change. We aware that there are two kind of ingredients will make different change in the IF value of ingredients in the regions: the ingredients in the new recipe and the ingredients that are not. In any case the denominator is added 1 because the number of recipe in the region has increased 1. But in the former case, the numerator doesn't change while in the later case it does. Thus, we have two following formulas for the former and later case respectively.

$$IF'(i, R) = IF(i, R) - \frac{IF(i, R) - 1}{|R| + 1}$$

and

$$IF'(i, R) = IF(i, R) - \frac{IF(i, R)}{|R| + 1}$$

Remember that the IF value is calculated for only the ingredients in the region not in other regions. The number of recalculated IF is the number ingredients in the region which has the new recipe.

4.5.2 IU recalculation on update

The IU value of ingredient i only changes when i is added for the first time in a region. In the case, the denominator in this formula is added 1. Thus we can have the following formula for the new IU value of ingredient i.

$$IU'(i, A) = log(1 + \frac{|A| \times (e^{IU(i,A)} - 1)}{|A| + e^{IU(i,A)} - 1})$$

4.5.3 IA recalculation on update

The IA formula is complex. When we have one new recipe, the average amount of ingredient in region and all over the country has been changed. This causes both the numerator and denominator in the main formula change. Thus, we have new formula as below:

$$IA'(i,R) = \frac{M'(i,R)}{M'(i,AR)}$$

Our task is recalculating both the average amount of ingredient i in region R and all over the country A. Because we use the same formula for both of these cases, we can use the same new formula. When a new recipe is added, the numerator in the below formula will be added by the new amount of the ingredient i in the new recipe and the denominator is added by 1.

$$M(i,S) = \frac{\sum^{i \in r, r \in S} amount(i,r)}{|\{i | i \in r, r \in S\}|}$$

Thus, from the original formula above, we have a formula for recalculation as follows:

$$M'(i,S) = M(i,S) - \frac{M(i,S) - amount(i,r)}{|\{i | i \in r, r \in S\}| + 1}$$

Because only the average amount of ingredients in the new recipe has been changed, thus we only recalculate for these ingredients.

4.5.4 FI recalculation on update

The FI value of ingredient i in region R is the production of the above values, thus we just need to keep the old formula.

$$FI(i,R) = IF(i,R) \times IA(i,R) \times IU(i,A)$$

Chapter 5

Prototype System and Its Implementation

This section describes the recipe databse and experimental studies on this database by applying the featured ingredient analysis algorithm.

5.0.5 The Recipe Database

In order to make advantages of recipes in food analysis, we collected recipes in many regions to build a recipe database.

We build a recipe database in which recipes are grouped by region. A script written in Python crawls all the recipes from a Japanese cooking website [13]. We chose this website because the recipes are typical foods grouped by regions. The website is only for Japanese recipes, thus we now only have the database for Japanese foods. Each food is characterized by its name, the region it belongs to and its recipe. Each recipe is stored as a map collection in which the ingredient is the key and the couple of amount and unit is the value. Each of the recipes we get from the website is created for various amounts of people. For example, there are recipes for 4 people but there are also recipes for 3 people. Thus we need to normalize the ingredients' amount in each recipe for one person.

There are about 200 recipes over 7 regions in Japan: Kanto, Hokkaido-Tohoku, Shikoku, Tyubu, Kyusyu-Okinawa, Kansai and Tyugoku. We calculate all the above functions for

Table 5.1: Ingredient Frequency of Ingredients in Kanto region vs Shikoku region

Kanto region		**Shikoku region**	
Ingredient	**IF**	**Ingredient**	**IF**
Soy Sauce (醤油)	1.00	Soy Sauce (醤油)	1.00
Miso (みそ)	0.9	Salt (塩)	1.00
Sugar (砂糖)	0.83	Rice (米)	0.83
Sake (酒)	0.83	Sake (酒)	0.67
Salt (塩)	0.67	Green onion (万能ねぎ)	0.50
...	...	...	...
Dried bonito (かつお節)	0.08	Kelp soup (ダシ昆布)	0.16
Pumpkin (かぼちゃ)	0.08	Deep-fried Tofu (油揚げ)	0.16
Kamaage Shirashi (釜揚げしらす)	0.08	Seared bonito (鰹の敲き)	0.16

every recipe in Japan, but we only show the experimental results of Kanto and Shikoku within this paper. We chose these two regions because they lie far apart in different islands of Japan. The experimental results are discussed below.

5.0.6 Ingredient Frequency

Table 5.1 shows that there are some common ingredients which often appear in both Kanto and Shikoku regions such as Soy Sauce (しょうゆ), Sake (酒), Salt (塩),... This is reasonable because we know that these ingredients are common in Japan. Because they often appear in other regions, the IF function is not enough to evaluate the region's featured ingredients. However, it helps us partially understand the habit in using materials in regions. For example, Green onion (万能ねぎ) often appears in Shikoku but not in Kanto region and Sugar (砂糖) often appears in Kanto but not in Shikoku region. This leads us to the idea that typical Kanto foods are often sweeter than Shikoku foods.

Table 5.2: Ingredient Amount of Ingredients in Kanto region vs Shikoku region

Kanto region		**Shikoku region**	
Ingredient	**IA**	**Ingredient**	**IA**
White radish (大根)	4.27	Shredded seaweed (刻みのり)	6.00
Tempura flour (天ぷら粉)	3.20	Carrot (にんじん)	3.95
Shredded seaweed (刻みのり)	3.00	Tempura flour (天ぷら粉)	3.20
...	...	...	...
Taro (里芋)	0.02	Sweet potato (さつまいも)	0.06
Cake flour (薄力粉)	0.02	Chicken thigh (鶏もも肉)	0.05
Field mustard (菜の花)	0.02	Sushi vinegar (すし酢)	0.05

5.0.7 Ingredient Amount

Table 5.2 shows the result of the IA value for Kanto and Shikoku region. We can see that most of the IA values are around 1, which means there is not much difference in the way of using an ingredients' amount between Shikoku region and other regions. However, there are some interesting results. For example, in Kansai region, the mean amount of pepper (こしょう) is 11 times greater than the mean amount of total peper in Japan. See details in Table 5.3.

5.0.8 Ingredient Uniqueness

Table 5.4 reflects the fact that the common ingredients such as Salt (塩), Sweet cooking wine (みりん), Ginger (しょうが), Soy sauce (しょうゆ) appear in almost every regions in Japan while the ingredients such as Peanut (落花生) and Chive (あさつき) are not too common and mostly appear in only one region. The ingredients which have the IU value of 0 appear in every region.

Table 5.3: Ingredient Amount of Ingredients in Tyubu region vs Kansai region

Tyubu region		**Kansai region**	
Ingredient	**IA**	**Ingredient**	**IA**
Pork loin (豚ロース肉)	25.50	Pepper (こしょう)	11.00
Seaweed (刻みのり)	6.00	Sweet cooking wine (みりん)	6.88
Green onion (長ねぎ)	3.72	Soy sauce (醤油)	5.49
Onion (玉ねぎ)	3.60	Green onion (長ねぎ)	3.72
...	...	...	...
Taro (里芋)	0.03	Milk (牛乳)	0.04
Cake flour (薄力粉)	0.02	Minced chicken (鶏ひき肉)	0.03
Pepper (こしょう)	0.01	Soup (だし汁)	0.01

Table 5.4: Ingredient Uniqueness of Ingredients in Japan

Ingredient	**IU**
Peanut (落花生)	2.80
Chive (あさつき)	2.80
...	...
Salt (塩)	0.00
Sweet cooking wine (みりん)	0.00
Ginger (しょうが)	0.00
Soy sauce (しょうゆ)	0.00

Table 5.5: Featured Index of Ingredients in Kanto region and Shikoku region

Kanto region		**Shikoku region**	
Ingredient	**FI**	**Ingredient**	**FI**
Natto (納豆)	0.60	Kelp (昆布)	1.40
Dried radish (切干大根)	0.47	Sea bream (鯛の切り身)	0.94
Saury (さんま)	0.47	Ponzu suace (ポン酢)	0.90
...	...	...	...
Vineger (酢)	0.00	Sweet cooking wine (みりん)	0.00
Shredded seaweed (刻みのり)	0.00	Egg (卵)	0.00
Wine (酒)	0.00	Wine (酒)	0.00
Ginger (しょうが)	0.00	Rice (米)	0.00

5.0.9 Featured Index

The Featured Index (FI) is the main metric we use to elvaluate the regions' featured ingredients. Table 5.5, which is the experimental result of FI calculation for Kanto vs Shikoku region, shows us some interesting information. For example, Natto (納豆) is the ingredient which has the highest FI value in Kanto region. This means Natto (納豆) is possibly the featured ingredient of Kanto region. In Shikoku region, Ponzu sauce (ポン酢) is also often used for Shikoku's foods. The FI of the same ingredient for different regions might differentiate but we figure that if an ingredient ranks high in one region, it cannot rank high in any other regions. The same thing is true for the low-rank ingredients.

Chapter 6

Web-based Application

Using the algorithm we propose a system that will help cooking people transform the typical region's food from the original recipe to a new one that has a typical taste of specified region. For convenience and wider use, we develop this system as a web-based system. The system's outline and the model are described below.

6.1 The System's Outline

The system has two main functions:

- Suggesting possible recipes from the set of available materials inputted by the user.

  When people cook, they might already have many materials available in their house such as pepper, chili, chicken, etc. But they have no idea which food is the best choice to cook. Thus, we provide a system which has an extra function that accepts available materials inputted by users and then searches in the recipe database for recipes that are suitable for the inputted materials. "Suitable" means the number of extra-buy materials are the least. The suitable recipes will be shown in order; the smaller the number of extra-buy materials there are, the higher rank that recipe will be.

- Transforming a recipe so that it has a specific region's taste.

This is the most important function of the system. It uses the algorithm to extract the featured materials of the specified region and then transforms the original recipe to the new one.

Based on these two functions we divide the system into three modules. These three modules are shown in the middle of Fig. 6.1, represented by three rectangle boxes. The model of the system is described in the next subsection.

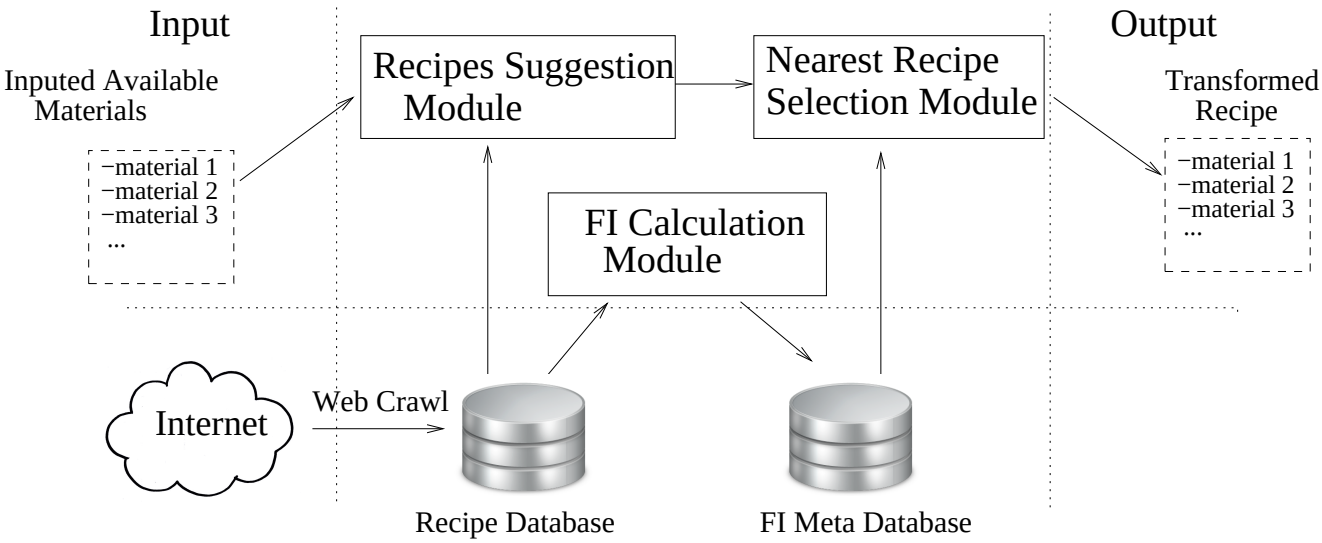

Figure 6.1: The System's Model with Recipe Suggestion Module, Featured Index Calculation Module and Nearest Recipe Selection Module.

6.2 The System's Model

6.2.1 The Recipe Suggestion Module Based On Available Materials

This module responds to the first function of the system, suggesting the possible recipes based on available materials inputted by the user. The input of this module is a set of available materials that the user has. It accesses the recipes database during the calculation and its output will be the list of the recipes which include most of the inputted materials.

This output is passed to the Featured Index Calculation Module as shown in Fig. 6.1. After inputting available materials, the system will search in the recipe database for the most suitable recipes and show them in rank order. The pseudo code is shown as below.

for $recipe \in recipes$ **do**

$recipe.lack \leftarrow |recipe| - |recipe \cup inputed\ materials|$

end for

sort the recipes by recipe.lack

return *recipes*

6.2.2 The Featured Index Calculation Module

The user selects one of the recipes recommended by the Recipe Suggestion Module. Then selects the region which they want to transform the recipe in order to have that region's taste. This module applies the region's Featured Materials Extracting Algorithm and outputs the list of Featured Index for all materials in the region then stores them in the FI Meta Database as shown in Fig. 6.1. Because we are not using all of the lists to extract the featured materials, we only look at two kinds of the following materials:

- The top rank FI materials.

  These materials are the materials which are often used in the desired region, but not in other regions.

- The bottom rank FI materials.

  These materials are the most common materials which are used in almost all regions, but with different amounts.

We use both kinds and combine them with the materials appearing in the original recipe. The result is the list of materials and their amount for the food. The output should look in

the shape as follows:

- Onion 2 (original)
- Lemon 1/2 (original)
- ...
- Natto 100g (top FI, newly added, region's average)
- Sugar 100g (bottom FI, newly added, region's average)

Among the bottom rank FI materials, we only take the materials which are already in original recipe to apply into a new recipe. Among the newly added materials we use the average amount of them in the region. The output of this module is passed to the Nearest Recipe Selection Module.

6.2.3 The Nearest Recipe Selection Module

The output of Featured Index Calculation Module gives us the list of materials and their amount which is suitable for the region's taste. But it doesn't mean that we could use that list to make food. If we immediately apply the list of ingredients with the associated amount, we may have a wrong solution. This is because the newly added ingredients and their associated amounts are just the mean value of ingredients in the region. In result, there is the possibility of a bad tasting food. Thus, we propose to search in the region the nearest recipe in term of ingredients and amount. Then apply the suitable ingredients and its amount in that recipe to our food.

Consider the list of materials as a vector. We calculate the similarity between the region's recipe and the average output above. Because we currently have ingredients and their amounts, there is the problem that the unit of ingredient's amounts are different and we

cannot calculate the similarity. Thus we need to normalize these units. The alternative, we propose, is taking the fraction between the recipe's amount and the average amount all over the country. This gives us the values that are unit-independent, therefore usable for the similarity calculation. There are various methods to calculate the similarity between two vectors [14–16]. Among of these methods, Cosine similarity and Euclidean distance are the most famous methods. In this paper, we use the Euclidean distance, therefore the minimum value is adapted. The details of the algorithm is shown below. $X(x_1, x_2, \ldots, x_m)$ with $m \in N$ is the list outputted by the FI Calculation Module and $Y(y_1, y_2, \ldots, y_n)$ with $n \in N$ represents a list in the lists of the specified region's recipes

for $ingredient \in recipe\ X$ **do**

$x_i \leftarrow \frac{amount}{average\ amount\ in\ the\ country}$

end for

$min \leftarrow \infty$

for $recipe \in region's\ recipes$ **do**

for $ingredient \in recipe\ Y$ **do**

$y_i \leftarrow \frac{amount}{average\ amount\ in\ the\ country}$

end for

$similarity \leftarrow \sqrt{\sum_{i=k}^{l} (x_i - y_i)^2}$

if $similarity < min$ **then**

$min \leftarrow similarity$

end if

end for

return $recipes$

Note that though X and Y don't have to have the same dimensions but we only select the

ingredients i which appears both in X and Y to calculate the similarity. x_i and y_i in which $i \in [k, l]$, are the amounts of ingredient i in X and Y respectively.

6.3 Database Design

6.3.1 Entity Relation Diagram

Basically, we have the following entities:

- Entity of Ingredient mainly has attributes of ingredient such as: ingredient's name, ingredient's unit, ingredient's
- Entity of Recipe has recipe's name, introduction, instruction, image attributes.
- Entity of User has user's name, email, password attributes.
- Entity of Region has attributes of region's name, description about the region.

and the following relations between these entities:

- The relation named "belong to" between entity of recipe and region. Many recipes might belong to one region. This is a many-one relation.
- The relation named "content" between entity of recipe and ingredient. One recipe has many ingredient and one ingredient could appear in many recipe. This is a many-many relation.
- The relation named "in" between entity of user and entity region. The relation indicate which region the user is living in and it make a base to recognize the region of the newly registered by user. This is a many-one relation.
- The relation named "available" between entity user and entity ingredient. As we describe in the previous sections, our system will have a function that allow user

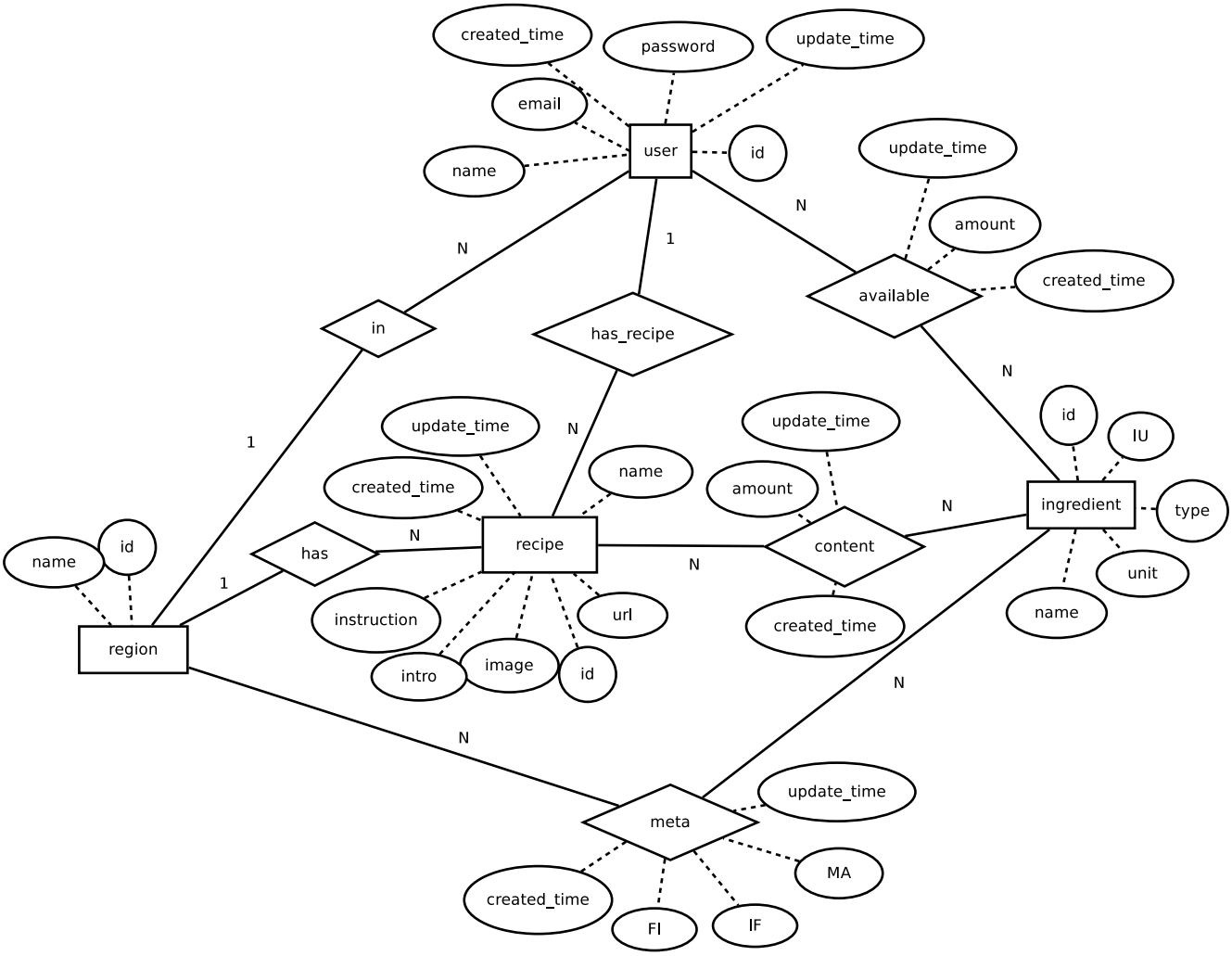

Figure 6.2: Entity Relation Diagram

to register their available ingredients in their home and the system will recommend suitable recipes. A user might have many ingredients and one ingredient might appear in many user's home. Thus the relation here is many-many relation.

- The relation named "meta" between entity of region and ingredient. This is the most important relation in our system because it reflects the algorithm. The meta data of IF, FI, IA are included in this relation. Because these data are different from region to region and ingredient to ingredient.

The entities relations diagram of the system is shown in Fig. 6.2

6.3.2 Database Scheme

After we have a entity relation diagram we turn it into scheme in relational database. For example, we use SQL in this research. Each entity we create a table with columns reflecting attributes of the entity. For example, we create a table user for the entity of user with columns username, password, email which are the attributes of the entity. For many-one relation such as the relation between user and region we don't need to create a table. Instead, we add the primary key of region to user table as a foreign key. For a many-many relation we create a new table that include both primary keys of two entities which are having the relation.

Fig. 6.3 shows the scheme for the system's database based on proposed entities relation diagram.

6.3.3 Database for real Data

Fig. 6.4 shows the real data for ingredients in Japan's recipes. For each ingredient we have an IU value of it. The value reflect uniqueness of the ingredient among regions in Japan. The amount is not included in the table because the amount of ingredient depends on in which recipe it appears.

The TF, IA, MA and IF value of ingredient are shown in Fig. 6.4. We only use the FI value for evaluating the featured ingredient but we also store the TF, IA, MA value because they are necessary to recalculate the new meta data as we discussed in chapter 6, the Meta data recalculation section.

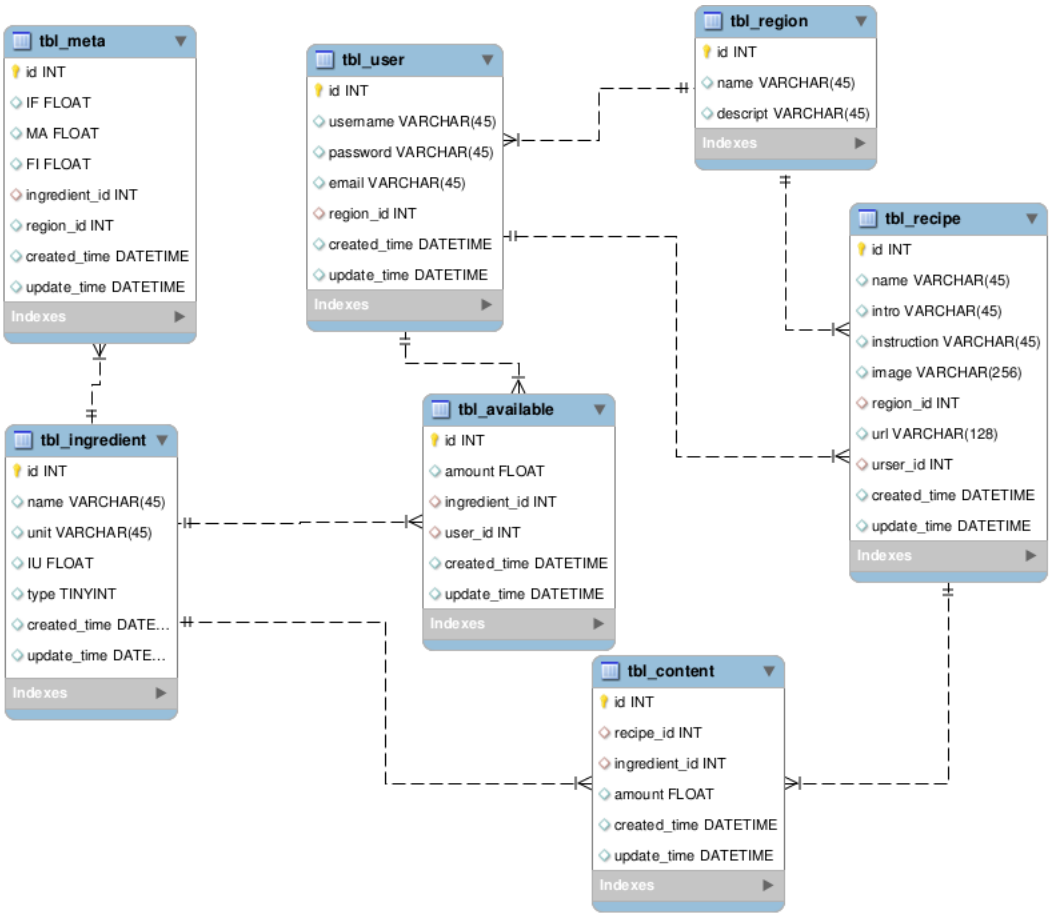

Figure 6.3: Scheme for the System's Database

←T→	id	name	unit	IU	type
Edit Inline Edit Copy Delete	121	油揚げ	枚	0.69	0
Edit Inline Edit Copy Delete	122	切り昆布	パック	2.2	0
Edit Inline Edit Copy Delete	123	いのしし肉	g	2.2	0
Edit Inline Edit Copy Delete	124	そば(生麺)	玉	2.2	0
Edit Inline Edit Copy Delete	125	鶏むね肉	g	1.61	0
Edit Inline Edit Copy Delete	126	鶏がらスープの素	大さじ	2.2	0
Edit Inline Edit Copy Delete	127	干ししいたけ	枚	1.1	0
Edit Inline Edit Copy Delete	128	奈良漬	g	2.2	0

Figure 6.4: Ingredient Table with real data. It includes 404 ingredients.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

In this paper, we have presented the regional foods' features extracting algorithm and the experimental results. The experimetal results partially reflect the featured ingredients in regions.

To show the feasibility and applicability of our algorithm, we build the cooking support system that helps cooking people transform the original recipes to have a featured taste of another regions.

In this paper, the recipes from 8 regions of Japan are used. However, the proposed algorithm is scalable to adapt to any recipe from any region in the world. As a future work, we intent to develop a multilingual translation recipe with the function that can transform recipes between countries.

7.2 Future Works

In fact, the seasoning and non-seasoning ingredients affect the food's taste in different ways. Thus, we also develop the research in that direction, analyze food's features by applying different methods for seasiong ingredients and non-seasoning ingredients.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof.Yasushi Kiyoki for the continuous support of my bachelor study and research, for his patience, motivation, enthusiasm, and immense knowledge since I joined Multimedia Database Laboratory. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my bachelor.

I would like to thank my advisor, Mrs.Nguyen Thi Ngoc Diep who has always taken care of me since I joined Multimedia Databse Laboratory. She gave me a lot of advice and support when I am not sure about my research motivations.

I would like to thank Dr.Sashiori Sasaki and Mr.Jeremy as well for their encouragement, insightful comments, and hard questions...

I thank to the members of Multimedia Database Laboratory for their great friendship and supports.

I would like to thank to Mr.Burgin and Mr.Loyld for checking my English writings.

Last but not the least, I would like to thank my family: my parents Nguyen Trung Dung and Bui Thi Thuy Lan, for giving birth to me at the first place and supporting me spiritually throughout my life. I am also grateful to my friends in Japan who have helped me a lot to start a new life in Japan.

January 15, 2014

Trung Duc Nguyen

Publications

Long Talk

Trung Duc Nguyen, Diep Thi-Ngoc Nguyen, Yasushi Kiyoki. A Regional Food's Features Extraction Algorithm and Its Application. In proceeding of Workshop on Cooking and Eating Activities in conjunction with ACM Conference on Multimedia. Oct 21, Barcelona, Spain. [17]

Poster

SFC OPEN RESEARCH FORUM 2014.

Appendix:

Bibliography

[1] S. Morioka and H. Ueda, "Cooking support system utilizing built-in cameras and projectors," in *MVA2011 IAPR Conference on Machine Vision Applications*, pp. 84–89, TeX Users Group, June 13-15 2011.

[2] Y. Nakauchi, T. Suzuki, A. Tokumasu, and S. Murakami, "Cooking procedure recognition and support system by intelligent environments," in *RIISS '09*, TeX Users Group, 2009.

[3] G. Villalobos, R. Almaghrabi, P. Pouladzadeh, and S. Shirmohammadi, "An image procesing approach for calorie intake measurement," 2012.

[4] I. Ide, Y. Shidochi, Y. Nakamura, D. Deguchi, T. Takahashi, and H. Murase, "Multimedia supplementation to a cooking recipe text for facilitating its understanding to inexperienced users," in *IEEE International Symposium on Multimedia*, TeX Users Group, 2010.

[5] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, pp. 11–21, 1972.

[6] M. Svensson, K. Höök, and R. Cöster, "Designing and evaluating kalas: A social navigation system for food recipes," *ACM Trans. Comput.-Hum. Interact.*, vol. 12, pp. 374–400, Sept. 2005.

[7] H. Xie, L. Yu, and Q. Li, "A hybrid semantic item model for recipe search by example," in *Multimedia (ISM), 2010 IEEE International Symposium on*, pp. 254–259, 2010.

[8] Y. Mino and I. Kobayashi, "Recipe recommendation for a diet considering a user's schedule and the balance of nourishment," in *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, vol. 3, pp. 383–387, 2009.

[9] S. Karikome and A. Fujii, "A system for supporting dietary habits: Planning menus and visualizing nutritional intake balance," in *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*, ICUIMC '10, (New York, NY, USA), pp. 56:1–56:6, ACM, 2010.

[10] Y. Shidochi, T. Takahashi, I. Ide, and H. Murase, "Finding replaceable materials in cooking recipe texts considering characteristic cooking actions," in *Proceedings of the ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating Activities*, CEA '09, (New York, NY, USA), pp. 9–14, ACM, 2009.

[11] Y. van Pinxteren, G. Geleijnse, and P. Kamsteeg, "Deriving a recipe similarity measure for recommending healthful meals," in *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, (New York, NY, USA), pp. 105–114, ACM, 2011.

[12] J. Wagner, G. Geleijnse, and A. van Halteren, "Guidance and support for healthy food preparation in an augmented kitchen," in *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*, CaRR '11, (New York, NY, USA), pp. 47–50, ACM, 2011.

[13] Wiki, "Typical foods by area." `http://www.s-recipe.com/`.

[14] Wiki, "Cosine similarity." http://en.wikipedia.org/wiki/Cosine_similarity.

[15] Wiki, "Euclidean distance." http://en.wikipedia.org/wiki/Euclidean_distance.

[16] G. Qian, S. Sural, Y. Gu, and S. Pramanik, "Similarity between euclidean and cosine angle distance for nearest neighbor queries," in *Proceedings of the 2004 ACM symposium on Applied computing*, SAC '04, (New York, NY, USA), pp. 1232–1237, ACM, 2004.

[17] T. D. Nguyen, D. T.-N. Nguyen, and Y. Kiyoki, "A regional food's features extraction algorithm and its application," in *Proceedings of the 5th International Workshop on Multimedia for Cooking & Eating Activities*, CEA '13, (New York, NY, USA), pp. 15–20, ACM, 2013.