International Journal of Innovative Computing, Information and Control
Volume **9**, Number **1**, January **2013**
 ISSN 1349-4198
pp. 397–411

NATURAL LANGUAGE GENERATION METHOD USING AUTOMATICALLY CONSTRUCTED LEXICAL RESOURCES

Naho Ito and Masafumi Hagiwara

Faculty of Science and Technology
Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522, Japan
{ naho; hagiwara }@soft.ics.keio.ac.jp

Received November 2011; revised March 2012

Abstract. *In this paper, we propose a natural language generation method based on automatically constructed lexical resources. Many conventional approaches in sentence generation use manually constructed templates. Therefore, the variety of available sentences depends heavily on the quality and quantity of the templates, and the cost to construct these templates is very high. The proposed sentence generation method uses large-scale case frames and Google N-gram, which both are compiled automatically from Web documents. The proposed method uses words as an input. It generates a sentence from case frames, using Google N-gram as to consider co-occurrence frequency between words. Since we only use lexical resources which are constructed automatically, the proposed method has high coverage compared with the other methods using manually constructed templates. We carried out experiments to examine the quality of generated sentences and obtained satisfactory results.*
Keywords: Sentence generation, N-gram, Case frame

1. **Introduction.** As an important step for an interface of human and robot, it is necessary for the robots to use natural language. In the topic of dialogue system, the main goal of Natural Language Generation (NLG) is to investigate how computer programs can produce a high-quality natural language text from internal representation of information [1]. In recent years, the increasing feasibility of human-computer dialogue systems has prompted the need for better responses by generating diverse sentences.

Some NLG systems use grammar rules, much like parsers with semantic or syntactic grammars [2]. An example of a rule-based system is SURGE [3]. For Japanese, there is a dictionary IPAL [4]. In general, well generated grammar rules enable an NLG system to have wide coverage, be domain independent, and be reusable [3]. However, a great deal of time is required to design such a system.

Template-based approach to NLG refers to those in which the developer handcrafts a set of templates. Therefore, the performance heavily depends on the quality and quantity of the templates. Many dialogue systems which are domain dependent use this approach [5-8]. Generally, this approach is not applicable and reusable. It cannot generate the sentence that templates do not cover.

Other than these two approaches, Shibata proposed a method to select a sentence from Web that suits the dialogue [9]. This solves the problems of the cost to design the system and the coverage of the generated sentences. However, it is likely that the system chooses a sentence that does not fit the dialogue. It is difficult to choose an appropriate sentence to the dialogue's context and atmosphere.

Other than the topic of dialogue systems, NLG technique is needed in machine translation and question answering systems, etc. Sasayama et al. [10] used Japanese-English bilingual corpus in translation. Bosma et al. [11] focused on question answering system, and they applied sentence fusion to combining partial answers from different sources into a single more complete answer. Yang et al. [12] improved retrieval efficiency in question answering system, using classic Chinese literature's feature. However, few researches focus on sentence generation independently, and most approaches are unable to be applied to other topics.

One available solution to these problems is to use lexical resources which are constructed automatically from Web documents. Lexical resources are data of speech data, lexicons, text corpora, terminology, and various tools for language processing. In Japan, non-profit organization Gengo-Shigen-Kyokai (GSK) [13] is distributing these data. As for America and Europe, organizations called Linguistic Data Consortium (LDC) [14] and European Language Resources Association (ELRA) [15] are distributing lexical resources and they are aiming to promote the natural language processing technology.

In this paper, we propose a method that generates sentences by only using lexical resources constructed automatically. The proposed method can reduce the problem of the cost and coverage of the words.

Our proposed method uses two lexical resources: Kyoto University's case frame data [16] and Google N-gram [17]. Case frame is a data which describes what kind of noun is related to each predicate. The case frames were obtained from approximately 1.6 billion Japanese sentences extracted from Web. Google N-gram contains data of 1-7grams' frequency. They are extracted from approximately 20 billion Japanese sentences on Web. Our proposed method uses case frame to select appropriate words and case particles. Google N-gram is used to consider the co-occurence between words.

This paper is organized as follows. Section 2 explains the case frame data we employ. Section 3 details our proposed method. We show the experimental results to evaluate our method in Section 4 and conclude the paper in Section 5.

2. Kyoto University's Case Frame Data.

Kyoto University's case frame data is represented as a predicate and a set of its case filler words. For example, let us show a case frame of the Japanese verb "tsumu" (load/accumulate):

積む(*tsumu* : load) {従業員(*jyuugyouin* : employee), ドライバー(driver), …}が (*ga*)
{車(*kuruma* : car), トラック(truck), …}に(*ni*)
{荷物(*nimotsu* : baggage), 物資(*busshi* : supply)}を(*wo*)

where *"ga"*, *"wo"* and *"ni"* are Japanese case-marking postpositions, corresponding to nominative, accusative and dative, respectively. Such case frames have been utilized to improve not only fundamental analyses but also NLP applications such as information retrieval, automatic summarization and machine translation. To make practical use of case frames, wide-coverage case frames are required.

We use automatically constructed case frames [16] for our proposed method. The case frames were obtained from approximately 1.6 billion Japanese sentences extracted from Web. The database has about 40,000 predicates, 13 case frames on average for each predicate.

In Table 1, examples of resulting case frames of predicate "yaku" are shown. In this table, 'CS' means a case slot of Japanese postpositions. The number is the frequency of the noun in the case frame.

3. NLG Using Lexical Resources.

This section describes our proposed method of natural language generation. Figure 1 shows the flow of our proposed method. The method uses a verb v and nouns $\boldsymbol{w}$ as the input to generate sentences. Figure 2 shows an example of input words and the output. In this case, a sentence "*kissaten-de juice-wo*

TABLE 1. Examples of case frame

	CS	examples (noun:frequency)
yaku (1) (broil)	*ga*	I:18, person:15, craftsman:10, . . .
	wo	bread:2484, meat:1521, cake:1283, . . .
	ni	oven:1630, frying pan:1311, . . .
yaku (2) (have difficulty)	*ga*	teacher:3, government:3, person:3, . . .
	wo	fingers:2950, . . .
	ni	attack:18, action:15, son:15, . . .
yaku (2) (burn)	*ga*	maker:1, distributor:3, . . .
	wo	data:178, file:107, copy:9, . . .
	ni	R:1583, CD:664, CDR:3, . . .

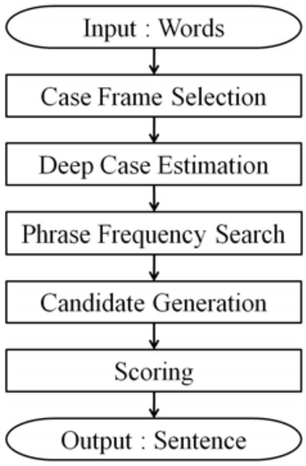

FIGURE 1. Flow of the system

Input : 喫茶店(*kissaten* : cafe), ジュース(juice), 飲む(*nomu* : drink)

Output : 喫茶店で(*kissaten-de* : at cafe) ジュースを(*juice-wo* : juice) 飲む(*nomu* : drink)。
Drink juice at cafe.

FIGURE 2. An example of output

nomu (Drink juice at cafe)" was generated from two nouns "*kissaten* (cafe), juice" and a verb "*nomu* (drink)".

In order to generate natural sentences from a verb and nouns, the following information is required.

- Selection of appropriate Japanese postpositions ("*ga*", "*wo*", etc.) for each noun.
- Set words in suitable order.
- Evaluation of correctness of the sentence.

Our method uses Kyoto University's case frame data [16] to select appropriate Japanese postposition, and Google N-gram to set words in order and evaluate each sentence's correctness.

The method first selects a case frame from Kyoto University's case frame data which includes all input words $(v, \boldsymbol{w})$. The selected case frame is used to generate the sentence. In the next step, the proposed method estimates the deep case for each example noun in the case frame, and searches the frequency of the noun using Google N-gram. This enables us to consider the co-occurence between nouns. The method generates the candidates of the output, and calculates the score for each candidate. One sentence is selected from the candidates as an output.

Each step is explained in detail from the next section.

3.1. **Case frame selection.** In this step, a case frame is selected to use in the proposed method.

First, we search for the case frame $x^{(k)}$ which has the same verb as the input v and includes all input nouns $\boldsymbol{w}$. We define the case slot in the case frame as $c_i^{(k)}$, and the noun in case slot $c_i^{(k)}$ as $z_{i,j}^{(k)}$. The frequency defined in the case frame for each noun is expressed as $f_{i,j}^{(k)}$.

In the next step, the sum of the frequency $S^{(k)}$ will be calculated for each case frame $x^{(k)}$ as follows:

$$S^{(k)} = \sum_i \sum_j f_{i,j}^{(k)} \quad (\text{if } z_{i,j}^{(k)} \text{ is included in } \boldsymbol{w}). \tag{1}$$

Finally, a case frame $x^{(k)}$ with maximum sum $S^{(k)}$ is selected to use in the next process.

We show an example of this step using the input as

$$(v = \{\textit{nomu} \text{ (drink)}\}, \boldsymbol{w} = \{\textit{kissaten} \text{ (cafe), juice}\}).$$

In the first step, two case frames $x^{(1)} = (\textit{nomu}\,(1))$, $x^{(2)} = (\textit{nomu}\,(2))$ in Table 2 are selected. By using Equation (1), each sum is calculated as $S^{(1)} = 1493$, $S^{(2)} = 147$. Case frame $x^{(1)}$ will be chosen for the next step to generate sentence.

TABLE 2. input ($v = \{\textit{nomu}$ (drink)$\}$, $\boldsymbol{w} = \{\textit{kissaten}$ (cafe), juice$\}$)

	CS	examples(noun:frequence)
nomu (1)	*wo*	juice:974, ...
	de	cafe:491, juice:10, ...
	ni	cafe:2, ...
	no	cafe:10, juice:6, ...
nomu (2)	*wo*	juice:135, ...
	de	cafe:12, ...

3.2. **Deep case estimation.** The technique to estimate the deep case is based on [18-20]. The deep case takes a different semantic role by the verb [21]. Table 3 shows the kinds of deep cases we employed in the proposed method.

Since Japanese uses case particles to underestimate the semantic role of the word, this approach matches each case particle with the candidate deep cases.

Deep case is estimated for each noun in the selected case frame. Table 5 shows an example of deep case estimation with case frame Table 4. By using this technique, even "*hitobito* (people)" in case slot "*ga*" will be estimated as an Agent role. In the case of "*kissaten* (cafe)" and "*izakaya* (bar)" in case slot "*de*", these two segments "*kissatenn-de* (at cafe)" and "*izakaya-de* (at bar)" will be estimated as a Location role.

3.3. **Phrase frequency search.** In this step, we use Google N-gram to search co-occurence frequency between nouns. The process will be carried out as follows.

Step 1: Express each noun by segment s by connecting noun with a case particle.
Step 2: Connect two segments s_m, s_n, and search for the frequency of phrase $s_m s_n$ with Google N-gram.
Step 3: Continue this step to all words in the case frame.

TABLE 3. Deep case and semantic role [18-20]

Deep Case	Role of Deep Case
Agent	Person or thing who is doing the event
Patient	The surface object of the verb
Instrument	Inanimate thing that an agent uses to implement the event
Location	Location or spatial orientation of a state or action
Goal	Place to which something moves or thing toward an action is directed
Source	The place of origin
Time	Temporal placement of an event
Cause	What caused the action
Modification	Other deep cases

TABLE 4. An example of case frame

	CS	Example(noun:frequency)
飲む(1) (*nomu*:drink)	*ga*	人々(*hitobito*:people):40, 子供(*kodomo*:child):72, …
	de	喫茶店(*kissaten*:cafe):491, ロック(rock):186, 居酒屋(*izakaya*:bar):166, コップ(cup):163, …
	wo	ウィスキー(whisky):1891, ジュース(juice):9746, …

TABLE 5. Estimation of deep case

	Deep case	Example(noun:frequency)
飲む(1) (*nomu*:drink)	Agent	人々が(*hitobito-ga*:people):40, 子供が(*kodomo-ga*:child):72, …
	Location	居酒屋で(*izakaya-de*:at bar):166, 喫茶店で(*kissaten-de*:at cafe):491,…
	Instrument	ロックで(*rock-de*:on the rock):187, コップで(*cup-de*:by cup):163, …
	Patient	ウィスキーを(*whisky-wo*:whisky):1891, ジュースを(*juice-wo*:juice):9746, …

An example is shown with the case frame "*nomu* (1) (drink)" in Table 4. First, each noun is expressed as a segment s. Noun "*hitobito* (people)" is expressed as a segment "*hitobito-ga*", by connecting the case particle to the end of each noun. In the next step, frequency is searched for each phrase $s_m s_n$. For example, frequency of two phrases as shown below is searched in Google N-gram.

ロックでウィスキーを(*rock-de whisky-wo* : (drink) whisky on the rock): 20
ロックでジュースを (*rock-de juice-wo* : (drink) juice on the rock) : 0

From this result, it is obtained that the segment "*rock-de* (by rock)" is likely to co-occur with segment "*whisky-wo* (whisky)", rather than with segment "*juice-wo* (juice)".

The result of frequency search with case frame *nomu* (1) (drink) in Table 4 is shown in Figure 3. In this figure, frequency is expressed with directed line segment. If two

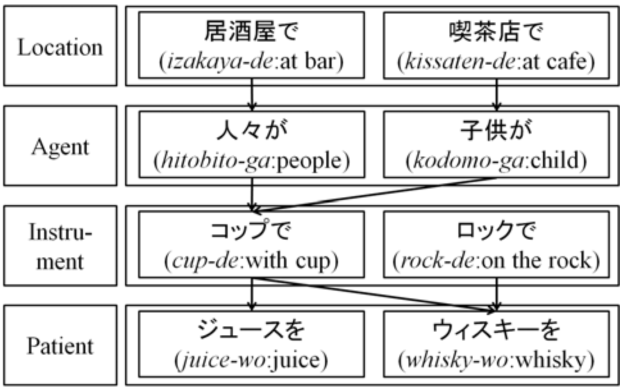

FIGURE 3. Result of frequency with case frame *nomu* (1) (drink)

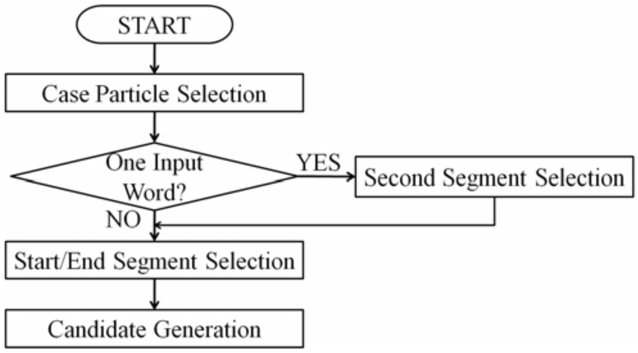

FIGURE 4. Flow of candidate sentence generation

segments s_m, s_n are connected with a line segment, this means that the phrase $s_m s_n$ has more than 0 frequency in Google N-gram.

From the next process, the method mainly uses this case frame with frequency information to generate sentence.

3.4. **Candidate generation.**

3.4.1. *Overview.* This section illustrates the process to generate candidate sentences. Figure 4 shows the flow of this step. First the case particle for each input word is selected. The selected case particle is used in the sentence generation. Then, the method selects two segments among the input words. The segments are used as the start and end segment in the sentence. Finally, it generates the candidate sentences.

3.4.2. *Case particle selection.* At first case particle to use for each input word w_i is selected. The selection is based on the frequency in the case frame database. We define this frequency of noun w_j in case slot c_i as $f_{i,j}^{(k)}$. This frequency means how important the segment is, in the case frame. Therefore, the case particle c_i with maximum frequency for each noun is selected.

We show an example of this step with case frame *nomu* (1) (drink) and input words as $\boldsymbol{w} = \{kissaten$ (cafe), juice$\}$. In this case, the noun "*kissaten* (cafe)" is included in three case slots "*ni, de, no*". However, the case slot with maximum frequency is "*de*". Therefore, case particle "*de*" is chosen for the noun "*kissaten* (cafe)". As for the noun "juice", case particle "*wo*" is selected.

3.4.3. *Second segment selection.* If there is only one word in the input $\boldsymbol{w}$, the second segment is selected to use in the sentence. The selected segment has to be a different deep case with the input $\boldsymbol{w}$ and the co-occurence must be high.

In order to find second segments, co-occurence score L is calculated for each segment in the case frame. This score L_{s_m,s_n} of two segments s_m, s_n are calculated as:

$$L_{s_m,s_n} = \max\left\{\frac{F_{s_m s_n}}{F_{s_m}+F_{s_n}}, \frac{F_{s_n s_m}}{F_{s_m}+F_{s_n}}\right\}. \tag{2}$$

where $F_{s_m s_n}$ is the frequency of phrase $s_m s_n$ searched in Google N-gram.

Co-occurence score L between input segment s_w and all other segments in case frame is calculated. The segment with highest co-occurence score is selected as the second segment.

Table 7 shows an example of score L with the input $\boldsymbol{w} = \{kissaten$ (cafe)$\}$. In this case, the co-occurence score with "*juice-wo* (juice)" is the higher than with "*beer-wo* (beer)". Therefore, segment "*juice-wo* (juice)" is selected as the second segment to use with "*kissaten-de* (at cafe)" in sentence generation.

3.4.4. *Start/End segment selection.* This step selects two segments from the input as Start/End segment of the sentence to generate. The selection is carried out in the following rules.

- If there was Patient (deep case) segment, select it as End.
- If there was Agent (deep case) segment, select it as Start.
- Other than that, choose Start/End segment with higher frequency f_s.

TABLE 6. Example of case frame *nomu*(1) $\boldsymbol{w}$ ={cafe, juice}

	CS	example(noun:frequency)
飲む(1) (*nomu*:drink)	に(*ni*)	喫茶店(*kissaten*:cafe):2, …
	で(*de*)	喫茶店(*kissaten*:cafe):491, …
	を(*wo*)	ジュース(juice):974, …
	の(*no*)	喫茶店(*kissaten*:cafe):10, ジュース(juice):6, …

TABLE 7. An example of co-occurence $\boldsymbol{w}$ ={cafe}

Segment *s*	F_s
喫茶店で (*kissaten-de*:at cafe)	379,540
ジュースを (*juice-wo*:juice)	705,689
ビールを (*beer-wo*:beer)	1,678,505

Segment *A*	Segment *B*	F_{AB}	L[1/1,000]
喫茶店で (*kissaten-de*:at cafe)	ジュースを (*juice-wo*:juice)	151	0.139
ジュースを (*juice-wo*:juice)	喫茶店で (*kissaten-de*:at cafe)	0	
喫茶店で (*kissaten-de*:at cafe)	ビールを (*beer-wo*:beer)	192	0.0933
ビールを (*beer-wo*:beer)	喫茶店で (*kissaten-de*:at cafe)	0	

The order of the words in Japanese is not strict in general. For example, two sentences with different order as below have the same meaning. However, it is said that Agent

喫茶店で(*kissatenn-de*)ジュースを(*juice-wo*)飲む(*nomu*)。
ジュースを(*juice-wo*)喫茶店で(*kissatenn-de*)飲む(*nomu*)。
Drink juice at cafe.

segment is likely to come to the head of the sentence and Patient segment to be near the verb at the end of the sentence [22-24]. From the grammatical knowledge, we adopted a rule to select Agent and Patient segment as the Start/End segment.

Table 8 shows some examples of the selection of Start/End segment using case frame shown in Table 4.

TABLE 8. Examples of start/end segment

	Segment		Begin	End
(1)	子供が (*kodomo-ga* : child)	ジュースを (*juice-wo* : juice)	子供が (*kodomo-ga* : child)	ジュースを (*juice-wo* : juice)
(2)	子供が (*kodomo-ga* : child)	喫茶店で (*kissaten-de* : at cafe)	子供が (*kodomo-ga* : child)	喫茶店で (*kissaten-de* : at cafe)
(3)	コップで (*cup-de* : by cup)	喫茶店で (*kissaten-de* : at cafe)	コップで (*cup-de* : by cup)	喫茶店で (*kissaten-de* : at cafe)

(1): "*kodomo-ga* (child)" is an Agent and "*juice-wo* (juice)" is a Patient. Therefore, each segment is selected as Start/End segment.
(2): "*kodomo-ga* (child)" is selected as Start segment, and the other segment "*kissaten-de* (at cafe)" is End segment.
(3): Neither "*cup-de* (by cup)" nor "*kissaten-de* (at cafe)" is Agent/Patient. Each frequency in the case frame is $f_{cup\text{-}de} = 163$, $f_{kissaten\text{-}de} = 491$, so the segment with higher frequency "*cup-de* (by cup)" is selected as the End, and the other "*kissaten-de* (at cafe)" is selected as the Start segment.

3.4.5. *Candidate generation.* In this step, the method generates sentences from case frame with frequency (Figure 3). The candidates require to satisfy the condition below:
(1): Start with the Start segment s_A.
(2): End with the End segment s_B.
(3): Use all input segments $s_1, s_2, \cdots$.
(4): The number of the segment that constructs the sentence is less than maximum $l_{\max}$.
(5): There is no segment with same deep case.
(6): There are co-occurence frequencies between two continuing segments.

Start/End segments s_A, s_B in (1) and (2) are selected in the previous section. The input segment $s_1, s_2, \cdots$ in (3) is selected in Section 3.4.2. We define the condition (4) so as not to generate sentences too complicated. (5) comes from the definition of Fillmore's Case Grammar theory. (6) means there are more than 0 frequency in Section 3.3.

For example, we illustrate the generated sentence with the case frame *nomu* (1) (drink) in Figure 3. When the input words are $\boldsymbol{w} = \{$*kissaten* (cafe), juice)$\}$, and the selected Start/End segments are $s_A =$ *kissaten-de* (at cafe), $s_B =$ *juice-wo* (juice), the generated candidates are as follows:

- "*kissaten-de juice-wo nomu* (Drink juice at cafe)" ($l_{\max} = 3$) (Figure 5)

- "*kissaten-de kodomo-ga juice-wo nomu* (A child drinks juice at cafe)" ($l_{\max} = 4$) (Figure 6)
- "*kissaten-de kodomo-ga cup-de juice-wo nomu* (A child drinks juice with cup at cafe)" ($l_{\max}$) (Figure 7)

3.5. **Scoring.** In the previous section, it is likely that the sentences with low co-occurence frequeny are to be the candidate. Therefore, a score for each candidate sentence is calculated in this step.

FIGURE 5. "*kissaten-de juice-wo nomu* (Drink juice at cafe)"

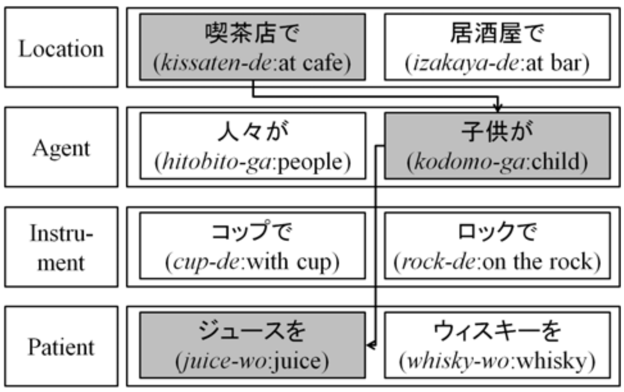

FIGURE 6. "*kissaten-de kodomo-ga juice-wo nomu* (A child drinks juice at cafe)"

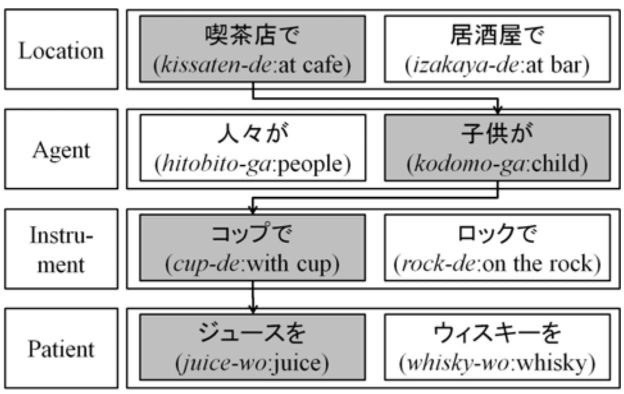

FIGURE 7. "*kissaten-de kodomo-ga cup-de juice-wo nomu* (A child drinks juice with cup at cafe)"

We define candidate sentence C_i and the score S_i. As shown in Figure 8, when the candidate sentence C_i consisted of segments $s_1, s_2, \cdots, s_n$, score S_i is calculated as follow:

$$S_i = \sum_{j=1}^{n-1} \frac{F_{s_j s_{j+1}}}{F_{s_j} + F_{s_{j+1}}}. \tag{3}$$

As shown in Figure 8, $F_{s_j s_{j+1}}$ is the frequency from Google N-gram of the phrase of two continuing segments s_j, s_{j+1}. F_{s_j} is the frequency from Google N-gram of the segment s_j.

3.6. **Examples of output.** Tables 9 and 10 show the examples of the ouput sentences. **segment max *m*** in the tables is the parameter of the sentence length in Section 3.4.5. Table 9 shows the output when the input is

$$(v = tobu \text{ (fly)}, \boldsymbol{w} = \{hakuchou \text{ (swan)}\}).$$

In this case, the second segment other than the input "*hakuchou*(swan)" is selected to generate sentence (Section 3.4.3). "*jyoukuu-wo* (in the sky)" is selected, and by changing the sentence length parameter n, four sentences are generated by the method.

Table 10 shows the output when the input was

$$(v = yaku \text{ (grill)}, \boldsymbol{w} = \{sumibi \text{ (charcoal)}, sakana \text{ (fish)}\}).$$

Four sentences, according to the length, have been generated using one verb and two nouns as the input.

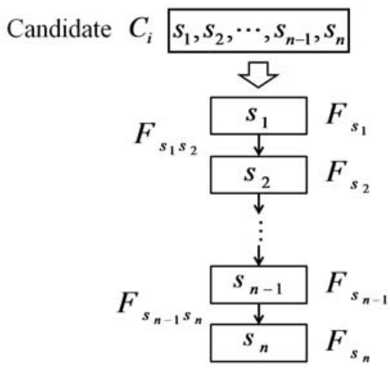

Figure 8. Candidate centence C_i

Table 9. input ($v = tobu$ (fly), $\boldsymbol{w} = \{hakuchou$ (swan)$\}$)

l_{max}	Output
3	白鳥が 上空を 飛ぶ (*hakuchou-ga jyoukuu-wo tobu* : A swan is flying in the sky.)
4	白鳥が 池の 上空を 飛ぶ (*hakuchou-ga ike-no jyoukuu-wo tobu* : A swan is flying over the pond in the sky.)
5	白鳥が 私の ように 上空を 飛ぶ (*hakuchou-ga watashi-no youni jyoukuu-wo tobu* : A swan is flying like me in the sky.)
6	白鳥が 私の ように へりで 上空を 飛ぶ (*hakuchou-ga watshi-no youni heri-de jyoukuu-wo tobu* : A swan is flying like me in the sky with helicopter)

TABLE 10. input (v = *yaku* (grill), $\boldsymbol{w}$ = {*sumibi* (charcoal), *sakana* (fish)})

l_{max}	Output
3	炭火で 魚を 焼く (*sumibi-de sakana-wo yaku* : Fish is grilled over charcoal.)
4	炭火で 旬の 魚を 焼く (*sumibi-de shunn-no sakana-wo yaku* : Seasonal fish is grilled over charcoal.)
5	炭火で 目の 前で 旬の 魚を 焼く (*sumibi-de me-no mae-de shunn-no sakana-wo yaku* : Seasonal fish is grilled in front of me over charcoal.)
6	炭火で 目の 前で 皆 様に 魚を 焼く (*sumibi-de me-no mae-de mina sama-ni sakana-wo yaku* : Fish is grilled for everyone in front of me over charcoal.)

リボンを 結んだ 風船を 空に 飛ばした。
(*ribbon-wo musunnda huusenn-wo sora-ni tobashita* :
Released a balloon with a ribbon tied to the sky.)

FIGURE 9. Sentence in text book

TABLE 11. Examples of dataset

verb (v)	noun (w)
結ぶ (*musubu* : tie)	リボン(ribbon)
飛ばす (*tobasu* : release)	風船(*huusenn* : balloon), 空(*sora* : sky)

TABLE 12. Extracted datasets ($v, \boldsymbol{w}$)

number of extracted noun w	1	2
Total	274	130

4. **Experiment.** We carried out experiments to evaluate the quality of the generated sentences.

4.1. **Datasets.** The input data for the experiment were extracted from Japanese text books for elementary school students. We used Japanese syntactic parser CaboCha [25] for each sentence in the text book. From the result of parsing, we obtained datasets of a verb and nouns related to the verb.

Figure 9 shows an example of the sentence in the Japanese text books, and Table 11 shows the examples of the extracted dataset from that sentence. In this case, two datasets for input have been obtained.

We executed this process on 430 sentences in the Japanese text books. Table 12 shows the total for each number of noun. These datasets were used for the experiment.

4.2. **Condition.** The conditions of the experiment were as below:

- Used dataset as shown in Table 12 for input.
- The maximum length of the sentence n was changed among $3, \cdots, 6$. This means four kinds of sentences can be generated from one dataset (Examples in Table 9 and Table 10).
- Evaluated by 18 Japanese subjects.
- Evaluated 160 generated sentences randomly selected (20 for each number of extracted noun $\boldsymbol{w} = \{1, 2\}$ and maximum length $l_{\max} = \{3, \cdots, 6\}$).

- Evaluation Item
  1. Understandable.
  2. Commonsensical.
  3. Grammatical.

In order to evaluate how robust the method acts to the variation of the parameter, we compare evaluation result for each parameters as follow.

- sentence length $l_{\max} = \{3, \cdots, 6\}$
- the number of input words $\boldsymbol{w} = \{1, 2\}$

Table 13 shows the example for each evaluation item.

TABLE 13. Evaluation example

Output	A	B	C		
			C-1	C-2	C-3
写真の内容を出来る (*shashinn-no naiyou-wo dekiru* : Picture's matter is available.)	×	×	×	×	×
車が自転車に乗る (*kuruma-ga jitennsha-ni noru* : Car is on the bicycle.)	○	×	○	○	○
家族で旅行で行く (*kazoku-de ryokou-de iku* : Go to a trip with family.)	○	○	×	○	○
沿いの道を歩く (*zoi-no michi-wo aruku* : Walk the street on the banks of.)	○	○	○	×	○
白鳥が湖の飛ぶ (*hakuchou-ga mizuumi-no tobu* : Swan lake's fly.)	○	○	○	○	×
大輪の花火が上がる (*tairinn-no hanabi-ga agaru* : Large fireworks is shot off.)	○	○	○	○	○

4.2.1. *A: Understandable.* We asked the subjects to judge whether or not they could understand the meaning of the sentence. In the example of Table 13, "*shashinn-no naiyou-wo dekiru* (Picture's matter is available)" was judged to be hard to understand. We asked not to estimate the grammar and the common sense of the sentence in this entry. This means that if the sentence was understandable with different word order, the subjects should judge the sentence understandable. For example, the sentence "*hakuchou-ga mizuumi-no tobu* (Swan lake's fly.)" can be understandable if the order was "*mizuumi-no hakuchou-ga tobu* (Swan by the lake flies.)". Therefore, we asked the subjects to evaluate these sentences as understandable.

4.2.2. *B: Commonsensical.* This entry evaluated whether the meaning of the sentence was commonsensical. The second example in Table 13 "*kuruma-ga jitennsha-ni noru* (Car is on the bicycle)", may be judged to be understandable but the meaning was unlikely from a commonsense point of view.

4.2.3. *C: Grammatical.* As for the grammatical evaluation entry, we asked the subjects to check the three items as follows:

C-1: Case particle is natural
C-2: Used words are natural
C-3: Word order is natural

C-1 checked the case particle for each noun. The third example in Table 13 was judged to be unnatural because the same case particle "*de*" was used.

C-2 evaluated whether the words used in the sentence were natural. The fourth example "*zoi-no michi-wo aruku* (Walk the street on the banks of.)" lacked the information of "on the banks of (where)". Therefore, this item was judged to be unsuitable.

C-3 was the entry to check the word order. The fifth example "*hakuchou-ga mizuumi-no tobu* (Swan lake's fly)" was understandable if the order was "*mizuumi-no hakuchou-ga tobu* (Swan in the lake flies.)". Due to this, this sentence was evaluated as unsuitable.

4.3. **Evaluation result.** When we defined "Perfect Sentence" to be evaluated all five entries to be suitable, the result of the prefect sentence generated from the proposed method is summarized in Table 14. The result rate shows that more than 40% of the sentences generated are perfectly natural sentences, in terms of grammar and the meaning.

Figrue 10 shows the rate of perfect sentence by the length of the sentence n. From these results, it is observed that the shorter the generated sentence is, the more natural sentence is likely to be generated. Also, from Table 14 and Figure 10, it is observed that the more nouns we use as the input, the more perfect the sentence is likely to be.

Table 15 illustrates the result rate for each entry. More than 80% of the generated sentences are understandable.

Figure 11 and Figure 12 are the result rates by sentence length n. Figure 11 is the result of the generated sentence from two input nouns $\boldsymbol{w}$. Figure 11 is the result of the generated sentence from one input noun $\boldsymbol{w}$. From these results, we can see that the entry of C-2 (Used word) changes the most when the sentence is longer. Comparing between the number of input nouns, the result with two inputs is higher in all entries. This tendency is observed especially in the entry C-3 (Word order). The difference was caused by the selection of second segment (Section 3.4.3).

5. **Conclusions.** As a novel approach for natural language generation, we proposed a method to generate sentence using only automatically constructed lexical resources, which has no cost to construct manually, and the coverage of the word data is very high. We used two lexical resources: Kyoto University's case frame and Google N-gram. Case

TABLE 14. Rate of perfect sentence

Number of input noun $\boldsymbol{w}$	Rate of Perfect Sentence [%]
2	48.9
1	41.2

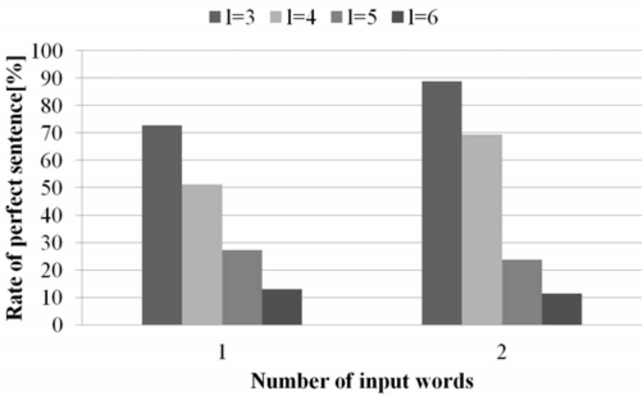

FIGURE 10. Rate of perfect sentence by length l

TABLE 15. Rate for each evaluation entry

Number of input noun $\boldsymbol{w}$	A [%]	B [%]	C [%]		
			C-1	C-2	C-3
2	89.7	81.9	73.9	65.8	80.8
1	80.6	69.4	69.5	58.9	63.4

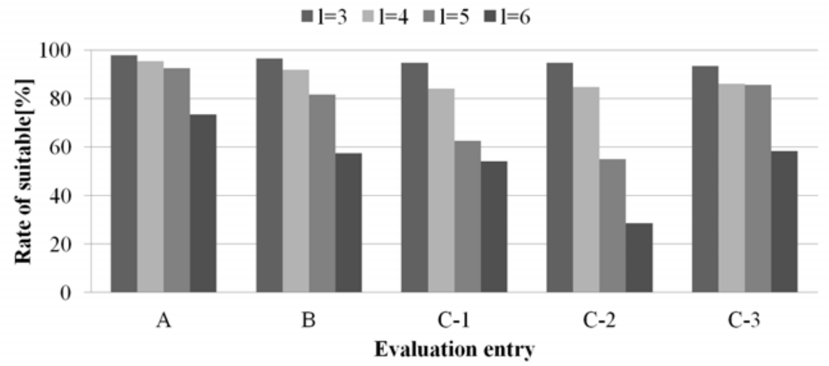

FIGURE 11. Result rate by sentence length l ("number of input noun $\boldsymbol{w}$" = 2)

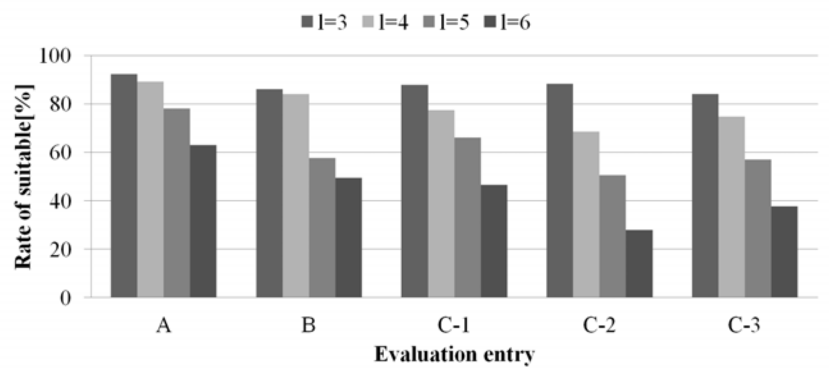

FIGURE 12. Result rate by sentence length l ("number of input noun $\boldsymbol{w}$" = 1)

frame enabled us to generate a sentence that has meanings, and Google N-gram enabled to consider co-occurence between words.

The experimental results show that the method can generate natural sentences even it is only using lexical resources that are automatically constructed. About 40% are perfect sentences, and more than 80% are understandable.

In our future work, we plan to apply the proposed method to a conversational system.

REFERENCES

[1] K. R. McKeown, Language generation: Applications, issues, and approaches, *Proc. of the IEEE*, vol.74, no.7, pp.961-968, 1986.
[2] A. H. Oh and A. I. Rudnicky, Stochastic natural language generation for spoken dialog systems, *Computer Speech & Language*, vol.16, no.3-4, pp.387-407, 2002.
[3] M. Elhadad and J. Robin, An overview of SURGE: A reusable comprehensive syntactic realization component, *Technical Report*, Department of Mathematics and Computer Science, 1996.
[4] H. Inozuka, M. Hosoi, Y. Aramomi and K. Murata, Sentence generation system by IPAL (SURFACE/DEEP), *IPSJ Technical Report, NL*, vol.99, no.22, pp.113-119, 1999.
[5] J.-H. Hong, S. Lim and S.-B. Cho, Autonomous language development using dialogue-act templates and genetic programming, *IEEE Trans. Evolutionary Computation*, vol.11, no.2, pp.213-225, 2007.

[6] G. Ferguson and J. F. Allen, TRIPS: An integrated intelligent problem-solving assistant, *Proc. of the 16th National Conference on Artificial Intelligence (AAAI-98)*, pp.567-572, 1998.
[7] V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen and L. Hetherington, JUPlTER: A telephone-based conversational interface for weather information, *IEEE Transactions on Speech and Audio Processing*, vol.8, no.1, pp.85-96, 2000.
[8] Y. Yagi, S. Takada, K. Hirose and N. Minematsu, Realization of concept-to-speech conversion for reply speech generation in a spoken dialogue system of road guidance and its evalution (speech processing), *IPSJ*, vol.48, no.9, pp.3300-3308, 2007.
[9] M. Shibata, Y. Tomiura and T. Nishiguchi, Method for selecting appropriate sentence from documents on the www for the open-ended conversation dialog system, *Transactions of the Japanese Society for Artificial Intelligence*, vol.24, no.6, pp.507-519, 2009.
[10] M. Sasayama, F. Ren and S. Kuroiwa, Automatic super-function extraction for translation of spoken dialogue, *International Journal of Innovative Computing Information and Control*, vol.4, no.6, pp.1371-1381, 2008.
[11] W. Bosma, E. Marsi, E. Krahmer and M. Theune, Text-to-text generation for question answering, in *Interactive Multi-modal Question-Answering*, A. Bosch and G. Bouma (eds.), Springer Berlin Heidelberg, 2011.
[12] Y. Yang, P. Jiang, S. Tsuchiya and F. Ren, Effect of using pragmatics information on question answering system of analects of confucius, *International Journal of Innovative Computing Information and Control*, vol.5, no.5, pp.1201-1212, 2009.
[13] *GSK Homepage*, http://www.gsk.or.jp/index_e.html.
[14] *LDC – Linguistic Data Consortium*, http://www.ldc.upenn.edu/.
[15] *ELRA European Language Resources Association*, http://www.elra.info/.
[16] D. Kawahara and S. Kurohashi, Case frame compilation from the web using high-performance computing, *Proc. of the 5th International Conference on Language Resources and Evaluation*, pp.1344-1347, 2006.
[17] T. Kudo and H. Kazawa, *Web Japanese N-gram Version 1*, 2007.
[18] T. Watanabe, M. Ohta, K. Ohta and H. Ishikawa, A page fusion method using case grammar, *DBSJ Letters*, vol.3, no.2, pp.97-100, 2004.
[19] T. Watanabe, S. Ohno, M. Ohta, K. Katayama and H. Ishikawa, *A Distinction Emphasis Multi-document Fusion Technique*, 2005.
[20] T. Oishi and Y. Matsumoto, Lexical knowledge acquisition for japanese verbs based on surface case pattern analysis, *IPSJ*, vol.36, no.11, pp.2597-2610, 1995.
[21] S. Amano, S. Ishizaki, T. Utsuro, M. Narita and J. Fukumoto, *Natural Language Processing*, Ohmsha, Tokyo, 2007.
[22] K. Nakamoto, L. Jae-ho and K. Kow, Preferred word orders correlate with sentential meanings that cannot be reduced to verb meanings: A new perspective on construction effects in japanese, *Cognitive Studies*, vol.13, no.3, pp.334-352, 2006.
[23] K. Tamaoka, H. Sakai, J. Kawahara, Y. Miyaoka, H. Lim and M. Koizumi, Priority information used for the processing of Japanese sentences: Thematic roles, case particles or grammatical functions? *Journal of Psycholinguistic Research*, vol.34, no.3, pp.281-332, 2005.
[24] D. Yasunaga, S. Muraoka and T. Sakamoto, Case marker effects of japanese in predicting following element, *Cognitive Studies*, vol.17, no.3, pp.663-669, 2010.
[25] T. Kudo and Y. Matsumoto, Japanese dependency analysis using cascaded chunking, *CoNLL: Proc. of the 6th Conference on Natural Language Learning (COLING Post-Conference Workshops)*, pp.63-69, 2002.