ReciPic: A Tool for Generating Infographic from Recipe Procedure Text

Lechang Zhang
Ochanomizu University
Tokyo, Japan
zhang.lechang@is.ocha.ac.jp

Takayuki itoh
Ochanomizu University
Tokyo, Japan
itot@is.ocha.ac.jp

***Abstract*—Food is one of the basic human needs and an important part of human culture. With the development of Internet, it has become increasingly common for people to share recipes on recipe websites. However, in most cases, these recipes contain only text information about the cooking procedures. It is difficult for users to have a big picture about the food materials and cooking methods at a glance. In this paper, we propose a tool for generating infographic from recipe procedure text, to supplement the pure text information on the recipe websites. This tool accepts one step text of the cooking procedures, and then segments the text stream into appropriate units. After that, it extracts the food material, cooking tool, cooking action segmentation and label them with F, T, Ac (F: Food material, T: Cooking tool, Ac: Cooking action) respectively. Then, it analyzes the grammatical relationship between the extracted words. Finally, it matches the words with prepared pictures and place them with appropriate positions in an infographic and represents it to users.**

***Index Terms*—Recipe/Cooking, User support, Visualization**

I. Introduction

Food is one of the most important part of our life. Human has been using simple processing methods like fire to process food before eating for more than 250,000 years. Cooking, as one of the food processing ways, can not only improve food safety, but also increase the appetite of consumers. In recent years, more and more people choose to use recipe websites to look for recipes, cooking advice or to share their own cooking ideas. The most popular recipe websites in Japan is Cookpad [1], which has over 50 million monthly users in Japan, and an additional 42 million monthly users across 74 countries, with 3 million recipes. In order to support such a large user community and improve their using satisfaction, it is necessary to analyze the recipes in recipe websites.

A recipe in recipe websites simply contains the recipe name, ingredients, and cooking procedures. Among these components, cooking procedure is the key part that users will follow. However, they are often displayed by pure text on the websites. It is difficult for users to grasp the valuable information and have a big picture about the food materials and cooking methods at a glance. For users who are new to cooking, it is especially hard to understand because they are not familiar with the process, food materials and the usage of cooking tools. Even though there are some recipes which also contain picture descriptions, those pictures usually show only the finished dishes to attract users rather than the cooking process. It is still not easy for users to comprehend the cooking procedures.

In this paper, we propose ReciPic, an interactive visualization tool to make the cooking process clearer, simpler and more understandable. Our tool allows users to enter recipe procedure step text, and then transform it into the corresponding infographic. Infographics, just as its name implies, are graphics which convey information. They are visual representation of any kind of information or data. We apply infographics to represent text information based on the fact that visual information is easier to understand than text information. Abundant research conducted in learning area showed how efficient graphics are in improving the learning comprehension and efficiency of students [2]. Reading recipes and cooking are also a form of learning. With the help of infographics, we suppose users can understand the procedure text information effectively.

II. Related Work

A. Procedure text analysis

In order to make computers understand human languages better, there are plentiful research conducted in the field of natural language processing to analyze text and extracting semantic structure that computers can easily read and understand.

Maeta et al. [3] described a method which takes recipe procedure text as the research object and converts it into a flow graph. This method first performs word segmentation on the procedure text, then identify the recipe named entities such as food, tools, action by the chef, etc. Finally, it builds the flow graph that uses recipe named entities as vertices and relationships between them as edges.

Hamada et al. [4] also proposed a similar method but take textbooks for cooking programs as their research object. Whereas Terashima et al. [5] analyzed the install instruction text and generated a flow graph based on their analysis.

These studies focus on procedure text analysis and flow graph generating. Our research also refers to their text analysis method but ultimately generates the infographic to assist users.

B. Recipe corpus

Recognizing useful terms, also known as Named Entity Recognition (NER), is a pretty important task in natural lan-

2375-0138/23/$31.00 ©2023 IEEE
DOI 10.1109/IV60283.2023.00012

guage processing. There are many studies that analyze newspaper articles and extract named entities out of it. However, little research has been done in analyzing recipe procedure texts.

Sasada et al. [6] discussed the definition of recipe terms and built a recipe corpus based on their definition. They also developed a tool that can automatically recognize the named entities of recipe procedure text.

Harashima et al. [7] introduced a new recipe dataset called Cookpad Parsed Corpus, which contains linguistic annotations for cookpad recipe procedure texts. They randomly extract recipes from cookpad recipe dataset and annotated recipe procedure sentences with morphemes, named entities, and dependency relations.

These studies concentrate on extracting useful information out of recipe procedure texts and provide a recipe corpus. Our research also refers to their recipe terms definition and classify different word segmentation parts into different groups by their definition.

C. Infographic

Infographics are graphical representation of text or data. They are broadly used in research to display information efficiently. Iwata et al. [8] proposed a method which uses infographic to visualize the source code to help coding beginners to learn faster. Wu et al. [9] discussed a system which takes recipe website information as input and generates infographics of recipe procedure texts.

These studies use infographics to improve peoples understanding in one particular material. Our research also makes use of the advantages of infographics, and proposes an interactive tool to transform text into infographics.

III. Proposed Method

A. Text analysis

To understand the meaning of the input, ReciPic will perform text analysis on it.

In text analysis, it will first segment the input into appropriate units and extract the useful parts, then classify different parts into three categories (namely food materials, cooking tools and cooking actions), and finally analyze the dependency relation between the segmented parts. In addition, ReciPic will also complement the omitted cooking tools before generating the infographic.

1) Word segmentation: In order to generate infographic from the input recipe procedure text, we first analyze the input text and segment the text into useful parts.

There are two popular Japanese text analysis tools. One is called kytea [10] and the other is called mecab [11]. We chose mecab for the reason that mecab has a dictionary called mecab-ipadic- NEologd which is constantly updated, while kytea has not been updated for many years. Here, we have randomly picked 5 recipes from cookpad to compare the accuracy of kytea and mecab (with mecab-ipadic-NEologd dictionary). The accuracy will be evaluated by Formula (1), and the result is shown in Table I.

$$\text{Accuracy of word segmentation} = \frac{\text{Number of correct food/tool/action}}{\text{Total number of food/tool/action}} \quad (1)$$

TABLE I
Results of accuracy test

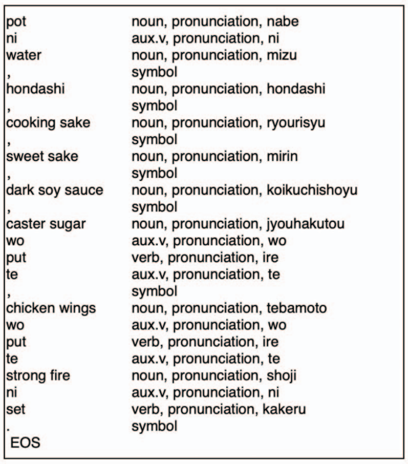
Accuracy	kytea	mecab
Accuracy of food	67.31%	96.15%
Accuracy of tool	71.43%	91.43%
Accuracy of action	97.44%	91.03%
Total Accuracy	82.42%	92.73%

As we can see, the total accuracy of word segmentation of kytea is about 10% lower than mecab. Kytea performs poorly especially in food and tool segmentations. Even though kytea does well in action segmentations, mecab is good enough for dealing with the cooking actions in most recipe procedure texts.

To read and analyze user's input on the server side, we chose server side language Node.js with its framework express to keep the language consistency between client side and server side. We will take a user input as an example to explain the word segmentation process in ReciPic.

[Input]: Put water, hondashi, cooking sake, sweet sake, dark soy sauce, caster sugar and the chicken wings in the pot, and set the strong fire.

First, ReciPic reads the input and parses it to a Node.js child process that runs mecab segmentation instruction in the terminal. Then it will generate the segmentation result as you can see in Figure 1.



pot	noun, pronunciation, nabe
ni	aux.v, pronunciation, ni
water	noun, pronunciation, mizu
,	symbol
hondashi	noun, pronunciation, hondashi
,	symbol
cooking sake	noun, pronunciation, ryourisyu
,	symbol
sweet sake	noun, pronunciation, mirin
,	symbol
dark soy sauce	noun, pronunciation, koikuchishoyu
,	symbol
caster sugar	noun, pronunciation, jyouhakutou
wo	aux.v, pronunciation, wo
put	verb, pronunciation, ire
te	aux.v, pronunciation, te
,	symbol
chicken wings	noun, pronunciation, tebamoto
wo	aux.v, pronunciation, wo
put	verb, pronunciation, ire
te	aux.v, pronunciation, te
strong fire	noun, pronunciation, shoji
ni	aux.v, pronunciation, ni
set	verb, pronunciation, kakeru
.	symbol
EOS	

Fig. 1. Word segmentation result

It consists of segmented words, parts of speech, pronunciation, etc. EOS in the end means End of Sentence. We simply want the segmented words such as cooking tools(noun), food materials(noun) and cooking actions(verb) which are useful in generating infographics. Therefore, we delete the unnecessary parts and connect the remaining parts with spaces. The word segmentation result is shown below:

[Output]: pot water hondashi cooking sake sweet sake dark soy sauce caster sugar put chicken wings put strong fire set

2) Word segmentation: After word segmentation, we successfully got the words needed for generating the corresponding infographic. But ReciPic has no idea about the meaning of each part. We need to classify the words in the word segmentation result to let ReciPic understand their general meaning.

Sasada et al. [6] and Harashima et al. [7] defined recipe terms in their work, I will refer to part of their recipe terms definition which is shown in Table II.

TABLE II
RECIPE TERMS DEFINITION

Recipe terms	Tag	Explanation
Food	F	F represents food materials, e.g. egg, rice, etc.
Tool	T	T represents cooking tools, e.g. fry pan, rice cooker, etc.
Action of chef	Ac	Ac represents the actions of chef, e.g. put, cut, etc.

The task of word classification can also be regarded as the task of Named-Entity Recognition, or NER for short. Here ReciPic used an NER tool for Japanese recipe terms recognition called PWNER [12]. PWNER only accepts input that has been segmented and separated by space. This is the reason that we separated the word segmentation result with spaces.

There still are some words such as hondashi and dark soy sauce that could not be recognized correctly by PWNER. To solve this problem, we build a word classification dictionary(Shown in Table III.) which contains the common vocabularies of food, cooking tool and cooking action to complement PWNERs result.

TABLE III
WORD CLASSIFICATION DICTIONARY

Category	Vocabularies
Food	hondashi, dark soy sauce, etc.
Tool	gas burner, microwave, etc.
Action of chef	cut, toast, etc.

The output of word classification step is shown below:

[Output]: pot/T water/F hondashi/F cooking sake/F sweet sake/F dark soy sauce/ F caster sugar/F put/Ac chicken wings/F put/Ac strong fire/T set/Ac

3) Dependency parsing: Now, we have seen how ReciPic performs word segmentation and classification on the input text. However, as we can find in the results of last step, we have already lost the dependency relations between the extracted words. For example, we do not know which food is operated by the action put, nor which tool is related to this action. Therefore, we also need to perform dependency parsing on the input text to determine the relations between words.

Here, ReciPic will use a Japanese dependency parsing tool called cabocha [13]. It is worth mentioning that cabocha is built based on mecab. They can use the same dictionary for text analysis. This will not cause inconsistent results between word segmentation and dependency parsing. In Wu et al.s [9] research, such errors eventually led to the generation of incorrect inforgraphic.

A dependency parsing result is shown below:

[Output]: pot/1-8 water/2-3 hondashi/3-4 cooking sake/4-5 sweet sake/5-6 dark soy sauce/6-7 caster sugar/7-8 put/8-12 chicken wings/9-10 put/10-12 strong fire/11-12 set/12-0

Each semantic unit is marked with two numbers connected with hyphen. The number before the hyphen indicates the order of this unit, and the number after the hyphen indicates its grammatically related unit. Take "pot/1-8" as an example, "1" shows "pot" is the first semantic unit, and "8" shows "put" is its grammatically related unit. It is worth mentioning that the "0" in "set/12-0" means "set" is the last semantic unit.

Finally, ReciPic will combine the word classification result with the dependency parsing result, and check if there is any omitted part in the input, and complement it by a prepared dictionary, then generate the text analysis result.

A text analysis result is shown below:

[Output]: pot/T/1-8 water/F/2-3 hondashi/F/3-4 cooking sake/F/4-5 sweet sake/F/5-6 dark soy sauce/F/6-7 caster sugar/F/7-8 put/Ac/8-12 chicken wings/F/9-10 put/Ac/10-12 strong fire/T/11-12 set/Ac/12-0

B. Infographic generating

1) Rules for generating infographics: In order to generate the infographics, we prepared an image database which contains the illustrations of food and tools. Each data is stored as an object in MongoDB, it has two properties name and image address to indicate the name and image of a certain food or tool. The images themselves are stored in the cloud. It is worth mentioning here that in order to solve the problem of spelling inconsistencies in Japanese, I have prepared an array of all possible spellings for each food/tool name. In addition, because some actions need to be completed by the combination of two tools, I also prepared images of the combination of tools. The details are shown in Table IV.

TABLE IV
IMAGE DATABASE

Data examples
"name": ['green onion'], "url": "negi.png"
"name": ['bowl-chopstricks'], "url": "bowl-chopstricks.png"

After the illustrations of each food and tools are determined, ReciPic will combine them together to generate the final infographic. This research referred the position relationships between food and tools defined by Wu et al. [9].

The details are shown in Figure 2.

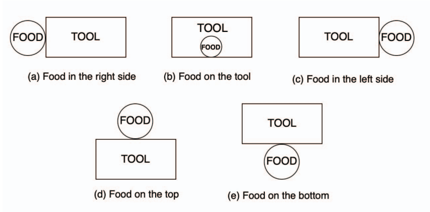


Fig. 2. Rules for generating infographics

Which rule to pick is determined by the cooking action relevant with the food and tool. For example, cooking action put means putting the food into the tool, so rule(d) will be picked. Action cut means cutting the food on the chopping-board, so rule(b) will be picked. Action wash means washing the food under the water-tap, so rule(e) will be picked.

2) Algorithm for processing text analysis result: ReciPic will generate infographic based on each combined unit of food, tool and action. We will introduce the algorithm for processing the text analysis result into combined unit.

The overview of the algorithm is shown in Figure 3. This algorithm accepts the text analysis result as input, and first splits the input string by space and stores the result in an array textAnalysisArr. Then it traverses the array, finds each element marked with Ac. For each Ac tagged element, find its related food and tool, combine them as a F/T/Ac unit, and then combine each F/T/Ac unit into an object containing a set of units. Next, it traverses the units object, and for the unit that contains no food or tool, find its related unit according to the related cooking action of this units action the related cooking action of one action is decided by the result of dependency parsing then use the prepared omitted-word completion dictionary to complete the remaining units with no tool. Finally, return the processed result.

```
Algorithm 1: Algorithm for processing text analysis result
Input: Text analysis result
Output: JSON data for generating the infographic
// get the infographic generating units
for i = 0 to textAnalysisArr.length do
    if textAnalysisArr[i][tag] = Ac then
        Find its related F/T
        Combine F/T/Ac as an infographic generating unit
    end
end
// process the infographic generating units
for each infographic generating unit do
    if unit[F] || unit[T] is empty then
        Find the related unit of this unit, and use its F/T
        Use the Omitted-word completion dictionary to complete the omitted
          tool
    end
end
Return the processed infographic generating unit
```

Fig. 3. Algorithm for processing text analysis result

A result of the processed infographic generating units is shown in Figure 4:

Finally, ReciPic will loop through each unit, find all the illustrations of food and tools in the image database, and look up for the infographic generating rule according to the action, then combine the illustrations of food and tools based on the corresponding rule. For the unit that contains a group of food, ReciPic will provide a 'next' button that allows users to see

```
{
  '1': {
    F: [ '水', '本だし', '料理酒', 'みりん', '濃口醤油', '上白糖' ],
    T: [ '鍋' ],
    Ac: [ '入れ' ]
  },
  '2': { F: [ '手羽元' ], T: [], Ac: [ '入れ' ] },
  '3': { F: [], T: [ '強火' ], Ac: [ 'かける' ] }
}
```

```
{
  '1': {
    F: ['水', '本だし', '料理酒', 'みりん',
     '濃口醤油', '上白糖', '手羽元'],
    T: [ '鍋' ],
    Ac: [ '入れ' ]
  },
  '3': { F: [], T: [ '強火' ], Ac: [ 'かける' ] }
}
```

Fig. 4. Algorithm for processing text analysis result

the food in sequence. And for the unit that contains word fire in its tool list, ReciPic will display a fire illustration at the bottom.

C. Experimental result

We used two different inputs to show the result of our tool ReciPic.

[Input1]: Put water, hondashi, cooking sake, sweet sake, dark soy sauce, caster sugar and the chicken wings in the pot, and set the strong fire.

[Infographic]: The result is shown in Figure 5.

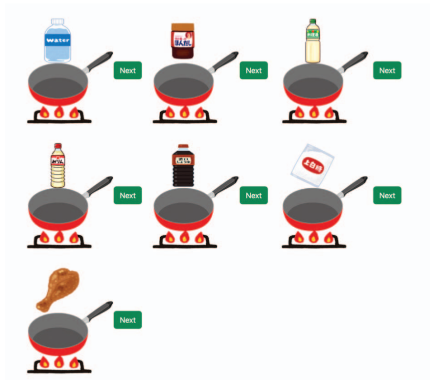


Fig. 5. Infographic of input 1

As we can from the result, food water, hondashi, cooking sake, sweet sake, dark soy sauce, caster sugar and chicken wings has been successfully extracted and present in the infographic. Tool pot and action put, set has also been extracted and present in the result. Besides, the result showed the order of food to be put, users could see them in sequence by clicking the next button. However, the infographic did not show the sequence of two actions put and set. This is because the infographic generating rules of ReciPic could only combine food with tool, when there are two tools to be combined, it will check if there is any special tool like fire and add it at the bottom. We aim to solve this problem by expanding the infographic generating rules of ReciPic.

[Input2]: Put rice, eggs, small leek and bacon into bowl and mix them well.

[Infographic]: The result is shown in Figure 6.

As we can from the result, food rice, egg, spring onion and bacon has been successfully extracted and present in the infographic. Tool bowl action put, mix has also been extracted and present in the result. Besides, the result showed the order

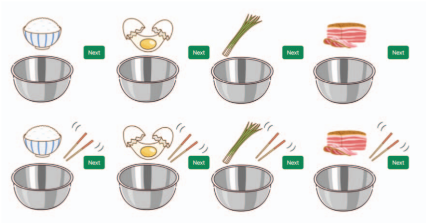


Fig. 6. Infographic of input 2

of food to be put, users could see them in sequence by clicking the next button. However, ReciPic presents action mix by combining its tool bowl and chopsticks with all the food materials of last action instead of mixing the food materials that has already been put in. This is because it is difficult to present the state of all the food materials that has been put in. We will discuss that in this situation, whether to combine all the food pictures and put them in the bowl or use other method in the future.

IV. Evalution

In order to evaluate the effectiveness of the proposed method, I experiment ReciPic with several recipes from recipe websites. To compare with Wu et al.'s [9] method, I chose the same 10 recipes from cookpad as input(Shown in Table V), and evaluate the accuracy of ReciPic by measuring how many food and cooking actions of the input can be correctly displayed in the generated infographic.

TABLE V
Image Database

Recipe Title	Number of Steps
Stir-fried Cabbage and Fish Sausage with Chili Mayo	4
Easy Boiled Potato and Tuna	2
Curry-flavored Pumpkin and Tuna	2
Stir-fried Cabbage	3
Stir-Fried Green Pepper and Bean Sprouts	4
Stir-fried Green Bean and Corn	1
Easy! Natto and Cubed Radish Kimchi Rice Bowl	2
Simple Breakfast: Stir-fried Vegetables and Fried Egg with Ketchup	4
Easy! Stir-fried Aburaage with Kelp Stock	2
Green Beans, Corn and Tomato Stir-fried with Pollack Roe Butter	2

Correctly displayed food means food that has been shown as illustration, and correctly displayed action means action that combines its related food and tool with a correct rule. The evaluation formulas are shown below:

$$\text{Precision} = \frac{\text{Number of correct food/action in the infographic}}{\text{Total number of food/action in the infographic}} \quad (2)$$

$$\text{Recall} = \frac{\text{Number of correct food/action in the infographic}}{\text{Total number of food/action in the input text}} \quad (3)$$

$$\text{F-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

And the result is shown in Table VI.

TABLE VI
Evaluation result of ReciPic

	Food	Action
Precision	88.89%	53.85%
Recall	82.35%	50.00%
F-Score	85.50%	51.85%

As we can see in the result, the general accuracy of ReciPic is higher than Wu et al. [9]'s method. This is because ReciPic has a better performance in text analysis, and has an algorithm to deal with the situation of omitted food/tool. However, the precision and recall of food and action are still not ideal, especially for actions, the result shows ReciPic can only deal with half of the input. We will try to solve this issue in the future.

V. Conclusion

In this paper, we introduced ReciPic, which is an interactive tool that allows users to enter recipe procedure text, and transforms it into infographic then represents it to users. We explained the key components of ReciPic in details, and experimented it with two input examples.

From the experimental results we can see that there are still many issues that can be improved in ReciPic. In the future, we plan to solve the problems we mentioned in experimental result, test ReciPic with a wider range of input and complement the text analysis dictionaries and the illustration database, to make it more applicable for more recipe texts. In addition, we will extract other words such as adjective and the quantity of ingredients, and discuss how to present them in the infographic.

References

[1] Cookpad company, https://cookpad.com/.

[2] ANGLIN, Gary J.VAEZ, Hossein; CUNNINGHAM, Kathryn L. Visual representations and learning: The role of static and animated graphics. Handbook of research on educational communications and technology. Routledge, 2013. p. 854-905.

[3] MAETA, Hirokuni; SASADA, Tetsuro; MORI, Shinsuke. A framework for recipe text interpretation. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. 2014. p. 553-558.

[4] HAMADA, Reiko, et al. Structural analysis of cooking preparation steps in Japanese. Proceedings of the fifth international workshop on on Information retrieval with Asian languages. 2000. p. 157-164.

[5] Terashima Taihei, Karabe Shiho, Sato Tetsuji, Structural Analysis Method for Visualizing Procedural Text, 10th Forum on Data Engineering and Information Management (DEIM2018), P7-6 (2018).

[6] Sasada Tetsuro, Mori Shinsuke, Yamakata Yoko, Maeda Hirokuni, Kawahara Tatsuya, Construction of a Tagged Corpus for Defining and Automatically Recognizing Recipe Terminology, Natural Language Processing, Vol. 22, No. 2 (2015).

[7] HARASHIMA, Jun; HIRAMATSU, Makoto. Cookpad parsed corpus: Linguistic annotations of Japanese recipes. Proceedings of the 14th Linguistic Annotation Workshop. 2020. p. 87-92.

[8] Iwata Madoka, Hayashida Naohiko, Mukaida Shigeru, Saito Hajime, Yasuda Mitsutaka, Visualization of Source Code using Infographics, Information Systems Society of Education, 39th National Conference, F5-1, 449-450 (2014).

[9] Zixin Wu, Ushiama Taketoshi, Automatic Generation of Infographics for Assisting Understanding of Cooking Recipes, 13th Forum on Data Engineering and Information Management, 2021.03.

[10] Kytea, Kyoto Text Analysis Toolkit, http:// www.phontron.com/kytea/.
[11] Mecab, Yet Another Part-of-Speech and Morphological Analyzer, https://taku910.github.io/mecab/.
[12] PointWise-based Named Entity Recognizer, http:// www.ar.media.kyoto-u.ac.jp/tool/PWNER/.
[13] Yet Another Japanese Dependency Structure Analyzer, http://taku910.github.io/cabocha/.