# Flow Graph Corpus from Recipe Texts

**Shinsuke Mori**[1], **Hirokuni Maeta**[2], **Yoko Yamakata**[3], **Tetsuro Sasada**[4]

[1,4]Academic Center for Computing and Media Studies, Kyoto University
[2,3]Graduate School of Informatics, Kyoto University
Yoshidahonmachi, Sakyo-ku, Kyoto, Japan
[1]forest@i.kyoto-u.ac.jp, [2]maeta@ar.media.kyoto-u.ac.jp,
[3]yamakata@dl.kuis.kyoto-u.ac.jp, [4]sasada@ar.media.kyoto-u.ac.jp

### Abstract

In this paper, we present our attempt at annotating procedural texts with a flow graph as a representation of *understanding*. The domain we focus on is cooking recipe. The flow graphs are directed acyclic graphs with a special root node corresponding to the final dish. The vertex labels are recipe named entities, such as foods, tools, cooking actions, etc. The arc labels denote relationships among them. We converted 266 Japanese recipe texts into flow graphs manually. 200 recipes are randomly selected from a web site and 66 are of the same dish. We detail the annotation framework and report some statistics on our corpus. The most typical usage of our corpus may be automatic conversion from texts to flow graphs which can be seen as an entire understanding of procedural texts. With our corpus, one can also try word segmentation, named entity recognition, predicate-argument structure analysis, and coreference resolution.

**Keywords:** Procedural text, Flow graph, Understainding

## 1. Introduction

One of the goals of natural language processing (NLP) is understanding. And we know it is very hard to give a definition to understanding for general texts. For procedural texts, however, we can almost define understanding. Momouchi (1980) proposed to represent a procedural text as a flow graph and tried an automatic conversion. Because this attempt was made before the corpus-based NLP era, there is no language resource made publicly available.

In this paper, we present our attempt at annotating procedural texts with a flow graph as a representation of *understanding*. The most typical usage of our corpus may be automatic conversion from texts to flow graphs. Thus we focus on recipe texts to make it realistic by limiting the vocabulary and expressions. Another advantage of recipe domain is that cooking actions deploy in a narrow area under control. So we can take video of the chef executing the instructions written in the text. Actually researchers in computer vision are taking cooking videos in a kitchen equipped with cameras and other sensors (Hashimoto et al., 2010). Our corpus shares some recipes with them. Thus their data and ours enable symbol grounding researches (Regneri et al., 2013; Yu and Siskind, 2013) or sentence generation from video or image (Yang et al., 2011) in realistic situations.

Actually there is an attempt at converting a recipe into a *tree* to match it with its cooking video in order to index the video (Hamada et al., 2000). The definition of their recipe tree is, however, not as mature nor precise as our flow graph definition that we present in this paper. In fact, while we fixed the flow graph definition, we encountered a variety of linguistic phenomena, including ellipsis, coreference of nouns and verbs, part-of relationship, etc. In this paper we present a definition for the recipe flow graph and our corpus constructed manually. With our corpus, one can try domain adaptation of word segmentation (Neubig et al., 2011), named entity recognition (Sang and Meulder, 2003), predicate-argument structure analysis (Marcus et al., 1993; Yoshino et al., 2013), coreference resolution (Yang et al., 2004), and entire understanding of recipe texts.

## 2. Recipe Flow Graph

In this section we give an overview of our recipe flow graph corpus, which we named the r-FG corpus.

### 2.1. Recipe Text

As the corpus source we collected recipes from an Internet recipe site[1] in Japanese. Nowadays there are many recipe sites in various languages. Thus we have enough recipes to work with.

The texts are written by Internet users and contain some typos and infrequent expressions. We left them as they are. So it is challenging for NLP to process them automatically.

A recipe consists of three parts: a title, an ingredient list, and cooking instruction sentences (we refer to them as "recipe text" for simplicity hereafter), as shown in Figure 1. Our flow graph corpus corresponds to the text part.

### 2.2. Recipe Flow Graph

A recipe text describes cooking instructions for a dish. Some instructions have an order relationship and some do not. For example, the chef needs to cut carrots before boiling it, but does not need to do it before cutting potatoes. In order to represent such relationships we use flow graphs.

Almost all recipes can be represented by a tree (Hamada et al., 2000). But we found that some recipes can not be represented by a tree. For example, in a recipe an ingredient is split into some parts and used separately. Thus generally a recipe flow graph is a directed acyclic graph (DAG) $G = (V, A)$, where $V$ is a set of vertices and $A$ is a set of arcs. Figure 2 shows an example [2]. Vertices correspond to foods,

[1] `http://cookpad.com/` accessed on 2014/02/01.

[2] *cms*, *cmd*, and *cmi* mean the case markers for the subject, direct object, and indirect object respecitively. *infl.* means inflectional ending of a verb or an adjective.

Title:

| | |
|---|---|
| ホットドッグ | (Baked Hot Dog) |

Ingredients:

| | |
|---|---|
| • フランクフルト 8 つ | (8 frankfurters) |
| • ホットドッグパン 8 つ | (8 hot dog buns) |
| • 豆入りのチリ 1 缶 (14.5 オンス) | (1 (14.5 ounce) can of chili with beans) |
| • カットオニオン 1/2 カップ | (1/2 cup onion (diced)) |
| • 千切りチェダーチーズ 2 カップ | (2 cups shredded cheddar cheese) |
| • マヨネーズ | (mayonnaise) |
| • マスタード | (mustard) |
| • 甘味料 | (sweet relish) |

Steps:

1. 各ホットドッグパンの内側にマヨネーズ、マスタード、甘味料を広げる。
   (Spread the inside of each hot dog bun with mayonnaise, mustard and sweet relish. )
   フランクフルトを入れ、13×9”のオーブン皿に置く。
   (Fill with a frankfurter and place into a 13×9” baking dish.)
2. 各ホットドッグにチリ、チーズ、オニオンをふりかける。
   (Sprinkle each hot dog with chili, cheese, and onion.)
3. アルミホイルで覆い、オーブンに置く。
   (Cover with aluminum foil and place into the oven)
   そして、350 度で 45 分間焼く。
   (Then bake at 350 degrees for 45 minutes.)

Figure 1: Recipe of “Baked Hot Dog.”

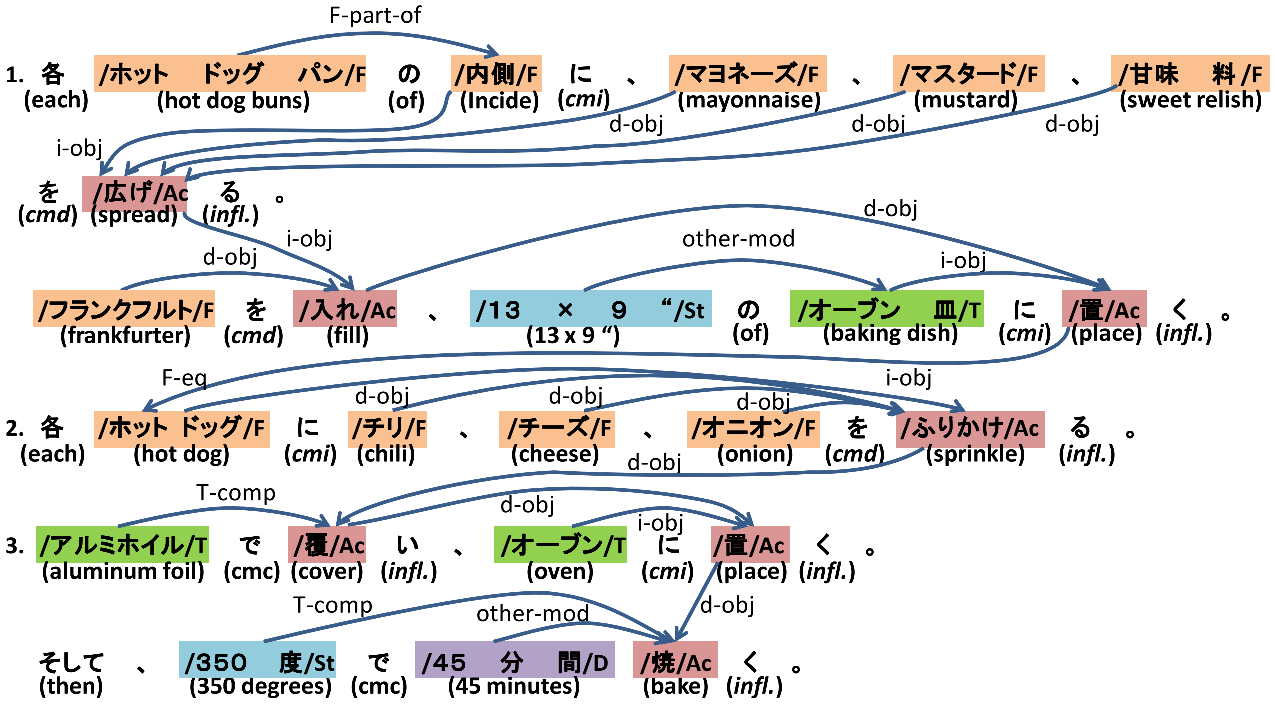

F-part-of
1. 各 (each) /ホット ドッグ パン/F (hot dog buns) の (of) /内側/F (Incide) に (cmi) 、 /マヨネーズ/F (mayonnaise) 、 /マスタード/F (mustard) 、 /甘味 料/F (sweet relish)
d-obj
d-obj
d-obj
i-obj
を (cmd) /広げ/Ac (spread) る (infl.) 。
i-obj
d-obj
other-mod
d-obj
i-obj
/フランクフルト/F (frankfurter) を (cmd) /入れ/Ac (fill) 、 /13 × 9 “/St (13 x 9 “) の (of) /オーブン 皿/T (baking dish) に (cmi) /置/Ac (place) く (infl.) 。
F-eq
d-obj
d-obj
d-obj
i-obj
d-obj
2. 各 (each) /ホット ドッグ/F (hot dog) に (cmi) /チリ/F (chili) 、 /チーズ/F (cheese) 、 /オニオン/F (onion) を (cmd) /ふりかけ/Ac (sprinkle) る (infl.) 。
T-comp
d-obj
i-obj
3. /アルミホイル/T (aluminum foil) で (cmc) /覆/Ac (cover) い (infl.) 、 /オーブン/T (oven) に (cmi) /置/Ac (place) く (infl.) 。
T-comp
other-mod
d-obj
そして (then) 、 /350 度/St (350 degrees) で (cmc) /45 分 間/D (45 minutes) /焼/Ac (bake) く (infl.) 。

Figure 2: The recipe flow graph of “Baked Hot Dog.”

tools, actions, etc. And arcs denotes relationships between them. Different from general DAGs, a recipe flow graph has a special vertex, root, corresponding to the final dish. We give explanations of vertices and arcs in the subsequent sections.

## 3. Verties

Vertices of a flow graph correspond to foods, tools, actions, etc. Their labels are composed of a character sequence in the recipe text and its type[3].

The character sequence is specified by the starting position and the ending position in the recipe text, because the same character sequence may appear in multiple places but their roles may differ. The character sequence is manually segmented into words following the same standard for Balanced Corpus of Contemporary Written Japanese (Maekawa et al., 2010) with inflectional endings separated

[3] In Japanese there is no obvious word boundary.

| NE tag | Meaning |
|---|---|
| F | Food |
| T | Tool |
| D | Duration |
| Q | Quantity |
| Ac | Action by the chef |
| Af | Action by foods |
| Sf | State of foods |
| St | State of tools |

Table 1: Named entity tags.

from adjectives, verbs, and auxiliary verbs as the only exceptions (Mori et al., 2014). The word boundary information for other parts is, however, not annotated manually. That is to say, some parts in a sentence have word boundary information and some do not.

In addition to a character sequence, a vertex has its type denoting foods, tools, actions, etc. as the label listed in Table 1. Although the label set contains verb phrases, we call it named entity (NE) tag set, because it can be recognized by a method similar to NE recognition (Chinchor, 1998; Sang and Meulder, 2003). Below we describe the NE types taken from our annotation guideline with typical examples. In the examples, word sequences are surrounded by slash symbols and an NE type sign follows the right slash symbol.

### 3.1. F: Food

Foods consist of ingredients, intermediate products, and the final dish in cooking. Here are typical examples.

**ex.)** /ホット (hot) ドッグ (dogs)/F

**ex.)** /甘味 (sweet) 料 (relish)/F

### 3.2. T: Tool

Clearly, objects such as cookwares, jars, bottles, and knives are tools. The followings are the examples appearing in Figure 1.

**ex.)** /アルミ (aluminum) ホイル (foil)/T

**ex.)** /オーブン (baking) 皿 (dish)/T

### 3.3. D: Duration

Expressions to denote duration of a cooking action, such as heating time, are tagged with D. This includes numbers and units, as shown in the following example.

**ex.)** /４５ 分 (minutes)/D 間 (for) /焼 (bake)/Ac く (*infl.*)

### 3.4. Q: Quantity

Expressions to specify the quantity of foods are tagged with Q. They are mainly number expressions followed by units. The following is a typical example.

**ex.)** /水 (water)/F /４００ ｃｃ/Q

### 3.5. Ac: Action by the Chef

Actions are also important in cooking. Among actions, Ac denotes those taken by the chef. They are mainly transitive verbs. The following is a typical sentence including an Ac.

**ex.)** /人参 (carrot)/F を (*cmd*) /切 (cut)/Ac る (*infl.*)

In this example, action 切る (cut) is tagged with Ac because it is the chef who cuts the carrot, although it is not clearly mentioned in the sentence who cuts the carrot. It depends on the context whether an action is taken by the chef or not.

### 3.6. Af: Action by Foods

Sometimes foods take actions. They are mainly intransitive verbs and copula expressions. Here is an example.

**ex.)** /水 (water)/F が (*cms*) /煮立 (boil)/Af
つ(*infl.*) たら (when)

In this example, action 煮立 (boil) is tagged with Af because what boils here is /水 (water)/F.

### 3.7. Sf: State of Foods

State of foods is an expression describing taste, color, etc. Here is an example.

**ex.)** /赤 (red)/Sf く (*infl.*) /な (get)/Af る (*infl.*)

### 3.8. St: State of Tools

Tools can also have a state. In our definition, expressions for characteristics of a tool are also tagged with St.

**ex.)** /深 (deep)/St い (*infl.*) /フライパン (frying pan)/T

There can be a verb denoting a change of the state of a tool. We decided to incorporate the verb part into St instead of introducing At (action by tool) as follows, because actions of tools are very infrequent and we want to avoid an inflation of NE tags.

**ex.)** /熱 (hot) く (*infl.*) な (get) った (*infl.*)/St
/フライパン (frying pan)/T

### 3.9. Others

There are some word sequences in a text that are not annotated with any NE tags above. Those words are ones to be tagged with O in the BIO tag set (Sang and Meulder, 2003). Typical ones are function words such as particles, inflectional endings, and symbols like commas and periods. There are some content words not annotated with an NE tag: conjunctions like “then”, adverbs like “again.” Since our source recipe texts are one of UCGs (User Generated Contents), they sometimes contain sentences stating other things than cooking instructions such as “my children like it.” These words are not annotated with NE tags.

## 4. Arcs

An arc denotes the relationship between two vertices. To classify relationships we defined arc labels listed in Table 2. Below we explain each of them with typical examples taken from our annotation guideline.

### 4.1. subj, d-obj, i-obj: Cases

A verb vertex (Af or Ac) takes some arguments. They are connected to the vertex by arcs annotated with subj (subject), d-obj (direct object), or i-obj (indirect object) depending on their syntactic relationships. Here are typical examples.

/チリ (chili)/F $\xrightarrow{\text{d-obj}}$ /ふりかけ (sprinkle)/Ac

/オーブン (oven)/T $\xrightarrow{\text{i-obj}}$ /置 (place)/Ac

In the above example, an arc goes from a vertex of a food or a tool to a vertex of an action. An arc, however, often goes from a vertex of an action to a vertex of another action as follows.

/人参 (carrot)/F $\xrightarrow{\text{d-obj}}$ /洗 (wash)/Ac $\xrightarrow{\text{d-obj}}$ /切 (cut)/Ac

This means that the result of an action /洗 (wash)/Ac, i.e. a washed carrot, is the direct object of another action /切 (cut)/Ac but not a carrot without being washed. This distinction is important in procedural texts.

### 4.2. F-comp: Food Complement

F-comp connects a vertex of a food to a vertex of an action that uses the food. In cooking, foods are not always cooked to make a final dish. For example, seasonings are used to adjust the taste, as described below.

/塩 (salt)/F $\xrightarrow{\text{F-comp}}$ /整え (adjust)/Ac

This says that the food /塩 (salt)/F is used in the action /整え (adjust)/Ac.

### 4.3. T-comp: Tool Complement

T-comp connects a vertex of a tool to a vertex of an action using that tool as follows.

/アルミ(aluminum) ホイル(foil)/T $\xrightarrow{\text{T-comp}}$ /覆(cover)/Ac

Tools which are not a container tend to be connected to an Ac with an arc of this label.

### 4.4. F-eq: Food Equality

There are some food expressions referred to repeatedly. This phenomenon is called coreference. Thus even if the word sequences are different, we connect them by an arc of this type as far as the object is the same in the real world. If the object is a food, the label is F-eq as the following example shows.

/牛肉 (beef)/F $\xrightarrow{\text{F-eq}}$ /肉 (meat)/F

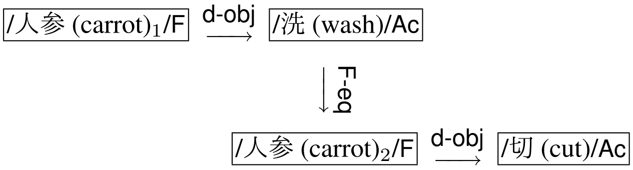
Sometimes the result of an action on a food is referred to by the food name.

人参$_1$ を洗う (wash the carrot$_1$)
その人参$_2$ を切る (cut that carrot$_2$)

In this case the carrot to be cut is the carrot which has been washed, but not the carrot before washing. Thus we get the following flow graph.

| Edge lable | Meaning |
|---|---|
| subj | Subject |
| d-obj | Direct object |
| i-obj | Indirect object |
| F-comp | Food complement |
| T-comp | Tool complement |
| F-eq | Food equality |
| F-part-of | Food part-of |
| F-set | Food set |
| T-eq | Tool equality |
| T-part-of | Tool part-of |
| A-eq | Action equality |
| V-tm | Head verb of a clause for timing, etc. |
| other-mod | Other relationships |

Table 2: Arc labels.



In many corpora annotated with coreference information, /人参 (carrot)$_1$/F and /人参 (carrot)$_2$/F in this example may be regarded as the same. In our corpus, however, we distinguish an object before an action and after that if the action affects the object as we mentioned in Subsection 4.1.

### 4.5. F-part-of: Food Part-of

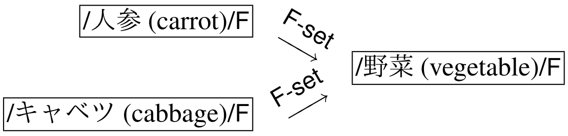
Two vertices of foods are connected with F-part-of, if one of them is a part of the other. Here is an example.

/ジャガイモ (potato)/F $\xrightarrow{\text{F-part-of}}$ /皮 (skin)/F

This annotation indicates that food /皮 (skin)/F is a part of another food /ジャガイモ (potato)/F.

### 4.6. F-set: Food Set

Sometimes a set of foods is referred to by a single expression, typically by a category name. In such a case, F-set connects vertices of foods to another vertex, whose word sequence may be the category name. Here is an example.



This annotation says that /野菜 (vegetable)/F refers to the set consisting of /人参 (ca-root)/F and /キャベツ (cabbage)/F.

### 4.7. T-eq: Tool Equality

Similar to F-eq, a tool expression refers to another tool. In such case we connect them by an arc of the label T-eq. Here is an example.

/圧力 (pressure) 鍋 (cooker)/T $\xrightarrow{\text{T-eq}}$ /鍋 (pot)/T

### 4.8. T-part-of: Tool Part-of

Similar to F-part-of, there are expressions referring only to a part of a tool. They are connected to the tool with an arc annotated with T-part-of as follows.

$$\boxed{\text{/鍋 (pot)/T}} \xrightarrow{\text{T-part-of}} \boxed{\text{/蓋 (lid)/T}}$$

This annotation indicates that a tool /蓋 (lid)/T is a part of a tool /鍋 (pot)/T.

### 4.9. A-eq: Action Equality

Sometimes an action verb is repeatedly written just to specify an object such as "carrot you've cut." We connect this verb to the verb which the chef has to really execute as follows.

$$\boxed{\text{/スライス (slice)/Ac}} \xrightarrow{\text{V-eq}} \boxed{\text{/切 (cut)/Ac}}$$

This clarifies that the chef does not need to execute the latter action connected to another action, In this example, the chef must /スライス (slice)/Ac an object but must not /切 (cut)/Ac the sliced object any more.

### 4.10. V-tm: Head of a Clause for Timing, etc.

Some clauses, whose head is a verb phrase annotated with Ac or Af, specify the timing or the condition of another action. To denote this, we annotate the arc with V-tm.

$$\boxed{\text{/沸騰 (boil)/Af}} \xrightarrow{\text{V-tm}} \boxed{\text{/入れ (add)/Ac}}$$

### 4.11. other-mod: Other Relationships

All the other modification relationships fall into this category. The most frequent ones are duration expressions D modifying an action Ac or quantity expressions Q modifying a food F. Here is an example taken from the recipe text in Figure 1.

$$\boxed{\text{/45 分 (minutes)/D}} \xrightarrow{\text{other-mod}} \boxed{\text{/焼 (bake)/Ac}}$$

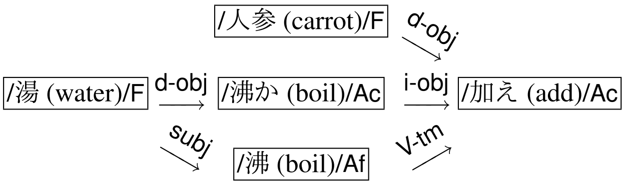
### 4.12. Reasons for Ramifications

As we mentioned in Section 2. our meaning expression for recipe texts is not a rooted tree like (Hamada et al., 2000), but a rooted DAG. The difference between a rooted tree and a rooted DAG is that some vertices have more than one outgoing arcs. The most typical reason which comes across to our mind is a food separation. An egg is separated into yolk and white and they are the direct objects of two different actions. Or a half of source is mixed with foods at the early stage and the rest is added to the result of some cooking actions.

A close look at our corpus, however, revealed that many ramifications are caused by coreference, etc. Let us take the following sentences for example.

お湯を沸かす (Boil water)
沸いたら人参を入れる (Add the carrots when it boils)

The flow graph for them is as follows.



In this example, the main instruction stream is the middle line starting from /湯 (water)/F to /加え (add)/Ac. And to specify the timing of the action /加え (add)/Ac, the chef waits for "it" to /沸 (boil)/Af[4]. Here the subject of /沸 (boil)/Af is the same as /湯 (water)/F.

## 5. Annotation Framework

In this section we detail our annotation framework.
The information we annotate to recipe texts is as follows. The example sentence is アルミホイルで覆います (cover with aluminum foil).

- An NE tag to a character sequence

  **ex.)** /アルミホイル/F で /覆/Ac います

- Word boundary information between each two characters in an NE character sequence

  **ex.)** /アルミ ホイル/F で /覆/Ac います

  where "います" is composed of three words ("い," "ま," and "す") but the annotator does not segment it into them

- Arc label between two NEs if they have any relationship defined in the standard

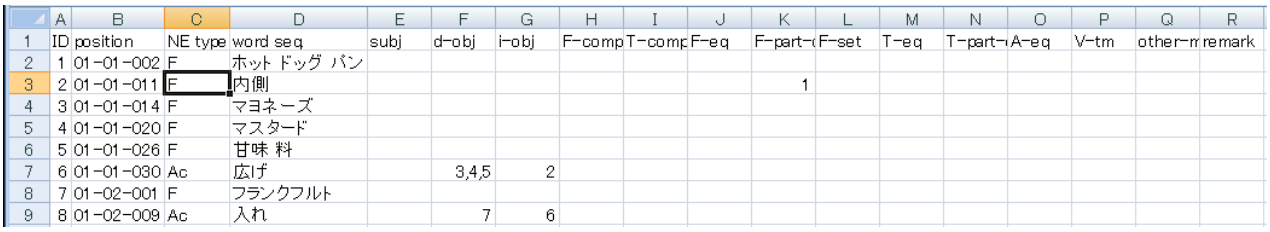
  **ex.)** $\boxed{\text{/アルミ ホイル/F}} \xrightarrow{\text{T-comp}} \boxed{\text{/覆/Ac}}$

We repeated the following work flow to annotate recipe texts with flow graphs.

1. Train a word segmenter (Neubig et al., 2011) and an NE tagger (Mori et al., 2012) from the currently available annotated corpus.
2. Run the word segmenter and the NE tagger (Mori et al., 2012) on the target recipe text. Then correct word boundary information and NE annotation (BIO tag) manually.
3. Make an empty spreadsheet with a header row indicating vertex information and relationships, and fill some left columns with the information of the NEs in the target recipe text (see Figure 3). The table does not have any arc annotations at this stage.
4. Fill the spreadsheet to make a flow graph. If there is an arc from a vertex $v_i$ to another vertex $v_j$ with a label $l$, the annotator fills the crossing cell of the column of $l$ and the row of $v_j$ with the ID number of $v_i$.

After finishing filling the spreadsheet, the annotator runs a program which checks whether the graph is a rooted DAG. The checking program runs on a web site, so the annotator posts a spreadsheet to the site and then receives some error messages or OK.

[4] In Japanese, the subject is omitted (zero pronoun) because it is semantically obvious.



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ID | position | NE type | word seq | subj | d-obj | i-obj | F-comp | T-comp | F-eq | F-part-( | F-set | T-eq | T-part- | A-eq | V-tm | other-m | remark |
| 2 | 1 | 01-01-002 | F | ホット ドッグ パン | | | | | | | | | | | | | | |
| 3 | 2 | 01-01-011 | F | 内側 | | | | | | | 1 | | | | | | | |
| 4 | 3 | 01-01-014 | F | マヨネーズ | | | | | | | | | | | | | | |
| 5 | 4 | 01-01-020 | F | マスタード | | | | | | | | | | | | | | |
| 6 | 5 | 01-01-026 | F | 甘味 料 | | | | | | | | | | | | | | |
| 7 | 6 | 01-01-030 | Ac | 広げ | | 3,4,5 | 2 | | | | | | | | | | | |
| 8 | 7 | 01-02-001 | F | フランクフルト | | | | | | | | | | | | | | |
| 9 | 8 | 01-02-009 | Ac | 入れ | | 7 | 6 | | | | | | | | | | | |

Figure 3: Spreadsheet for flow graph annotation.

| Source | #Sent. | #NEs (#Leaves, #Non-Leaves) | #Words | #Char. |
|---|---|---|---|---|
| Random (200 recipes) | 6.51 | 36.34 ( 13.91, 22.43) | 118.65 | 180.51 |
| *Nikujaga* (66 recipes) | 6.91 | 48.05 ( 22.12, 25.92) | 132.95 | 198.70 |

Table 3: Specifications of the flow graph corpus.

| NE type | Number |
|---|---|
| F | 11.87 |
| T | 3.83 |
| D | 0.67 |
| Q | 0.79 |
| Ac | 13.83 |
| Af | 2.04 |
| Sf | 3.02 |
| St | 0.30 |
| Total | 36.34 |

Table 4: Statistics on the NE tags.

## 6. Discussion

We converted 266 Japanese recipe texts into flow graphs manually. Among them 200 are randomly selected and 66 are taken from recipes of the same title "beef and potato" (*nikujaga*), a Japanese famous dish. It took 5-10 minutes for an annotator to annotate the sentences in a recipe with NE tags and about 40 minutes to fill the spreadsheet to form a flow graph. Below we show some statistics mainly on the randomly selected ones and discuss them[5].

### 6.1. Statistics on the Corpus

The average numbers of sentences, NEs, words, and characters in a recipe text are shown in Table 3. Note that the number of words is counted on the automatically segmented sentences except for the NE parts. From the table, for example, we can say that a sentence contains about six NEs.

The average numbers of NE types per recipe text in the random set are shown in Table 4. We can say that food (F) and action by the chef (Ac) are the most important in terms of frequency. Table 5 shows the statistics on arc labels in the random set. The table says that predicate-object relationships (d-obj and i-obj) are the most important in terms of frequency. These points are consistent with our intuition that recipe texts mainly describe what to do with what in which direction. The next frequent relationships are related to foods (subj, F-eq, and F-part-of).

| Arc label | Number |
|---|---|
| subj | 2.15 |
| d-obj | 15.67 |
| i-obj | 7.22 |
| F-comp | 0.65 |
| T-comp | 1.32 |
| F-eq | 3.15 |
| F-part-of | 2.37 |
| F-set | 0.15 |
| T-eq | 0.44 |
| T-part-of | 0.39 |
| A-eq | 0.53 |
| V-tm | 1.06 |
| other-mod | 3.54 |
| Total | 38.62 |

Table 5: Statistics on the arc labels.

The comparison between the totals in Table 4 and Table 5 says that the average number of arcs is slightly larger than that of vertices. If the graphs are trees, the number of arcs equals to that of vertices subtracted by one. Therefore the difference (38.62-36.34+1=3.28) indicates that how far our flow graphs are from trees.

Our recipe flow graph corpus enables us to compare one recipe or a set of recipes of the same title with others in various way.

### 6.2. Language Dependency

Ideally, a flow graph should be language independent. For example, the flow graph representation eliminates the difference between active voice and passive voice. However, the flow graph defined above is not a complete semantic representation of a recipe text. For example, we use shallow cases (syntactic case) to denote the relationship between an object and a verb.

To observe the language dependency we prepared a "baked hot dog" recipe in English and translated it into Japanese. Then two annotators constructed a flow graph from the texts in each language independently. Finally we compared them. Both of them have 23 vertices and all of them matched each other. The numbers of arcs are both 22. Among them 16 arcs matched including the arc la-

[5] The statistics on the *nikujaga* set are very similar to those on the random.

bel. The most frequent difference is the arc label of the cases. When we exchange d-obj and i-obj properly, 20 arcs matched (F-measure: 0.9). Thus we can say that our flow graph definition is almost language independent except for the predicate-argument part.

In Japanese i-obj has a strong tendency to be the destination (even if it is an F). Let us take the sentence of the second step in Figure 1, In Japanese the direct object of the verb ふりかけ (sprinkle) is チリ (chili) and ホットドッグ (hot dog) is its destination. In real action, the chef grasps チーズ (cheese) and moves it to the hot dog. We can classify the arguments from the semantic viewpoint into the main object and the destination in stead of syntactic d-obj and i-obj. Under that standard the annotation to English recipe text may be a little bit difficult.

# 7. Application

Our flow graph corpus has many potential applications. The most typical one is text analysis ranging from word segmentation to recipe text understanding. A flow graph can be regarded as a meaning representation of a sequence of procedures. Thus it can be used as the input of text generation systems for procedural texts. We can develop much more intelligent search engines than the current keyword matching. In this section we describe an overview of the potential applications of our corpus.

## 7.1. Text Analysis

The vertex labels have a word sequence segmented manually. Its position in the recipe text is specified. Thus we can convert the sentences in our corpus into partially segmented sentences. An example taken from Section 5. is as follows:

ア–ル–ミ | ホ–イ–ル | で | 覆 | い␣ま␣す,

where the symbols "|," "–," and "␣" mean word boundary, not a word boundary, and no information, respectively. There are some attempt at training a word segmenter on these partially segmented sentences (Tsuboi et al., 2008; Neubig and Mori, 2010; Neubig et al., 2011; Mori and Neubig, 2014). With our corpus, more accurate word segmenter in the cooking domain will be available.

With our flow graph and the recipe text, we can have sentences annotated with NEs in the cooking domain. One can try NE recognition task (Borthwick, 1999; McCallum and Li, 2003; Sang and Meulder, 2003) in this domain.

One can try a resolution of more complicated linguistic phenomena and test it on our corpus, including predicate-argument structure analysis (Marcus et al., 1993; Yoshino et al., 2013), coreference resolution (Yang et al., 2004), and entire understanding of recipe texts which converts a document into the flow graph.

## 7.2. Text Generation

In natural language generation (NLG) research, one of the most important problem is the input (Dale and Reiter, 2000). Our flow graph representation can serve NLG researchers as the source information. The vertices have a word sequence. Thus the main problem may be the selection of function words which describe the relationship among them. In addition there are many coreference relationships in our corpus and a reference expression generation is also important (Reiter and Belz, 2009).

It is also interesting to generate procedural texts from real cooking videos or still images if they are available (Regneri et al., 2013; Yang et al., 2011).

## 7.3. Others

The flow graph representation is useful for an intelligent recipe search. There are various ways of describing the same procedure or the same dish. For example, a phrase "add diced carrots to the pot" implies that the chef must dice carrots first. The flow graph representation derived from this is the same as that derived from "dice carrots and add to the pot." Our rooted DAG representation absorbs the difference in natural language level and enables more intelligent search (Wang et al., 2008; Yamakata et al., 2013).

A help system for a chef in the kitchen is also an interesting application. Hamada et al. (2005) proposed a cooking navigation system based on their tree representation (Hamada et al., 2000) that schedules the cooking procedures for two or more recipes. More generally Hashimoto et al. (2008) proposed a concept of smart kitchen which observes the cooking situation by computer vision, etc. and helps the chef by giving an advice in voice or with video. An automatic flow graph constructor trained on our flow graph corpus may contribute to these systems.

It is also interesting to extend our framework to other procedural texts and other sorts of texts, namely paper abstracts (Tateisi et al., 2014) or the world history.

# 8. Conclusion

In this paper, we reported the details of our flow graph corpus constructed from recipe texts in Japanese. We choose rooted directed acyclic graph as the meaning representation for recipe texts.

Since our flow graph corpus has manually corrected word sequences as the vertex labels with their positions in the text, one can train a word segmenter from it as a partially segmented corpus (Neubig and Mori, 2010). With NE tag information, our corpus allows NE recognition experiments (Sang and Meulder, 2003) in the recipe domain.

Our corpus can been seen as a representation of a natural language understanding for cooking procedural texts. It enables NLP researchers to try various problems ranging from predicate-argument structure analysis or coreference resolution to entire understanding of recipe texts.

# 9. Acknowledgments

This work was supported by JSPS Grants-in-Aid for Scientific Research Grant Numbers 23500177 and 24240030, and JSPS Grant-in-Aid for Young Scientists Grant Number 23700144. We are grateful to the annotators for their contribution to the design of the guidelines and the annotation effort.

# 10. References

Andrew Borthwick. 1999. *A Maximum Entropy Approach to Named Entity Recognition*. Ph.D. thesis, New York University.

Nancy A. Chinchor. 1998. Overview of muc-7/met-2. In *Proceedings of the Seventh Message Understanding Conference*.

Robert Dale and Ehud Reiter. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

Reiko Hamada, Ichiro Ide, Shuichi Sakai, and Hidehiko Tanaka. 2000. Structural analysis of cooking preparation steps in japanese. In *Proceedings of the fifth international workshop on Information retrieval with Asian languages*, number 8 in IRAL '00, pages 157–164.

Reiko Hamada, Jun Okabe, Ichiro Ide, Shinichi Satoh, Shuichi Sakai, and Hidehiko Tanaka. 2005. Cooking navi: Assistant for daily cooking in kitchen. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 371–374.

Atsushi Hashimoto, Naoyuki Mori, Takuya Funatomi, Yoko Yamakata, Koh Kakusho, and Michihiko Minoh. 2008. Smart kitchen: A user centric cooking support system. In *Proceedings of the 12th Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 848–854.

Atsushi Hashimoto, Naoyuki Mori, Takuya Funatomi, Masayuki Mukunoki, Koh Kakusho, and Michihiko Minoh. 2010. Tracking food materials with changing their appearance in food preparing. In *Proceedings of the Second Workshop on Multimedia for Cooking and Eating Activities*.

Kikuo Maekawa, Makoto Yamazaki, Takehiko Maruyama, Masaya Yamaguchi, Hideki Ogura, Wakako Kashino, Toshinobu Ogiso, Hanae Koiso, and Yasuharu Den. 2010. Design, compilation, and preliminary analyses of balanced corpus of contemporary written japanese. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*.

Mitchell Marcus, Grace Kim, and Mary Ann Marcinkiewicz. 1993. The penn treebank: A revised corpus design for extracting predicate argument structure. In *Proceedings of the International Symposium on Spoken Dialogue*, pages 77–81.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*, pages 188–191.

Yoshio Momouchi. 1980. Control structures for actions in procedural texts and pt-chart. In *Proceedings of the Eighth International Conference on Computational Linguistics*, pages 108–114.

Shinsuke Mori and Graham Neubig. 2014. Language resource addition: Dictionary or corpus? In *Proceedings of the Nineth International Conference on Language Resources and Evaluation*.

Shinsuke Mori, Tetsuro Sasada, Yoko Yamakata, and Koichiro Yoshino. 2012. A machine learning approach to recipe text processing. In *Proceedings of Cooking with Computer workshop*.

Shinsuke Mori, Hideki Ogura, and Tetsuro Sasada. 2014. A japanese word dependency corpus. In *Proceedings of the Nineth International Conference on Language Resources and Evaluation*.

Graham Neubig and Shinsuke Mori. 2010. Word-based partial annotation for efficient corpus construction. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1(Mar):25–36.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating nlg systems. *Computational Linguistics*, 35(4):529–558.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Computational Natural Language Learning*, pages 142–147.

Yuka Tateisi, Yo Shidahara, Yusuke Miyao, and Akiko Aizawa. 2014. Annotation of computer science papers for semantic relation extraction. In *Proceedings of the Nineth International Conference on Language Resources and Evaluation*.

Yuta Tsuboi, Hisashi Kashima, Shinsuke Mori, Hiroki Oda, and Yuji Matsumoto. 2008. Training conditional random fields using incomplete annotations. In *Proceedings of the 22th International Conference on Computational Linguistics*.

Liping Wang, Qing Li, Na Li, Guozhu Dong, and Yu Yang. 2008. Substructure similarity measurement in chinese recipes. In *Proceedings of the 17th international conference on World Wide Web*, pages 978–988.

Yoko Yamakata, Shinji Imahori, Yuichi Sugiyama, Shinsuke Mori, and Katsumi Tanaka. 2013. Feature extraction and summarization of recipes using flow graph. In *Proceedings of the 5th International Conference on Social Informatics*, LNCS 8238, pages 241–254.

Xiaofeng Yang, Jian Su, GuoDong Zhou, and Chew Lim Tan. 2004. An np-cluster based approach to coreference resolution. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 226–232.

Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. 2011. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Koichiro Yoshino, Shinsuke Mori, and Tatsuya Kawahara. 2013. Predicate argument structure analysis using partially annotated corpora. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*.

Haonan Yu and Jeffrey Mark Siskind. 2013. Grounded language learning from video described with sentences. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.