TYPICALITY ANALYSIS OF THE COMBINATION OF INGREDIENTS IN A COOKING RECIPE FOR ASSISTING THE ARRANGEMENT OF INGREDIENTS

Satoshi Yokoi[1], *Keisuke Doman*[2,1], *Takatsugu Hirayama*[1], *Ichiro Ide*[1],
Daisuke Deguchi[3,1], *Hiroshi Murase*[1]

1 Graduate School of Information Science, Nagoya University, Japan
yokois@murase.m.is.nagoya-u.ac.jp, {hirayama, ide, murase}@is.nagoya-u.ac.jp
2 School of Engineering, Chukyo University, Japan
kdoman@sist.chukyo-u.ac.jp
3 Information & Communications, Nagoya University, Japan
ddeguchi@nagoya-u.jp

ABSTRACT

As the number of cooking recipes posted on the Web increases, it becomes difficult to find a cooking recipe that a user needs. Moreover, even if it can be done, it is still difficult for users to arrange the cooking recipe, for example, by replacing ingredients with different ones. To deal with such problems, we propose a framework for typicality analysis of the combination of ingredients. The framework calculates a typicality value for each combination of ingredients. The list of ingredients can be arranged by adjusting the typicality value by adding or removing ingredients iteratively. The effectiveness of the proposed framework was confirmed through subjective experiments.

***Index Terms*—** Cooking recipe, typicality analysis, arrangement of ingredients

1. INTRODUCTION

In recent years, the number of cooking recipes posted on the Web is increasing. For example, there are more than two million cooking recipes on the COOKPAD Web site[1] and close to one million cooking recipes on the Rakuten Recipe Web site[2], as of April 2015. These recipe portal Web sites provide us with various choices of cooking recipes. There are, however, the following problems.

Problem 1) Too many candidates of cooking recipes exist for each dish category: Difficult for users to decide the one they should refer to cook.

Problem 2) Usually no information about the supplementation of ingredients: Difficult for users to know if ingredients they have can be used in place of the ingredients listed in the cooking recipe.

As for Problem 1, a standard approach to find cooking recipes is to search by keywords such as the title or the category of a dish, or names of ingredients. However, it is not always easy to find a cooking recipe that users really need. As for Problem 2, even if users manage to find a cooking recipe that matches their needs, the ingredients may not be accepted due to their preference or demand.

There are related work for the solution to such problems. Ueda et al. have proposed a method for cooking recipe recommendation based on user's preference [1]. The method quantizes the characteristics of ingredients from the user's history and preference for cooking. However, the method considers neither the dishes the users wish to eat nor the ingredients they wish to use, although it recommends them a cooking recipe based on their general preferences. Shidochi et al. have proposed a method for ingredient replacement based on the analysis of a large amount of cooking recipes [2]. However, the method does not consider the appropriateness of the combination of ingredients after the replacement. Tsukuda et al. have proposed a method for cooking recipe recommendation by adding and removing ingredients based on the co-occurrence probability of ingredient pairs [3]. The method regards that the higher the co-occurrence of an ingredient pair is, the more typical its combination is. Accordingly, a cooking recipe can be made more typical or more atypical by replacing ingredients according to the co-occurrence. However, although more than two ingredients are usually used in a cooking recipe, the method considers the co-occurrences between only two ingredients. Therefore, when considering the arrangement of ingredients, it should be better to consider the combination of all the ingredients simultaneously.

The work presented in this paper aims at the arrangement of ingredients considering users' intention, and proposes a

[1]COOKPAD Inc., "COOKPAD," `http://cookpad.com/`.
[2]Rakuten, Inc., "RAKUTEN RECIPE," `http://recipe.rakuten.co.jp/`.

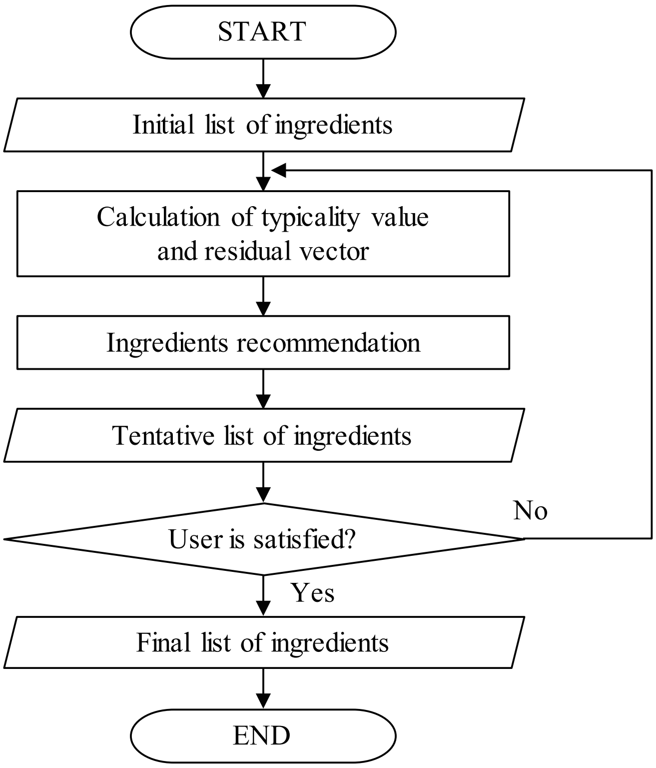


Fig. 1. Process flow of assisting the arrangement of ingredients based on typicality analysis.

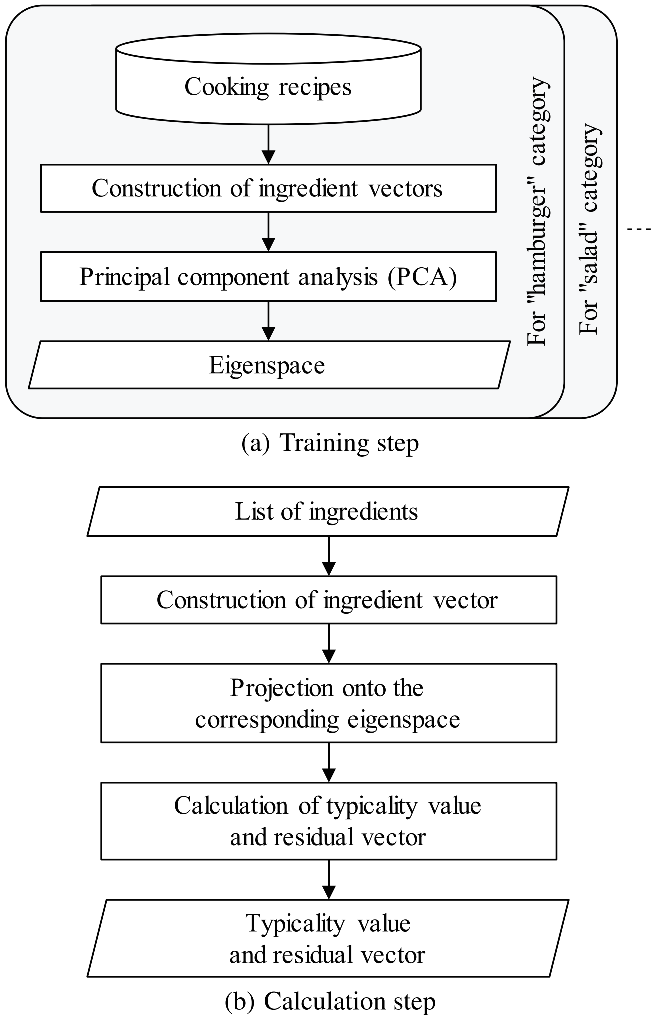


Fig. 2. Process flow of typicality analysis.

framework for typicality analysis on the combination of all the ingredients in a cooking recipe. The basic idea for typicality analysis is based on principal component analysis (PCA). Once an eigenspace is constructed based on the occurrences of combinations of ingredients in cooking recipes for each dish category, a typicality value of a combination of ingredients is calculated by the residual of its projection onto the eigenspace. The proposed method makes the combination of ingredients more typical or more atypical by controlling the residual vector by adding and removing ingredients.

This paper is organized as follows. The next section describes the proposed framework for typicality analysis of ingredients in cooking recipes toward the arrangement of ingredients. Section 3 reports and discusses experimental results for investigating the effectiveness of the proposed method. Section 4 concludes the paper.

2. PROPOSED METHOD: ARRANGEMENT OF INGREDIENTS BASED ON TYPICALITY ANALYSIS

The process flow of the arrangement of ingredients in the proposed framework is shown in Fig. 1. The framework recommends an user to update the list of ingredients considering the typicality of the combination of ingredients. The user repeats to add or remove ingredients until he/she is satisfied with the resultant list of ingredients. Note that we assume that a dataset of a sufficiently large number of cooking recipes is available, for example, via Rakuten Data Release[3] as we have done.

The details of typicality analysis and arrangement of ingredients are described below, respectively.

2.1. Typicality analysis

The proposed method calculates the typicality value of the combination of ingredients in cooking recipes as shown in Fig. 2. An eigenspace is constructed in advance from combinations of ingredients in cooking recipes for each dish category, as shown in Fig. 2(a). Then, the typicality value and the residual vector for the combination of ingredients in a given cooking recipe are calculated within the eigenspace, as shown in Fig. 2(b). The details of each step are described below.

[3]Rakuten Inc., "Rakuten Data," `http://rit.rakuten.co.jp/opendata.html`.

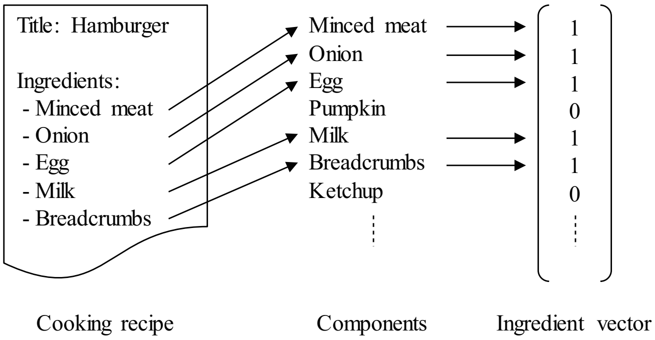


Fig. 3. Vectorization of the list of ingredients

2.1.1. *Construction of an eigenspace for each dish category*

As pre-processing, cooking recipes are categorized into dish categories since ingredients widely differ among different dish categories. For example, "Chinese-style hamburger" and "Soft Tofu hamburger –Japanese style–" should be categorized into the "hamburger" category. This can be realized by applying morphological analysis to the title of a cooking reicpe and then extracting a noun phrase according to certain grammatical rules.

Next, an eigenspace is constructed for each dish category. First, the list of ingredients is represented as a vector as shown in Fig. 3. The vector is composed of binary values, that is, each component indicates if an ingredient appears in the list of ingredients in a cooking recipe or not. Hereafter, we call such a vector an "ingredient vector". The number of dimensions of an ingredient vector is the number of ingredients used at least once in all the cooking recipes in the dish category. Next, principal component analysis (PCA) with the criteria of minimizing the reconstruction error is applied to the ingredient vectors. Here, in the first eigenvector, components corresponding to representative ingredients in the dish category would have high values. Also, in the second or later eigenvectors, each component takes values according to the frequency of ingredients in the dish category. We expect that these eigenvectors can construct an eigenspace that represents the typicality or atypicality of the combination of ingredients in a dish category.

2.1.2. *Calculation of the typicality value of the combination of ingredients*

The typicality value for the combination of ingredients is calculated as the L2 norm of the projection onto the eigenspace. For more details, let $\mathbf{x}$ be an ingredient vector, and let $\tilde{\mathbf{x}}$ be the normalized vector of $\mathbf{x}$ whose L2-norm is one. Then, the typicality value T of the combination of ingredients is defined as

$$T = \|\tilde{\mathbf{x}}^*\|_2, \tag{1}$$

where $\tilde{\mathbf{x}}^*$ is the projection of $\tilde{\mathbf{x}}$ onto the eigenspace. Here, the use of $\tilde{\mathbf{x}}$ instead of $\mathbf{x}$ is to normalize the typicality value T into the range of $[1, 0]$. Thus, T indicates the ratio of ingredients represented by an ingredient vector on the eigenspace. The typicality value T should become higher as the combination of ingredients becomes more typical, and vice-versa, since the eigenspace is a subspace that is well-designed to represent the typicality of the combination of ingredients for a dish category.

2.1.3. *Calculation of the residual vector*

As a means for controlling the typicality value T, we focus on the residual vector $\mathbf{x}^\sharp$ defined as

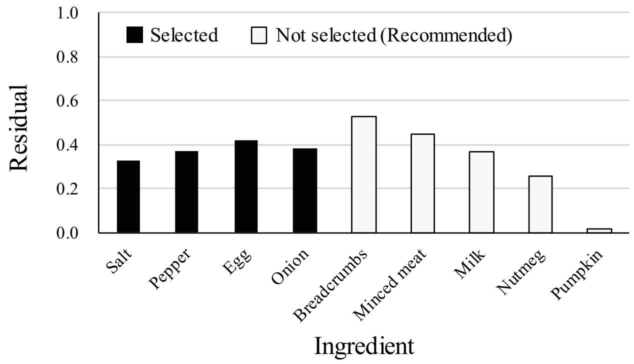
$$\mathbf{x}^\sharp = \mathbf{x} - \mathbf{x}^*, \tag{2}$$

where $\mathbf{x}^*$ is the projection of $\mathbf{x}$ onto the eigenspace. Note that each component of $\mathbf{x}^\sharp$ can be either positive or negative. The residual vector $\mathbf{x}^\sharp$ is the difference between $\mathbf{x}$ and $\mathbf{x}^*$, and a criteria for evaluating the accuracy of an ingredient vector's representation in the eigenspace.

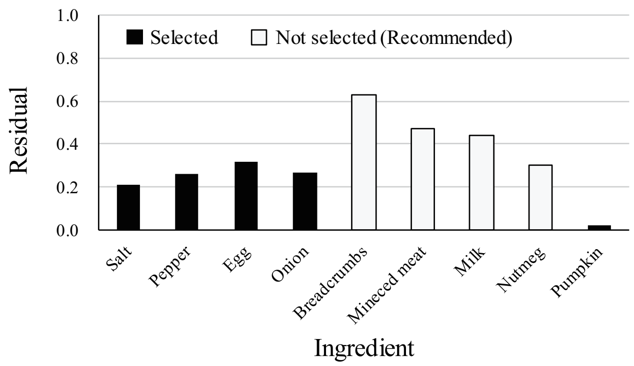
2.2. Arrangement of ingredients

The components with a high absolute value in $\mathbf{x}^\sharp$ contributes to a large residual in an eigenspace. A high absolute component value in a residual vector indicates that the corresponding ingredient is not typically used in the dish category. Thus, in order to adjust the typicality value T, we should first find the components with low or high absolute values, and then should add or remove the corresponding ingredients. This is the basic idea of the arrangement of ingredients in the proposed framework.

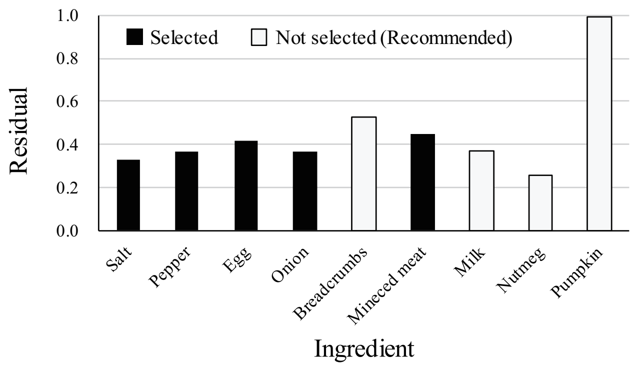
An example of updating the list of ingredients within one-dimensional eigenspace for the "hamburger" category is shown in Fig. 4. Figure 4(b) is the residual vector obtained by adding a "pumpkin" into the list of ingredients in Fig. 4(a), and conversely, Fig. 4(a) is the residual vector obtained by removing a "pumpkin" from the list of ingredients in Fig. 4(b). Here, the list of ingredients in Fig. 4(a) does not contain the "pumpkin" and its absolute residual value is low, which indicates that the additional use of a "pumpkin" is atypical for the "hamburger" category. As a result, the typicality value T decreases by adding the "pumpkin" because of its atypicality in the "hamburger" category. Similarly, Fig. 4(c) is the residual vector obtained by adding "minced meat" into the list of ingredients in Fig. 4(a), and conversely, Fig. 4(a) is the residual vector obtained by removing "minced meat" from the list of ingredients in Fig. 4(c). Here, the list of ingredients in Fig. 4(a) does not contain the "minced meat" and its absolute residual value is high, which indicates that the additional use of the "minced meat" is typical for the "hamburger" category. As a result, the typicality value T increases by adding the



(a) Initial state (Typicality value T: 0.64)



(b) After adding a pumpkin to (a) (Typicality value T: 0.52)



(c) After adding minced meat to (a) (Typicality value T: 0.69)

Fig. 4. Example of updating the list of ingredients for a hamburger recipe.

"minced meat" because of its typicality in the "hamburger" category.

As explained above, users can adjust the typicality T iteratively by adding or removing ingredients based on the component values of the residual vector. By this way, the user can arrange the ingredients on demand.

Table 1. Number of cooking recipes and ingredients for "hamburger" and "salad" categories in the experiment.

Dish category	# of cooking recipes	# of ingredients
Hamburger	780	768
Salad	6,409	3,343

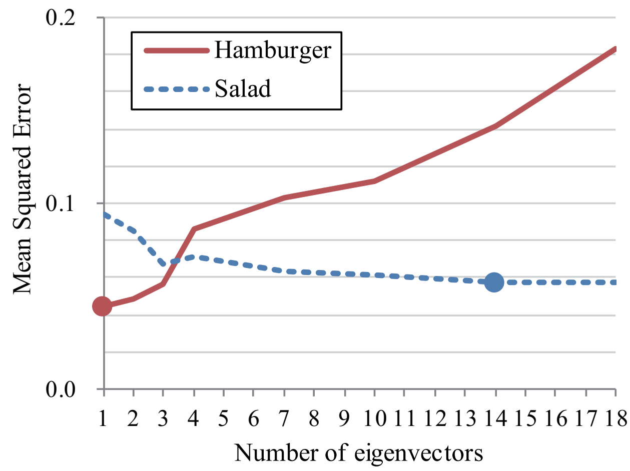


Fig. 5. Relationship between the mean squared error and the number of eigenvectors (The "●" indicates the point where the minimum mean squared error was obtained for each dish category).

3. EXPERIMENTS

We investigated the effectiveness of the proposed framework through subjective experiments. The following sections describe experimental data, a preliminary experiment for parameter optimization, and an evaluation experiment for the arrangement of ingredients.

3.1. Experimental data

We extracted cooking recipes for "hamburger" and "salad" categories from 440,000 cooking recipes posted on the "Rakuten Recipe" Web site[3] according to their titles. We chose these two dish categories because the variety of ingredients in the "hamburger" category is relatively small, but not so in the "salad" category, considering that the proposed framework is based on the analysis of the combination of ingredients. The number of cooking recipes and ingredients for each dish group is shown in Table 1.

3.2. Preliminary experiment: Parameter optimization for constructing the eigenspace for each dish category

We optimized the parameter for constructing the eigenspaces for each dish category so that the typicality values match our common sense.

3.2.1. Method

We chose ten cooking recipes with the following procedure in each dish category. First, we constructed an one-dimensional eigenspace for each dish category with the first eigenvector obtained from the experimental data in Table 1. Next, typicality values for combinations of ingredients in each cooking recipe were calculated. Then, we extracted ten cooking recipes at even intervals on the typicality T. Finally, we set a target typicality value for each cooking recipe based on Thurstone's paired comparison method [4]. Here, the subjects who participated in the comparison were two males and three females in their 20s to 40s who cooked regularly.

We evaluated the error between the target typicality values and typicality values calculated by the proposed framework while increasing the dimension of the eigenspace for each dish category. Here, the dimension is increased by adding eigenvectors to the eigenspace in descending order of their eigenvalues. Then, we searched the number of dimensions that minimized the mean squared error on the typicality calculation. Note that the lower the error is, the higher the accuracy of typicality calculation is.

3.2.2. Results

The results of parameter optimization are shown in Fig. 5. The highest accuracy for the "hamburger" category was obtained by using only the first eigenvector, whereas that for the "salad" category was obtained by using the first 14th eigenvectors. Here, the cumulative contribution ratios were 0.38 and 0.41 for the "hamburger" and the "salad" categories, respectively. As reference, the result of typicality calculation by the proposed framework is shown in Table 2.

3.2.3. Discussion

The combination of ingredients for the "hamburger" category could be represented within the low-dimensional eigenspace, but not so for the "salad" category. This result is reasonable because the variety of ingredients in the "hamburger" category is small, but not so in the "salad" category, as mentioned earlier. We used the optimal dimension obtained in this preliminary expeirment to construct an eigenspace for each dish category in the experiments described below.

3.3. Evaluation experiment for the arrangement of ingredients

We evaluated the effectiveness of the arrangement of ingredients by the proposed framework as described below.

3.3.1. Method

The number of subjects in the experiments were two males and three females in their 20s to 40s who cooked regularly.

Table 3. Experimental results: Number of subjects satisfied with the arrangement assistance (Total number of subjects were five).

Dish category	Goal	Satisfied	
		Proposed	Comparative [3]
Hamburger	Typical	5	4
	Atypical	4	4
Salad	Typical	4	4
	Atypical	4	3

For each dish category, each subject tried to make the list of ingredients with two different goals; one to make it typical, and the other to make it atypical. The subjects repeatedly added or removed ingredients to update the combination of ingredients toward the goal assisted by the proposed framework. Note that we pre-defined four different initial states for the combination of ingredients considering the occurrence frequency of the ingredients in the experimental data. Finally, they responded if they were satisfied with the assistance by the proposed framework. We counted the number of "satisfied" responses. For reference, the results were compared with those by Tsukuda et al.'s method [3].

3.3.2. Results

Experimental results are shown in Table 3. Almost all the subjects responded as "satisfied". Also, the proposed framework outperformed the comparative method [3], although we need more detailed investigation about the significance of its difference. These results showed that the proposed method was effective for assisting the arrangement of ingredients.

3.3.3. Discussion

Not all the subjects responded as "satisfied". One of the subjects commented that he/she could not make the combination of atypical ingredients even with the arrangment assistance during his/her update process. Thus, we should improve the method for calculating the atypicality value according to our common sense.

There are some problems yet to be solved. The proposed framework did not distinguish seasonings from ingredients. Seasonings play a different roll from so-called "ingredients", and should be treated separately. Also, the experimental data included irregular ingredients such as "pre-cooked hamburgers" and "leftovers", which may cause miscalculation of the typicality value. To deal with such problems, we should exclude ingredients whose occurrence frequency are extremely low. In addition, we expect to improve the performance of the proposed framework by integrating synonyms based on an ontology.

Table 2. Typicality values calculated by the proposed framework.
(a) Hamburger

Typicality	List of ingredients
0.85	Salt, onion, pepper, egg, breadcrumbs, minced meat, milk
0.76	Salt, onion, pepper, egg, breadcrumbs, minced meat, milk, nutmeg, bean sprouts
0.67	Salt, onion, pepper, egg, breadcrumbs, minced meat, Worcester sauce, Tofu
0.59	Salt, onion, pepper, egg, breadcrumbs, minced meat, ketchup, soy sauce, water, Worcester sauce, sugar, lotus root
0.55	Salt, onion, egg, breadcrumbs, minced meat, milk, ketchup, soy sauce, Worcester sauce, butter, Shiitake mushroom
0.47	Salt, onion, pepper, breadcrumbs, minced meat, ketchup, sauce, soy pulp, yam
0.39	Salt, onion, pepper, breadcrumbs, nutmeg, carrot, minced pork, ginger, green pepper
0.27	Salt, onion, pepper, minced chicken meat, flour, pumpkin
0.16	Salt, pepper, ketchup, potato starch, oil, soy pulp, potato
0.04	Onion, minced chicken meat, Tofu, flour, lotus root, Shiitake mushroom, Welsh onions

(b) Salad

Typicality	List of ingredients
0.61	Salt, mayonnaise, pepper, cucumber, pumpkin
0.54	Salt, mayonnaise, pepper, cucumber, radish, canned tuna
0.48	Salt, mayonnaise, pepper, cucumber, soy sauce, cabbage, crab stick, corn
0.42	Salt, mayonnaise, pepper, ham, potato, spinach
0.36	Salt, mayonnaise, pepper, soy sauce, vinegar, Shimeji mushroom, Enoki mushroom, taro, Sakura shrimp
0.30	Salt, pepper, soy sauce, olive oil, lemon juice, avocado, salmon
0.24	Salt, mayonnaise, pumpkin, broccoli sprouts, milk, Tofu
0.18	Salt, pepper, carrot, sesame dressing, Hijiki seaweed, chicken breast, green soybeans, black vinegar
0.12	Salt, olive oil, mini tomato, black pepper, bacon, garland chrysanthemum
0.06	Mayonnaise, radish, potherb mustard, dried bonito, noodle soup, Chikuwa

4. CONCLUSION

This paper proposed a framework for typicality analysis of the combination of ingredients for assisting the arrangment of ingredients. The proposed framework calculated a typicality value for a given combination of ingredients. The typicality value can be adjusted by adding or removing ingredients, which assists a user to arrange the list of ingredients considering his/her intention (i.e. typical or atypical). The effectiveness of the proposed framework was confirmed through subjective experiments.

Future work includes the improvement of the performance of the arrangement assistance. For example, we would study a way for distinguishing seasonings from ingredients, a way for excluding irregular ingredients, and a way for introducing an ontology to integrate synonyms.

5. ACKNOWLEDGMENT

The cooking recipe dataset was provided from Rakuten Inc. via Rakuten Data Release[3].

6. REFERENCES

[1] M. Ueda, M. Takahata, and S. Nakajima, "User's food preference extraction for personalized cooking recipe recommendation," *Proc. 2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation*, pp. 98–105, Oct. 2011.

[2] Y. Shidochi, T. Takahashi, I. Ide, and H. Murase, "Finding replaceable materials in cooking recipe texts considering characteristic cooking actions," *Proc. ACM Multimedia 2009 Workshop on Multimedia for Cooking and Eating Activities*, pp. 9–14, Oct. 2009.

[3] K. Tsukuda, T. Yamamoto, S. Nakamura, and K. Tanaka, "Plus one or minus one: A method to browse from an object to another object by adding or deleting an element," *Proc. 21st Int. Conf. on Database and Expert Systems Applications*, pp. 258–266, Aug. 2010.

[4] L.L. Thurstone, "A law of comparative judgement," *Psychological Review*, vol. 34, no. 4, pp. 273–286, July 1927.