A Framework for Recipe Text Interpretation

Hirokuni Maeta
Cybozu, Inc.
1-4-14 Koraku, Bukyo,
Tokyo, Japan
hirokuni.maeta@gmail.com

Shinsuke Mori
Academic Center for
Computing and Media
Studies, Kyoto University
Yoshidahonmachi, Sakyo-ku,
Kyoto, Japan
forest@i.kyoto-u.ac.jp

Tetsuro Sasada
Academic Center for
Computing and Media
Studies, Kyoto University
Yoshidahonmachi, Sakyo-ku,
Kyoto, Japan
sasada@ar.media.kyoto-
u.ac.jp

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

ACM 978-1-4503-3047-3/14/09...$15.00.
http://dx.doi.org/10.1145/2638728.2641333

Abstract
In this paper we describe a method for converting a recipe text into a meaning representation. The meaning representation is a flow graph, whose vertices are important word sequences in cooking (recipe named entity; NE) and edges denote relationships among them.

Our methods consists of three parts: word segmentation (WS), recipe NE recognition (NER), and flow graph construction. The first two processes are based on machine learning and are adapted to recipe texts. The last process is based on huristic rules.

As an evaluation we tested three processes on an annotated corpus. The results showed that WS and recipe NER achieved high accuracies and that flow graph construction is relatively difficult having a large room for improvement.

Author Keywords
Recipe text, Word segmentation, Named entity recognition, Flow graph construction, Text understanding

ACM Classification Keywords
H.5.1 [Information interfaces and presentation (e.g., HCI)]: Multimedia Information Systems.

Introduction

Recipes are one of the most successful Internet contents. Nowadays many users search for recipes when they prepare a dish or learn a new recipe. Some of them post their recipes to a recipe site. As it is well known, there can be many ways of describing how to make the same dish or similar ones. To allow a more intelligent search [10, 11] it is needed to abstract recipe texts.

One of the abstract representations for cooking instructions is a tree. Hamada et al. [2] have proposed to represent instructions for a dish by a tree and made an attempt at converting recipe texts into that representation with a small in-house data set taken from recipes written by professionals. Recently a recipe flow graph corpus has been made publicly available [4][1]. The recipes in this corpus are randomly taken from those written by general internet users. This allows us an objective evaluation of tree construction methods on a publicly available corpus.

In this paper, we describe our framework of natural language processing (NLP) to convert a recipe text into a rooted tree representation. The input of our problem is a recipe text which consists of several sentences describing how to prepare a single dish. First each sentence is segmented into words (without part-of-speech tags). This word segmentation (WS) is only required for the languages without clear word boundary. Next, some words are grouped together and annotated with recipe named entity (NE) tags. Finally, we connect the NEs to construct a rooted tree.

[1]The meaning representation in this corpus is a directed acyclic graph because of food ramifications, coreferences, etc. The average number of additional arcs is, however, very small and they are almost trees.

Table 1: Corpus statistics.

#recipes	#sentences	#NEs	#words
208	1,374	8,316	25,446

Similar researches [6] describes domain adaptation of WS, NE recognition (NER), and dependency parsing to the recipe text domain with the aim of converting recipe texts to a flow graph. This paper, however, lacks the flow graph construction part. Hamada et al. [2] describes a rule-based method to construct a tree from a recipe text. In their work, no WS evaluation was reported and the important word sequece recognition (recipe NER) problem has not been clearly defined.

Contrary to these works, we present an NLP framework for recipe interpretation starting from a raw text till the flow graph representation and give objective evaluations of each NLP. Our paper provides the solid baseline for future improvement in the recipe interpretation problem.

Recipe Flow Graph Corpus

As a test bed of the recipe interpretation problem, we adopt the recipe flow graph corpus (r-FG corpus) [4]. To our best knowledge, this is the only corpus annotated with an interpretation.

The r-FG corpus contains randomly crawled recipes in Japanese from a famous Internet recipe site. The specification of the corpus is shown in Table 1. The text part of a recipe consists of a sequence of steps and the steps have some sentences. Figure 1 shows an example.

All the concepts (entities and actions) appearing in the sentences are identified and annotated with a recipe NE tag. In addition, some of them are connected with

labeled arcs to form a flow graph representing the interpretation of the recipe. Figure 2 shows an example.

Each vertex of a flow graph corresponds to a recipe NE represented by a word sequence in the text and a recipe NE type such as food, tool, action, etc. Table 2 lists the recipe NE types along with the average number of occurrences per recipe. There is one special vertex, root, corresponding to the final dish.

An arc between two vertices indicates that they have a certain relationship. An arc has a label denoting its relationship type[2]. The most interesting relationships may be co-references and null-instantiated arguments. In Figure 2 for example, “macaroni” is equal to “pasta.” According to the world knowledge, macaroni is a sort of pasta, but in this recipe they are identical. Another example of a null-instantiated argument is the relationship between “heat” and “add.” Celery etc. should be added not to the initial cold Dutch oven without oil but to the hot Dutch oven with oil, which is the implicit result of the action “heat.”

1. **両手鍋で油を熱する。**
   (In a Dutch oven, heat oil.)
   セロリと青ねぎとニンニクを加え、1分ほど炒める。
   (Add celery, green onions, and garlic. Cook for about 1 minute.)

2. **ブイヨンと水とマカロニと胡椒を加えて、**
   (Add broth, water, macaroni, and pepper,
   パスタが柔らかくなるまで煮る。
   and simmer until the pasta is tender.)

3. **刻んだセージをまぶす。**
   (Sprinkle the snipped sage.)

Figure 1: A recipe example.

[2]The r-FG corpus defines 13 arc labels but we do not use them.

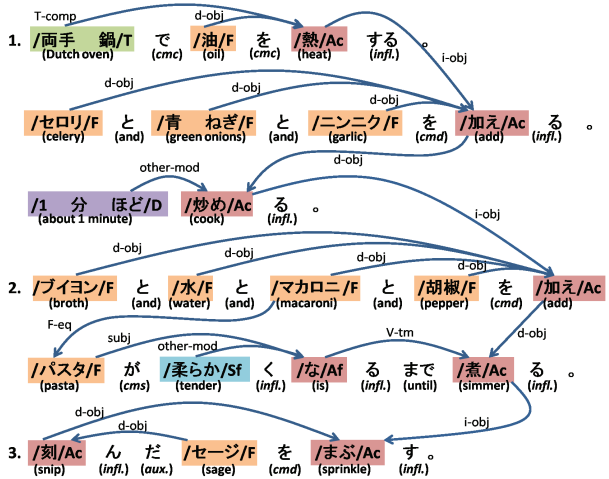

Figure 2: The flow graph of the example recipe.

Table 2: Recipe NE tags with frequencies per recipe.

Tag	Meaning	Freq.
F	Food	13.13
T	Tool	4.05
D	Duration	0.82
Q	Quantity	0.75
Ac	Action by the chef	14.74
Af	Action by foods	2.70
Sf	State of foods	3.45
St	State of tools	0.35
Total	–	39.98

Framework for Recipe Interpretation

In order to realize a practical system by the state-of-the-art NLP, we divide the recipe interpretation problem into the following three processes and combine them in the cascaded manner.

1. Word segmentation (WS) (Figure 1 → Figure 3)

2. Recipe named entity recognition (recipe NER) (Figure 3 → Figure 4)

3. Flow graph construction (Figure 4 → Figure 2)

In this section we explain each process one by one.

Word Segmentation
In this paper we use a word segmenter described in [8] because we can easily build corpora for the recipe domain by the reduction of annotation cost. In this approach, we can focus our resources on the annotation of difficult parts [5], and this reduces the annotation cost.

Word segmentation in the pointwise approach is formulated as a binary classification problem as in [9]. A word segmenter, given a sentence $x_1 x_2 \cdots x_h$, estimates boundary tag b_i between characters x_i and x_{i+1}. Tag b_i = 1 indicates that a word boundary exists, while b_i = 0 indicates that a word boundary does not exist. This classification problem can be solved by support vector machines [1].

Recipe Named Entity Recognition
An output of NER is vertices for a flow graph. We also use an NER described in [6] for the reduction of annotation cost. In the training step this NER estimates the parameters of a classifier based on logistic regression [1] from sentences fully (or partially) annotated with NEs. At the run-time, given a word sequence, the classifier enumerates all possible BIO tags for each word with their probabilities, and it searches for the tag sequence of the highest probability satisfying the tag sequence constraints.

1. 両手 鍋 で 油 を 熱 する 。
   セロリ と 青 ねぎ と ニンニク を 加え 、 1 分 ほど 炒め る 。
2. ブイヨン と 水 と マカロニ と 胡椒 を 加え て 、
   パスタ が 柔らか く な る まで 煮 る 。
3. 刻 ん だ セージ を まぶ す 。

Figure 3: A word segmentation result.

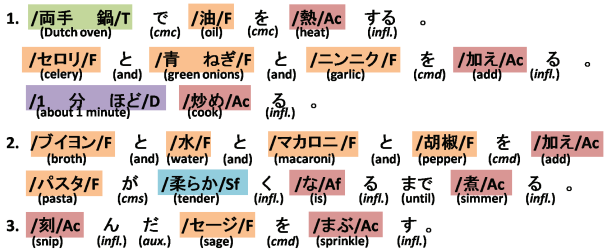

Figure 4: A recipe named entity recognition result.

Flow Graph Construction
We adopt the rule-based arc generator proposed by [2]. This method simply connects an NE with an action NE at the back of the NE. Thus this flow graph constructor finally outputs a tree from vertices of NEs.

This method is designed only for Japanese recipe parsing. Since Japanese dependencies almost go from left to right, this approach is reasonable for flow graph generation of Japanese recipe texts.

The original method [2] resolves coreference using in-house dictionaries. From the paper, the coverage of the dictionaries seems to be small. Thus we do not have specific rules for coreference resolution. To solve this type of relationships, we need an ontology of a broad coverage in the recipe domain.

Table 3: F-measure of each task.

Task	Input	F
WS	Raw texts	98.6
NER	Gold WS results	90.7
Flow graph const.	Gold WS and NER results	69.9

Evaluation

In this section we give experimental results and evaluations of our framework. We tested each step independently (WS, NER, and flow graph construction) to clarify the problem for a practical recipe text interpretation.

The performance measurement is F-measure. The units are a word, a recipe NE, and a tuple $(\langle \boldsymbol{w}_s, c_s \rangle, \langle \boldsymbol{w}_e, c_e \rangle)$, respectively. Here, $\boldsymbol{w}_s$ and c_s are the word sequence of the out-going vertex of the arc and its recipe NE type, respectively. $\boldsymbol{w}_e$ and c_e are those of its in-coming vertex. A tuple is correct if and only if both elements match with those of an arc in the manually annotated data.

Settings of Word Segmentation and Named Entity Recognition

The WS and NER are based on machine learning. The followings are the settings for the evaluation.

WS We used an SVM-based word segmenter [8] trained on the following corpora.

1. Balanced Corpus of Contemporary Written Japanese [3] containing fully segmented 53,899 sentences from newspaper articles, books, magazines, whitepapers, Web logs, and Web QAs.
2. The partially segmented sentences derived from 208 recipes in the r-FG corpus and additional 208 recipes annotated with recipe NE types. In the experiment, we excluded the test part in 10-fold cross validation. Thus we built 10 models in total from 208/10 + 208 recipes.
3. Partially annotated 1,651 sentences crawled from another recipe Web site[3].

NER We used an NER [6] based on the logistic regression combined with DP-based best path search. The model is trained on the corpus 2. used in the WS training in the same way. Thus we built 10 models in total for 10-fold cross validation.

Discussion

Table 3 shows the accuracy of WS, NER, and the flow graph construction.

As we see in the table, the WS accuracy is enough high and it can be said that this part is almost well solved with these language resources and the method. The NER accuracy of the model trained on less than 3,000 sentences is as high as the general NER whose accuracy is about 90% with about training 10,000 sentences. This part is ready to be used. Thus our framework dividing the recipe interpretation problem into the three processes is a good strategy for practical uses. We may try to adopt some more sophisticated techniques proposed for NER, such as CRFs, knowledge aquisition from the Web, etc. to improve NER.

[3] **http://park.ajinomoto.co.jp/** (accessed on the 16th June, 2014).

The flow graph construction task is the most difficult. The accuracy is comparable to the original one [2]. And we can say that there is a large room for improvement. Although the data size is not large, it is worth trying some machine learning techniques for this part as well. The arc label estimation is also a future work. This is solved straightforwardly as a classification problem given an arc.

Conclusion

In this paper, explained our framework to solve recipe interpretation problem from a raw text. The framework consists of the three processes combined the cascaded manner. We evaluated them independently and found that the last flow graph construction part is the bottleneck with a large room for improvement. The WS is almost as accurate as the in-domain case and is ready to be used. The recipe NE recognition is as good as the general NE recognition but we need some more works to make it practically useful. As a research direction, it may be worth trying a flow graph construction based on machine learning. And world knowldege including an ontology [7] improves more.

Acknowledgments

This work was supported by JSPS Grants-in-Aid for Scientific Research Grant Numbers 26280084, 24240030, and 26280039.

References

[1] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research 9* (2008), 1871–1874.

[2] Hamada, R., Ide, I., Sakai, S., and Tanaka, H. Structural analysis of cooking preparation steps in Japanese. In *Proc. of the IRAL00* (2000), 157–164.

[3] Maekawa, K. Balanced corpus of contemporary written Japanese. In *Proc. of the 6th Workshop on Asian Language Resources* (2008), 101–102.

[4] Mori, S., Maeta, H., Yamakata, Y., and Sasada, T. Flow graph corpus from recipe texts. In *Proc. of the LREC14* (2014), 2370–2377.

[5] Mori, S., and Neubig, G. Language resource addition: Dictionary or corpus? In *Proc. of the LREC14* (2014), 1631–1636.

[6] Mori, S., Sasada, T., Yamakata, Y., and Yoshino, K. A machine learning approach to recipe text processing. In *Proc. of the 1st Cooking with Computer Workshop* (2012), 29–34.

[7] Nanba, H., Doi, Y., Tsujita, M., Takezawa, T., and Sumiya, K. Construction of a cooking ontology from cooking recipes and patents. In *Proc. of the CEA14* (2014).

[8] Neubig, G., Nakata, Y., and Mori, S. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proc. of the ACL11* (2011), 529–533.

[9] Sassano, M. An empirical study of active learning with support vector machines for Japanese word segmentation. In *Proc. of the ACL02* (2002), 505–512.

[10] Wang, L., Li, Q., Li, N., Dong, G., and Yang, Y. Substructure similarity measurement in Chinese recipes. In *Proc. of the WWW08* (2008), 978–988.

[11] Yamakata, Y., Imahori, S., Sugiyama, Y., Mori, S., and Tanaka, K. Feature extraction and summarization of recipes using flow graph. In *Proc. of the SocInfo13*, LNCS 8238 (2013), 241–254.