Recipe Search for Blog-type Recipe Articles Based on a User's Situation

Takuya KADOWAKI
Graduate School of
Informatics, Kyoto University
Yoshida-Honmachi, Sakyo,
Kyoto, Japan
kadowaki@dl.kuis.kyoto-u.ac.jp

Shinsuke MORI
Academic Center for
Computing and Media Studies,
Kyoto University
Yoshida-Honmachi, Sakyo,
Kyoto, Japan
forest@i.kyoto-u.ac.jp

Yoko YAMAKATA
Graduate School of
Informatics, Kyoto University
Yoshida-Honmachi, Sakyo,
Kyoto, Japan
yamakata@dl.kuis.kyoto-u.ac.jp

Katsumi TANAKA
Graduate School of
Informatics, Kyoto University
Yoshida-Honmachi, Sakyo,
Kyoto, Japan
tanaka@dl.kuis.kyoto-u.ac.jp

UbiComp'14 Adjunct, September 13 - 17, 2014, Seattle, WA, USA

ACM 978-1-4503-3047-3/14/09$15.00.
http://dx.doi.org/10.1145/2638728.2641327

Abstract
Many homemakers feel stressed when deciding on the menu of the day [?]. Even though they have a vague idea of some dishes, it is not easy for them to make the idea clear. Therefore, this research proposes a system that finds recipes corresponding to the user's vague requirements. On the Web, many blog-type recipes describe not only a recipe itself, but also the reasons why a given recipe were created or selected by the page authors. Firstly, the system extracts the author's reasons for the creation or selection of a recipe from the blog-type recipes. Secondly, the system lets the user input his/her present situation or feelings which must include his/her vague requirement and mine his/her reasons for recipe selection from the input. Finally, the system outputs recipes that meet the user's vague requirements by associating the reasons for recipe selection from the user's input text with the reasons accompanying the blog-type recipes.

Author Keywords
Blog-type recipe, cooking, situational search.

ACM Classification Keywords
H.1.2 [MODELS AND PRINCIPLES]: User/Machine Systems—*Human information processing*.

Introduction

Most Web search engines assume that a user roughly knows what they wish to find (e.g., a food recipe) and requires them to input a query word set that represents the properties of the searched target object (e.g., "beef stew"). However, even though users have vague requirements for the target object, they often cannot concretely explain their search needs. As many web pages present not only the information about the object, but also the reasons why the object was created or selected by the page authors, we consider that such reasons of creation and selection are exactly the kind of information that should be matched with the user's vague requirements. Therefore, this research is intended to discover the objects that match the user's vague search requirements by letting the user input a description of their situation or feelings, extract phrases that represent the author's reasons for creating or selecting a recipe, and find objects (recipes) that are accompanied by reasons corresponding to the user's vague requirements. To implement the system as a concrete search task, this research uses a cooking recipe search; one of the most popular services on the Web.

Searching for a recipe is a typical task in which a user has vague search requirements and cannot concretely determine the target information or object. If the user has decided on the name of the dish to be searched such as "beef stew" or the name of an ingredient such as "bitter melon", existing recipe search engines can sufficiently find suitable recipes. However, if the user has a rather vague search requirement such as "I want to find recipes that can make me feel refreshed after working overtime," it is often difficult for them to produce a concrete name for the required dishes. Therefore, it would be helpful if the user could input their situation into the proposed recipe search engine to discover suitable recipes.

In order to realize such a recipe search service for users, this research treats blog-type recipes as the search target. "Recipeblog [**?**]," one of the biggest blog-type recipe portal sites, provides around 600,000 blogs, which include not only the blogger's personal log, but also the recipe they created or selected at that time. The personal log usually contains information about the reasons why the blogger created or selected the given recipes. However, the personal log also contains information that is not directly related to the recipe. Hence, in order to know why the blogger created or selected the recipe, it is necessary to extract the statements that represent the reasons for creating or selecting the recipe from the personal log. Firstly, we categorized reasons of creation or selection of a recipe by analyzing 1,000 Recipeblog articles manually. Secondly, we trained SVM (Support Vector Machine) models to extract reason statements using manually analyzed data. These SVM models are used for extracting not only the reasons from the recipe blogs, but also the users' reasons in the context of their situations or feelings. Finally, we implemented the system as a web application and evaluated our proposed method.

Related research

Many recipe search services are available on the Web. In Japan, Cookpad [**?**], the biggest recipe portal site, has more than 1.7 million recipes and 20 million monthly users, while Rakuten-Recipe [**?**] has more than 810,000 recipes. In the United States, Food.com provides more than 475,000 recipes, while Allrecipes.com and FoodNetWork.com have more than ten million monthly users. Google also offers a recipe search service. On such recipe search services, a user routinely searches for a recipe using keywords of items that should be included in the target recipe, such as the name of a dish or its ingredients. However, as mentioned in the introduction, the motivations for searching for a recipe are various and the existing recipe search services cannot satisfy all of a user's motivations.

Several existing methods achieve recipe search using not only keywords included in the recipe, but also the user's tastes, feelings, or consumption of stock ingredients. Freyne et al. [?] propose a personalized recipe recommendation system. They investigated that preference of a food was related to the preference of the component ingredients of the food by analyzing 8701 preferences and ratings provided by 183 users on recipes and food items, and constructed an algorithm that recommended recipes using collaborative filtering. Morishita et al. [?] developed "the menu search system by feelings," which proposed a menu suitable for the user's feelings as "time," "taste," "variety," "fatigue," "price," and "easiness." In addition, to efficiently utilize ingredients, which cannot be consumed in one dish, Kihara et al. While most research has been on recipe recommendation and retrieval, Kuo et al. [?] proposed a method to recommending sets of recipes by user-specified ingredients..

This research aims to realize a recipe search based not only on the information included in a recipe, but also the surrounding information of the recipe, such as why the recipe was created or selected by the author. Druck [?] leverages "recipe attributes" (e.g. "comfort dish", "refreshing", "creamy" etc.) to build a system that predicts what users would say about a recipe. Like this research, he also uses textual information linked to the recipes. However, while he uses reviews, in most cases, which were written about only recipe, we use weblogs which were written about not only recipe but also personal affairs. This research differs from his research in the way.

Other domains allow users to use several methods when searching for a target object according to the information surrounding the target rather than a keyword included in the target. However, even if a web page gives the impression of being "interesting" or makes people "cry", such a page does not always include the keywords "interesting" and "cry" as part of its text data, meaning that a user cannot find a suitable web page by inputting the keywords based on the user's impression. Shoji et al. [?] solved this problem using people's reactions toward the target web page on web communication data such as "twitter." Tabelog [?], a restaurant search service in Japan, searches not only the restaurant information but also the surrounding information such as "reviews" and "user scores" to satisfy the user's niche needs.

Recipe search with a user's vague requirement

When a user searches for an object, the following three levels describe how much the user imagines the target object:

Type 1 A user can assign the keywords/phrases which the target object contains (e.g., a recipe using an apple).

Type 2 A user can describe the condition that the target object should satisfy (e.g., a recipe for something sweet).

Type 3 Even though the user cannot describe the target object, (s)he can judge whether the proposed recipe suits him/herself. (e.g., I have to cook for my children. I'm so tired. My mother gave me a lot of fruit yesterday.)

If a user is of type 1, they may input the keywords/phrases as a query for the search engine to simply find an object, which has the keywords/phrases as a part of its text. Existing search engines partly achieve the search request of type 2 using a semantic SEO technique [**?**], but it is insufficient. Moreover, a type 3 search request cannot be solved by merely addressing the target object itself; thus, it might be necessary to get additional information from the information surrounding the target object to complement the gap between the target and the user's requirements. Therefore, this research enables type 2 and 3 search requests. Notice that "a search request of a user who wants to find a recipe" should be matched with "a reason why a recipe author created or selected the recipe". Therefore, we discover such reasons in the blog-type recipes, which are essentially weblogs that include a recipe.

Suppose that a user submits their request as text. You may think our purpose can be achieved just by finding a blog-type recipe, which has many statements of high similarity with the user's request. However, even though most of the words overlap between the two statements, they can express completely different categories of reason. For example, when the statement is "since my children don't like anchovies, I used canned tuna," the category of the reason should be "cook for someone." On the other hand, when the statement is "since I was out of anchovies, I used canned tuna," the category of reason should be "lack of ingredients". Therefore, our proposed method finds recipes whose reasons of creation match the user's requests, not only in the cosine similarity, but also their categories of reason.

Analysis of reasons of recipe creation and selection in Recipeblog

In this section, we analyze the reasons of recipe creation and selection in blog-type recipes.

We collected 52,369 blog-type recipes provided by Ameba[1], posted between November 1, 2012 and October 31, 2013, which are also linked in Recipeblog.

Firstly, we defined the categories of reasons for recipe creation and selection. We extracted statements containing reasons from 20 randomly selected blog articles and categorized the statements manually. As a result, we created 18 categories, as shown in **??**. Secondly, nine annotators extracted the reasons from the 1,000 randomly selected blog articles, which were accompanied by at least one reason. The extracted reasons were then classified according to their categories.

Result of analysis

The incidence of each reason category is shown in **??**. The most common category, with 19.6%, is category (a): "the chef has specific ingredients or dishes." Since categories (a), (i), (g), and (r) occupy 56.1% of the total, it is assumed that these four categories are especially important. It is possible that one statement expresses multiple reasons of categories. However, after investigating how many statements were assigned to each reason category, we found that 92.8% of all statements were assigned to one reason category, 6.86% to two, and 0.387% were assigned to three categories. None of the statements were assigned to four or more reason categories. Since more than 90% of all statements were assigned exclusively to one category, the proposed categories are considered appropriate for analyzing reasons of creation and selection of recipes.

[1] http://ameblo.jp/

Table 1: Categories of reasons for creation and selection.

ID	Category name	Examples of corresponding statements
a	The chef has specific ingredients or dishes.	"There is a lot of pumpkin in my house."
b	The chef does not have specific ingredients or dishes.	"Unfortunately, I was out of demiglace at that time."
c	There are specific cooking utensils.	"Since I bought new cooking utensils, I wanted to try them out."
d	There are not specific cooking utensils.	"I don't have a cake stand for it! :("
e	The chef has enough time.	"Since the children stayed late at school, I might have time to cook it :-D"
f	The chef does not have enough time.	"For this reason, I was so busy today."
g	It fits the season or climate.	"It is hard to get out of bed on a cold morning."
h	There are specific events.	"I cooked it quickly when my friend came the other day."
i	It fits the tastes or desires of the chef.	"I wanted to eat rice cakes."
j	It fits the feeling or health condition of the chef.	"When I can't get motivated for cooking."
k	It is motivated by the experience of the chef.	"My mother often made steamed bread for me."
l	It is motivated by the experience of someone who eats the dish.	"She said, "It is delicious when a salad is wrapped in lettuce.""
m	A specific ingredient is compatible with another specific ingredient.	"Needless to say, eggplant is compatible with tomato sauce."
n	A specific ingredient is compatible with a specific dish.	"Season-fresh onion is very compatible with salad!"
o	A specific dish is compatible with another specific dish.	"This time, I put the TOMPEI YAKI (a Japanese dish) on toast."
p	It agrees with the tastes or desires of someone who eats the dish.	"My daughter has been asking me to make this since last summer."
q	It agrees with the feeling or health condition of someone who eats the dish.	"My husband and daughter seem to have a cold."
r	It is motivated by as a consumer investigator or by participating recipe contest.	"Sukiya gave me the ingredients of gyuudon as a trial."

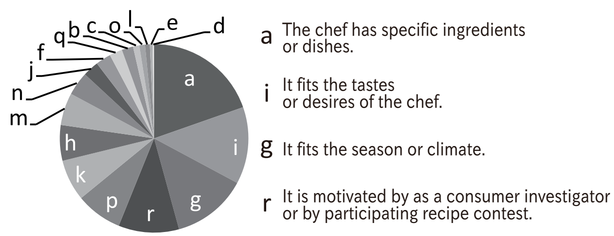

Figure 1: The incidence of each reason category (the alphabetic letters (a) to (r) in this figure correspond to ID letters in **??**).

Recipe search algorithm

In this section, we describe the algorithm for recipe search based on reasons of creation and selection.

Outline of the process

The overall process flow is shown in **??**. Step 1 and 2 are pre-processing steps. When a user searches for recipes, the system requests that they input their situation, which is tweeted on "twitter" (e.g., what they have experienced today or what they want for today's dishes).

Figure 2: The recipe search based on reasons of creation and selection — the overall process flow.

The algorithm is as follows:

Step 1: Extract the blogger's personal log from the blog articles.

Since reasons of creation and selection always appear in the user's personal log, the blogger's personal log statements are extracted from each blog article using a language model [**?**, **?**] and an ingredient name dictionary. The language model was built from 100,000 recipes collected from Cookpad [**?**], and the dictionary was built from 422,150 recipes collected from Cookpad.

Step 2: Extract the reasons of creation or selection from the personal log and classify them.
The reason statements are extracted automatically from the personal log, which was extracted in Step 1. The extracted reasons are classified into each category, and the reasons are extracted in the same manner from the user's input text.

Step 3: Find the recipes introduced in the personal log of blog articles for the same reasons as the user's input text.
For each reason extracted from the user's input text, the proposed system searches blog articles that have similar statements and reason category as the user's input statement. The greater number of blog statements that match the user's input, the more relevant the blog is to the user's requirements. The system selects and outputs the top K recipes in terms of the number of matched statements as the search results.

Extraction and category classification of reason of creation and selection
According to the investigation in the previous section, 92.8% of statements describing the reason of creation and selection had only one category. Therefore, in our method, we assumed that one statement, which describes the reason of creation or selection, can be assigned to only one category.

Classifiers
We considered that the statements assigned to the reason are different in terms of wording from the statements not assigned to the reason. Therefore, first, the statements of reasons were extracted using a classifier trained by all the statements of reasons extracted manually in the previous section, and assigned as positive samples. The training data also retains the statements not assigned as reasons as negative samples. We then apply a category classifier, discussed below, to classify the extracted statements into each category.

Classifier for reason statement extraction
A binary classifier was trained to assign the statements of reasons as a positive class and all other statements as a negative class. Since the number of statements in the positive class was 1,808 while the number of statements in the negative class was 28,176 in the blog-type recipes, we under-sampled the negative samples such that the number of statements in the negative class became the same as the positive class. A SVM (Support Vector Machine [**?**]) was employed for the classification.

Classifier for reason category classification
When the training data is insufficient, the classification performance significantly decreases. Therefore, in this research, we focused on eight of 18 categories, which have more than 90 statements. The categories were a, g, h, i, k, m, p, and r. All the remaining statements in the ten categories b, c, d, e, f, j, l, n, o, and q were unified into one category, hereafter, represented by the ID "s". We then built a nine-class classifier, which classifies the input data into one of the nine categories (a, g, h, i, k, m, p, r, and s). Since the fewest number statements of the nine categories was 99, we under-sampled all nine categories such that the number of statements in each category became 99. A SVM was employed for the classification.

Recipe search based on user's input
The system finds blog articles relevant to the user's input. The relevance is estimated according to the similarity between the reasons extracted from the article and the reasons extracted from the user's input text.

Suppose that X reason statements are extracted from a user's input text. Each statement is represented by r_x and its category is represented by c_x $(1 \leq x \leq X)$. In addition, suppose that Y reason statements are extracted from blog article B. Each statement is represented by R_y^B and its category is represented by C_y^B (where $1 \leq y \leq Y$). For each reason statement r_x (respectively R_y^B), a document vector $\boldsymbol{r}_x$ (respectively $\boldsymbol{R}_y^B$) is generated by the composition of the words' vector contained in the statement.

Algorithm

1. The degree of coincidence $score_{x,y}^B$ between statement r_x and R_y^B is calculated as:

$$score_{x,y}^B = \begin{cases} \dfrac{\boldsymbol{r}_x \cdot \boldsymbol{R}_y^B}{|\boldsymbol{r}_x||\boldsymbol{R}_y^B|} & (\text{ if } c_x \text{ and } C_y^B \text{ are equal.}), \\ 0 & (\text{Otherwise}), \end{cases}$$

   where $|\cdot|$ is the size of a vector.
2. The relevant score, $BScore^B$, between blog article B and the user's input is calculated as:

$$BScore^B = \frac{\sum_{x=1}^{X} \sum_{y=1}^{Y} score_{x,y}^B}{score\_non\_zero\_num^B},$$

   where $score\_non\_zero\_num^B$ is the number of combinations of x $(1 \leq x \leq X)$ and y $(1 \leq y \leq Y)$ whose score, $score_{x,y}^B$ is not 0.
3. The system finds the top K blog articles in terms of $BScore^N$ and outputs the recipes introduced in the articles as the search results.

Experiments and Considerations

In this section, we describe the classification accuracies introduced in the previous section.

Evaluation for personal log extraction
We evaluated the algorithm of the personal log extraction using 1,000 blog articles which contained at least one reason statement for each. Totally, the 1,000 articles contain 50,298 statements with 30,697 reason statements. In the result, 27,215 statements were extracted correctly from the 1,000 articles. The system gave an F-measure of 0.897, a precision of 0.908, a recall of 0.887, and an accuracy of 0.876. We considered the F-measure of 0.897 high enough for the following processes.

Evaluation for reason extraction and category classification
We used Libsvm [**?**] as the linear SVM classifier and Libsvm's default value as the parameter (C-SVC value is 1). The test data was selected such that the ratio of positive and negative samples became the same as the ratio of actual blog articles. Generating a document vector based on a composition of an original word's vector contained in a statement was considered insufficient to extract and classify the reasons and categories. Therefore, the following four pre-processing steps were taken to generate a document vector from each statement.

Pre-processing 1: All words are converted into the base form.

- e.g. "(use) (*infl.*)" → "(use) (*infl.*)"
  , where *infl.* stands for inflectional ending

We consider the difference of inflected forms of the words contained in statements ineffective for judging whether the statement is a reason or not and for deciding its reason category. Thus, all words are converted into the base form using the morphological analyzer JUMAN [**?**].

Pre-processing 2: The part-of-speech label is assigned to each word.

- e.g. "(enjoyment)" → "(enjoyment)/Noun"

Often the meanings of two words differ, despite being spelled in the same way. Such differences can be recognized by their part-of-speech. The variation in meanings is useful for judging whether the statement is a reason and to judge its reason category. Therefore, every word is labeled not only according to its spelling but also to its part-of-speech using the morphological analyzer JUMAN [**?**].

Pre-processing 3: The numbers of word types of specific categories contained in the statement are added as new elements to the document vector.

- e.g. "(Japanese radish)/F (and) (checken)/F (*cmo*) /T (*cmi*) (put into) (*ending*)"
  , where *cmo* and *cmi* means case marker for the direct object and indirect object, respectively.
  Hereafter, "F" means "food" and "T" means "tool".
  → the number of appearances of F: 2, those of T: 1

The numbers of word types such as "(F)ood" and "(T)ool" are useful for classification, especially in the categories a, b, c, d, and m of **??** ("the chef has specific ingredients or dishes," "the chef does not have specific ingredients or dishes," "there are specific cooking utensils," "there are not specific cooking utensils," and "a specific ingredient is compatible with another specific ingredient," respectively). Thus, all words are labeled using the named entity recognizer for the recipe processing proposed in [**?**], and the numbers of word types of "F" and "T" contained in the statement are added as new elements to the document vector.

Pre-processing 4: The number of words used for explaining a reason (such as "because," "since," "as," etc.) in the statement is added as a new element to the document vector.

- e.g. "(today) (*cms*) (cold) (*infl.*) (Japanese nabe) (*cmi*) (made) "
  → Since "" means "because" in Japanese, the number of terms used for explaining any reason is one.

In Japanese, the terms "" "" "" "" "" are used to give a reason in a statement. Such terms are more likely to appear in the reason statements than other statements. Therefore, the number of terms in the reason statement is added as a new element to the document vector.

To evaluate which pre-processing step is useful for the classifications, we performed all combinations of the pre-processing steps from 1 through 4 and compared the classification results.

Evaluation for reason statement extraction

The highest F-measure value was given when pre-processing steps 1 and 4 were adopted, when both of the numbers of training data in the positive and negative classes were 1,628, respectively. The F-measure value was 0.266 (precision 0.163, recall 0.733, accuracy 0.756). To estimate the behavior of the F-measure value when the training data is increased, we calculated the F-measure values while down sampling the training data in "2^n". The results are shown in **??**.

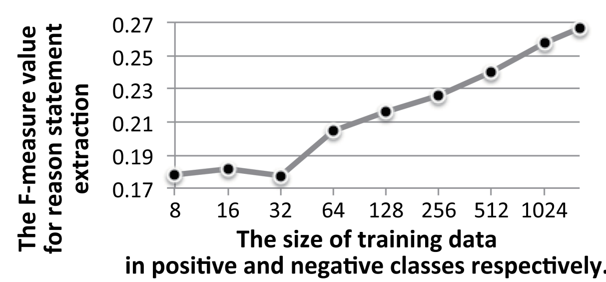

Figure 3: The size of training data and its F-measure value for reason statement extraction.

As shown in **??**, the F-measure value continues to rise, meaning that although the F-measure value was low in the present condition, the extraction accuracy will be high when the training data increases.

Evaluation for reason category classification

The F-measure value of each category is shown in **??**; the labels of the graphs on the right side of the figure, such as "# 1101 #" indicate which pre-processing step was adopted. The n-th digit corresponds to n-th pre-processing step (pre-processing n). "1" means the corresponding pre-processing step was adopted and "0" means it was not. For example, "# 1101 #" means that pre-processing steps 1, 2, and 4 were adopted. We show the results when no pre-processing step was adopted("# 0000 #") and when only one of four pre-processing step was adopted(e.g. "# 1000 #") as a baseline. The result of "# 1101 #" was shown in which the average F-measure value was the highest of all combinations of the pre-processing steps.

Figure 4: The F-measure value for reason category classification.

Although the F-measure values varied between categories, the average F-measure value was sufficiently high at 0.794. Therefore, the proposed method for reason category classification must improve with the discovery and application of a suitable feature for each category.

Evaluation for recommended recipes in our approach
To evaluate our approach, we compared the recommended recipes by our method and by the ordinary recipe search method. In order to eliminate the influence of reason statements mis-extraction, manually extracted reason statements were used in the evaluation.

We prepare three types of fictitious users with concrete situations. Two of them had vague requirements for recipes and they could only describe their situation as "it was so busy today". Hereafter, we refer these two scenario as "situational 1" and "situational 2". One of them had concrete requirement as "I bought Pacific saury with good price." We refer this scenario as "procedural". We input the descriptions of the situations to the proposed system and obtained top 10 results for each scenario. Also we made queries which could be introduced from each scenario and found 10 recipes which contained the queries as a part of the recipes as an ordinal recipe search algorithm. Five examinees evaluated each recipes recommended by the two method with five-point scale. **??** shows the average and maximum scores of satisfaction.

Table 2: The average and the maximum(in a parenthesis) of satisfactory degree

Type of scenario	[proposed method] search by situation	[ordinal method] search by queries
Situational 1	2.48 (4.40)	2.46 (4.00)
Situational 2***	3.48 (5.00)	1.70 (2.80)
Procedural**	1.50 (5.00)	2.00 (5.00)
Whole average**	2.49 (4.80)	2.05 (3.93)

** It was significantly different in 1% level between averages.
*** It was significantly different in 0.1% level between averages.

As shown in the result, the proposed method obtained higher score than the ordinal method in two situational scenarios while the ordinal method got higher score in procedural one. It means that the proposed method is useful when a user cannot make his/her own requirement clear as mentioned in section 1.

Summary and Challenges for the future

In this research, we proposed a recipe search system, which locates suitable recipes for the user according to their situation and matches the user's requirements to the author's reasons for creating or selecting a recipe for their blog. Firstly, we analyzed 1,000 blog-type recipes and found that the reasons of creation and selection of recipes could be classified into 18 reason categories. Secondly, we devised an algorithm for extracting reason statements from an article and classified the reason statements into the 18 reason categories. Although the current F-measure value of reason extraction was only 0.266, this is expected to increase when the training data increases, according to the analysis of the result. We constructed a recipe search system, which found compatible recipes with the user's input situation in terms of recipe selection. You can try the demo system at `http://www.dl.kuis.kyoto-u.ac.jp/~kadowaki/recipeblog_for_segovia/recipe_search.html`.
Only Japanese is available in the current version. As future work, we would like to design a more effective method for visualization of the search results via a field experiment.

Acknowledgements

Part of this research was supported by JSPS KAKENHI Grant Numbers 24240013, 23700144, and 23500177.