Exploiting Lexical Conceptual Structure for Paraphrase Generation

Atsushi Fujita[1], Kentaro Inui[2], and Yuji Matsumoto[2]

[1] Graduate School of Informatics, Kyoto University
fujita@pine.kuee.kyoto-u.ac.jp
[2] Graduate School of Information Science, Nara Institute of Science and Technology
{inui,matsu}@is.naist.jp

Abstract. Lexical Conceptual Structure (LCS) represents verbs as semantic structures with a limited number of semantic predicates. This paper attempts to exploit how LCS can be used to explain the regularities underlying lexical and syntactic paraphrases, such as verb alternation, compound word decomposition, and lexical derivation. We propose a paraphrase generation model which transforms LCSs of verbs, and then conduct an empirical experiment taking the paraphrasing of Japanese light-verb constructions as an example. Experimental results justify that syntactic and semantic properties of verbs encoded in LCS are useful to semantically constrain the syntactic transformation in paraphrase generation.

1 Introduction

Automatic paraphrasing has recently been attracting increasing attention due to its potential in a broad range of natural language processing tasks. For example, a system that is capable of simplifying a given text, or showing the user several alternative expressions conveying the same content, would be useful for assisting a reader.

There are several classes of paraphrase that exhibit a degree of regularity. For example, paraphrasing associated with verb alternation, lexical derivation, compound word decomposition, and paraphrasing of light-verb constructions (LVC(s)) all fall into such classes. Examples[1] (1) and (2) appear to exhibit the same transformation pattern, in which a compound noun is transformed into a verb phrase. Likewise, paraphrases involving an LVC as in (3) and (4) (from [4]) have considerable similarities.

(1) s. **His machine operation** is very **good**.
t. **He operates the machine** very **well**.

(2) s. My **son's bat control** is **unskillful** yet.
t. My **son controls his bat poorly** yet.

(3) s. Steven **made an attempt** to stop playing.
t. Steven **attempted** to stop playing.

(4) s. It **had a noticeable effect** on the trade.
t. It **noticeably affected** the trade.

[1] For each example, "s" and "t" denote an original sentence and its paraphrase, respectively. Note that our target language is Japanese. English examples are used for an explanatory purpose.

However, the regularity we find in these examples is not so simple that it cannot be captured only in syntactic terms. For example, the transformation pattern as in (1) and (2) does not apply to another compound noun "machine translation." We can also find a range of varieties in paraphrasing of LVCs as we describe in Section 3.

In spite of this complexity, the regularity each paraphrase class exhibits were explained by recent advances in lexical semantics, such as the Lexical Conceptual Structure (LCS) [8] and the Generative Lexicon [17]. According to the LCS, for instance, a wide variety of paraphrases including word association within compounds, transitivity alternation, and lexical derivation, were explained by means of the syntactic and semantic properties of the verb involved. The systematicity underlying such linguistic accounts is intriguing also from the engineering viewpoint as it could enable us to take a more theoretically motivated but still practical approach to paraphrase generation.

The issue we address in this paper is to empirically clarify (i) what types of regularities underlying paraphrases can be explained by means of lexical semantics and how, and (ii) how lexical semantics theories can be enhanced with feedback from practical use, namely, paraphrase generation. We make an attempt to exploit the LCS among several lexical semantics frameworks, and propose a paraphrase generation model which utilizes LCS combining with syntactic transformation.

2 Lexical Conceptual Structure

2.1 Basic framework

Among several frameworks of lexical semantics, we focus on the Lexical Conceptual Structure (LCS) [8] due to the following reasons. First, several studies [9, 3, 19] have shown that the theory of the LCS provides a systematic explanation of semantic decomposition as well as syntax determines. In particular, Kageyama [9] has shown that even a simple typology of LCS can explain a wide variety of linguistic phenomena including word association within compounds, transitivity alternation, and lexical derivation. Second, large-scale LCS dictionaries have been developed through practical use on machine translation and compound noun analysis [3, 19]. The LCS dictionary for English [3] (4,163-verbs with 468 LCS types) was tailored based on a verb classification [12] with an expansion for the semantic role delivered to arguments. For Japanese, Takeuchi *et al.* [19] developed a 1,210-verbs LCS dictionary (with 12 LCS types) called the T-LCS dictionary, following Kageyama's analysis [9]. In this paper, we make use of the current version of the T-LCS dictionary, because it provides a set of concrete rules for LCS assignment, which ensures the reliability of the dictionary.

Examples of LCS in the T-LCS dictionary are shown in Table 1. An LCS consists of a combination of semantic predicates ("CONTROL," "BE AT," etc.) and their argument slots (x, y, and z). Each argument slot corresponds to a semantic role, such as "Agent," "Theme," and "Goal," depending on its surrounding semantic predicates. Let us take "*yakusu* (to translate)" as an example. The inner structure "[y BE AT z]" denotes the state of affairs where z ("Goal") indicates the state or physical location of y ("Theme"). The predicate "BECOME" expresses a change of y. In the case of example phrase in Table 1, the change of the language of the book is represented. The leftmost

Table 1. Examples of LCS

LCS for verb (example verb) example Japanese phrase
[y BE AT z] (*ichi-suru* (to locate), *sonzai-suru* (to exist)) *gakkou-ga* *kawa-no* *chikaku-ni* *ichi-suru.* school-NOM river-GEN near-DAT to locate-PRES The school (Theme) locates near the river (Goal).
[BECOME [y BE AT z]] (*houwa-suru* (to become saturate), *bunpu-suru* (to be distributed)) *kono-hana-ga* *sekaiju-ni* *bunpu-suru.* this flower-NOM all over the world-DAT to distribute-PRES This flower (Theme) is distributed all over the world (Goal).
[x CONTROL [BECOME [y BE AT z]]] (*yakusu* (to translate), *shoukai-suru* (to introduce)) *kare-ga* *hon-o* *nihongo-ni* *yakusu.* he-NOM book-ACC Japanese-DAT to translate-PRES He (Agent) translates the book (Theme) into Japanese (Goal).
[x ACT ON y] (*unten-suru* (to drive), *sousa-suru* (to operate)) *kare-ga* *kikai-o* *sousa-suru.* he-NOM machine-ACC to operate-PRES He (Agent) operates the machine (Theme).
[y MOVE TO z] (*ido-suru* (to move), *sen'i-suru* (to propagate)) *ane-ga* *tonarimachi-ni* *ido-suru.* my sister-NOM neighboring town-DAT to move-PRES My sister (Theme) moves to a neighboring town (Goal).

part "[x CONTROL . . .]" denotes that the "Agent" causes the state change. The difference between "BECOME BE AT" and "MOVE TO" is underlying their telicity: the former indicates telic, and thus the verb can be perfective, while the latter atelic. Likewise, "CONTROL" implicates a state change, while "ACT ON" merely denotes an action. The following are examples of syntactic and semantic properties represented in LCS:

- Semantic role of argument (e.g. "[x CONTROL . . .]" indicates x="Agent")
- Syntactic case particle pattern (e.g. "[y MOVE TO z]" indicates y=NOM, z=DAT)
- Aspectual property (e.g. "MOVE TO" is atelic ("**ket-tearu* (to kick-PERF)"), while "BECOME BE AT" is telic ("*oi-tearu* (to place-PERF).").)
- Focus of statement
  (e.g. x is focused in "[x CONTROL . . .]", while z in "[z BE WITH . . .]")
- Semantic relations in lexical derivation
  - transitivity alternation ("*kowasu* (to break (vt))" $\Leftrightarrow$ "*kowareru* (to break (vi))")
  - lexical active-passive alternation ("*oshieru* (to teach)" $\Leftrightarrow$ "*osowaru* (to be taught)")

2.2 Disambiguation in LCS analysis

In principle, a verb is associated with more than one LCS if it has multiple senses. The mapping from syntactic case assignments to argument slots in LCS is also many-

to-many in general. In the case of Japanese, the case particle "*ni*" tends to be highly ambiguous as demonstrated in (5).

(5) a. *shuushin-jikan-o* ***yoru-11ji-ni*** *henkou-shita.*
bedtime-ACC 11 p.m.-DAT (complement) to change-PAST
I changed my bedtime to 11 p.m.

b. ***yoru-11ji-ni*** ***yuujin-ni*** *mail-o* *okut-ta.*
11 p.m.-DAT (adjunct) friends-DAT (complement) mail-ACC to send-PAST
I sent a mail to my friends at 11 p.m.

Resolution of these sorts of ambiguity is called semantic parsing and has been actively studied by many researchers recently [6, 2] as semantically annotated corpora and lexical resources such as the FrameNet [1] and the Proposition Bank [16] have become available. Relying on the promising results of this trend of research, we do not address the issue of semantic parsing in this paper to focus our attention on the generation side of the whole problem.

3 Paraphrasing of light-verb constructions

In this paper, we focus our discussion on one class of paraphrases, i.e., paraphrasing of light-verb constructions (LVCs). Sentence (6s) shows an example of an LVC. An LVC is a verb phrase ("*kandou-o atae-ta* (made an impression)," c.f., Figure 1) that consists of a light-verb ("*atae-ta* (to give-PAST)") that syntactically governs a deverbal noun ("*kandou* (an impression)"). A paraphrase of (6s) is shown in sentence (6t), where the deverbal noun functions as the main verb with its verbalized form ("*kandou-s-**ase**-ta* (to be impressed-CAUSATIVE-PAST)").

(6) s. *eiga-ga* *kare-**ni*** *saikou-**no*** ***kandou-o*** ***atae**-ta.*
film-NOM him-DAT supreme-GEN impression-ACC to give-PAST
The film made an supreme impression on him.

t. *eiga-ga* *kare-**o*** *saikou-**ni*** ***kandou-s-ase**-ta.*
film-NOM him-ACC supreme-DAT to be impressed-CAUSATIVE-PAST
The film supremely impressed him.

Example (6) indicates that we need an information to determine how the voice of target sentence must be changed and how the case particles of the nominal elements must be reassigned. These decisions depend not only on the syntactic and semantic attributes of the light-verb, but also on those of the deverbal noun [14]. LVC paraphrasing is thus a novel challenging material for exploiting LCS.

Figure 1 demonstrates tree representations of source and target expressions involved in LVC paraphrasing, taking (6) as an example. To generate this type of paraphrase, we need a computational model that is capable of the following operations:

Change of the dependence: Change the dependences of the elements (a) and (b) due to the elimination of the original modifiee, the light-verb. This operation can be done by just making them dependent on the resultant verb.

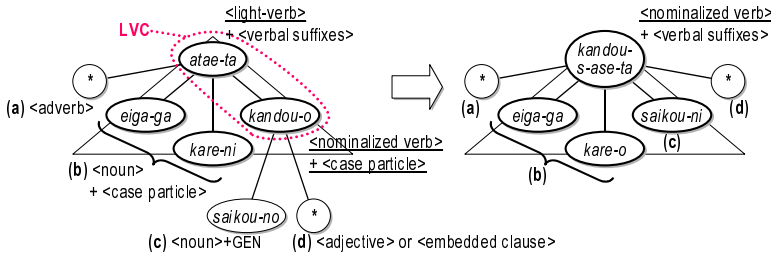

Fig. 1. Dependency structure showing the range which the LVC paraphrasing affects. The oval objects denote Japanese base-chunks so-called *bunsetsu*.

Re-conjugation: Change the conjugation form of the elements (d) and occasionally (c), according to the syntactic category change of their modifiee: the given deverbal noun is verbalized. This operation can be carried out independently of the LVC paraphrasing.

Selection of the voice: Choose the voice of the target sentence among active, passive, causative, etc. In example (6), the causative (the auxiliary verb "*ase*") is chosen. The decision depends on the syntactic and semantic attributes of both the given light-verb and the deverbal noun [14].

Reassignment of the cases: Assign the case particles of the elements (b) and (c), the arguments of the main verb. In (6), the syntactic case of "*kare* (him)," which was originally assigned dative case "*ni*" is changed to accusative "*o*."

Among these operations, this paper focuses on the last two, namely handling the element (b), the sibling cases of the deverbal noun. Triangles in both trees in Figure 1 indicate the range which we handle. Henceforth, elements outside of the triangles, namely, (a), (c), and (d), are used only for explanatory purposes.

4 LCS-based paraphrase generation model

Figure 2 illustrates how our model paraphrases the LVC, taking (7) as an example.

(7) s. *Ken-ga* *eiga-ni* ***shigeki-o*** ***uke**-ta*.
Ken-NOM film-DAT inspiration-ACC to receive-PAST
Ken received an inspiration from the film.

t. *Ken-ga* *eiga-ni* ***shigeki-s-<u>are</u>**-ta*.
Ken-NOM film-DAT to inspire-<u>PASSIVE</u>-PAST
Ken was inspired by the film.

The generation process consists of the following three steps:

Step 1. Semantic analysis: The model first analyzes a given input sentence including an LVC to obtain its LCS representation. In Figure 2, this step generates LCS_{V1} by filling arguments of LCS_{V0} with nominal elements.

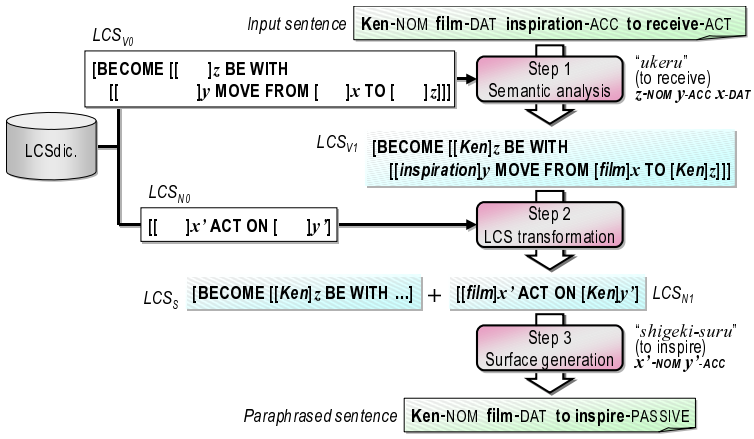

Fig. 2. LCS-based paraphrase generation model.

Step 2. Semantic transformation (LCS transformation): The model then transfers the obtained semantic structure to another semantic structure so that the target structure consists of the LCS of the verbalized form of the deverbal noun. In our example, this step generates LCS_{N1} together with the supplement "[BECOME [. . .]]". We refer to such a supplement as LCS_S.

Step 3. Surface generation: Having obtained the target LCS representation, the model finally generates the output sentence from it. LCS_S triggers another syntactic alternation such as passivization and causativization.

The idea is to use the LCS representation as a semantic representation and to retrieve semantic constraints to relieve the syntactic underspecificity underlying the LVC paraphrasing. Each step consists of a handful of linguistically explainable rules, and thus is scalable when the typology and resource of LCS is given. The rest of this section elaborates on each step, differentiating symbols to denote arguments; x, y, and z for LCS_V, and x', y', and z' for LCS_N.

4.1 Semantic analysis

Given an input sentence (a simple clause with an LVC), the model first looks up the LCS template LCS_{V0} for the given light-verb in the T-LCS dictionary, and then applies the case assignment rule below to obtain its LCS representation LCS_{V1}:

- In the case of the LCS_{V0} having argument x, fill the leftmost argument of the LCS_{V0} with the nominative case of the input, the second leftmost with the accusative, and the rest with the dative case.
- Otherwise, fill arguments y and z of the LCS_{V0} with the nominative and the dative cases, respectively.

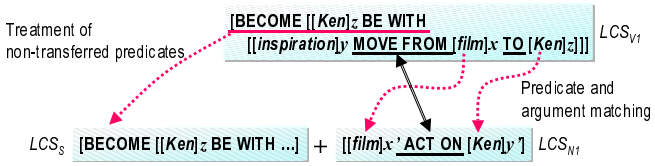

Fig. 3. An example of LCS transformation.

This rule is proposed in [19] instead of semantic parsing in order to tentatively automate LCS-based processing. In the example shown in Figure 2, LCS_{V0} for the given light-verb "*ukeru* (to receive)" has argument x, thus the nominative case, "*Ken*," fills the leftmost argument z. Accordingly, the accusative ("*shigeki* (inspiration)") and the dative ("*eiga* (film)") fill y and x, respectively.

4.2 LCS transformation

The second step matches LCS_{V1} with the another LCS for the verbalized form of the deverbal noun LCS_{N0} to generate the target LCS representation LCS_{N1}. Figure 3 shows a more detailed view of this process for the example shown in Figure 2.

Muraki [14] described that the direction of action and the focus of statement are important clues to determine the voice in LVC paraphrasing. We therefore incorporate the below assumptions into matching process. The model first matches predicates in LCS_{V1} and LCS_{N0}, assuming that the agentive argument x is relevant to the direction of action. We classify the semantic predicates into the following three groups: (i) agentive predicates (involve argument x): "CONTROL," "ACT ON," "ACT TO," "ACT," and "MOVE FROM TO," (ii) state of affair predicates (involve only argument y or z): "MOVE TO," "BE AT," and "BE WITH," and (iii) aspectual predicates (with no argument): "BECOME," and allowed any pair of predicates in the same group to match. In our example, "MOVE FROM TO" matches "ACT ON" as shown in Figure 3.

Having matched the predicates, the model then fills each argument slot in LCS_{N0} with its corresponding argument in LCS_{V1}. In Figure 3, argument z is matched with y', and x with x'. As a result, "*Ken*" and "*eiga*" come to y' and x' slots, respectively. When an argument is filled with another LCS, arguments within the inner LCS are also taken into account. Likewise, we introduced some exceptional rules assuming that the input sentences are periphrastic. For instance, arguments filled with the implicit filler (e.g. "name" for "to sign" is usually not expressed in Japanese) and the deverbal noun, which is already represented by LCS_{N0} are never matched. Argument z in LCS_{V1} is allowed to match with y' in LCS_{N0}.

LCS representations have right-embedding structures, and inner-embedded predicates denote the state of affairs. We thus prioritize the rightmost predicates in this matching process. In other words, the proceeds from the rightmost inner predicates to the outer ones, and the matching process is repeated until the leftmost predicate in LCS_{N0} or that in LCS_{V1} matched.

If LCS_{V1} has any non-transferred part LCS_S when the predicate and argument matching has been completed, it represents the semantic content that is not expressed by LCS_{N1} and needs to be expressed by auxiliary linguistic devices such as voice auxiliaries. As described in Section 2.1, the leftmost part specifies the focus of statement. The model thus attaches LCS_S to LCS_{N0} as a supplement, and then use it to determine auxiliaries in the next step, the surface generation. In the case of Figure 3, "[BECOME [[*Ken*]$_z$ BE WITH]]" in LCS_{V1} remains non-transferred and be attached.

4.3 Surface generation

The model again applies the aforementioned case assignment rule to generate a sentence from the resultant LCS. From the LCS_{N1} in Figure 2, sentence (8) is generated.

(8) *eiga-**ga*** *Ken-**o*** ***shigeki-shi**-ta*.
film-NOM Ken-ACC to inspire-PAST
The film inspired Ken.

The model then makes the final decision on the selection of the voice and the reassignment of the cases. As we described above, the attached structure LCS_S is a clue to determine what the focus is. We therefore use the following decision list:

1. If the leftmost argument of LCS_S has the same value as the leftmost argument in LCS_{N1}, the viewpoints of LCS_S and LCS_{N1} are same. Thus, the active voice is selected and the case structure is left as is.
2. If the leftmost argument of LCS_S has the same value as either z' or y' in LCS_{N1}, the model makes the argument a subject (nominative). That is, the passive voice is selected and case alternation (passivization) is applied.
3. If LCS_S has "BE WITH" and its argument has the same value as x' in LCS_{N1}, the causative voice is selected and case alternation (causativization) is applied.
4. If LCS_S has an agentive predicate, and its argument is filled with a value different from those of the other arguments, then the causative voice is selected and case alternation (causativization) is applied.
5. Otherwise, active voice is selected and thus no modification is applied.

The example in Figure 2 satisfies the second condition, thus the model chooses "*s-**are**-ru* (PASSIVE)" and passivizes the sentence (8). As a result, "*Ken*" becomes to be the nominative "*ga*" as in (7t).

5 Experiment

5.1 Paraphrase generation and evaluation

To conduct an empirical experiment, we collected the following data sets. Note that more than one LCS was assigned to a verb if it was polysemous.

Deverbal nouns: We regard "*sahen*-nouns" and adverbial forms of verbs as deverbal nouns. We retrieved 1,210 deverbal nouns from the T-LCS dictionary. The set consists of (i) activity nouns (e.g., "*sasoi* (invitation)" and "*odoroki* (surprise)"), (ii) Sino-Japanese verbal nouns (e.g., "*kandou* (impression)" and "*shigeki* (inspiration)"), and (iii) English borrowings (e.g., "drive" and "support").

Tuples of light-verb and case particle: A verb takes different meanings when it constitutes LVCs with different case particles, and not every tuple of a light-verb v and a case particle c functions as an LVC. We therefore tailored an objective collection of tuples $\langle v, c \rangle$ from corpus in the following manner:

Step 1. From a corpus consisting of 25 million parsed sentences of newspaper articles, we collected 876,101 types of triplet $\langle v, c, n \rangle$, where v, c, and n denote a base form of verb, a case particle, and an deverbal noun.

Step 2. For each of the 50 most frequent $\langle v, c \rangle$ tuples, we extracted the 10 most frequent triplets $\langle v, c, n \rangle$.

Step 3. Each $\langle v, c, n \rangle$ was manually evaluated to determine whether it functioned as an LVC. If any of 10 triplets functioned as an LVC, the tuple $\langle v, c \rangle$ was merged into the list of light-verbs, assigning an LCS according to the linguistic tests examined in [19]. As a result, we collected 40 types of $\langle v, c \rangle$ for light-verbs.

Paraphrase examples: A collection of paraphrase examples, pairs of an LVC and its correct paraphrase, were constructed in the following way:

Step 1. From the 876,101 types of triplet $\langle v, c, n \rangle$ collected above, 23,608 types of $\langle v, c, n \rangle$ were extracted, whose components, n and $\langle v, c \rangle$, were in the dictionaries.

Step 2. For each of the 245 most frequent $\langle v, c, n \rangle$, the 3 most frequent simple clauses including the $\langle v, c, n \rangle$ were extracted from the same corpus.

Step 3. Two native speakers of Japanese, adults graduated from university, were employed to build a gold-standard collection. 711 out of 735 sentences were manually paraphrased in the manner of LVC, while the remaining 24 sentences were not because $\langle v, c, n \rangle$ within them did not function as LVCs.

The real coverage of these 245 $\langle v, c, n \rangle$ with regard to all LVCs among the corpus falls in the range between the below two:

Lower bound: If every $\langle v, c, n \rangle$ is an LVC, the coverage of the collection is estimated at 6.47% (492,737 / 7,621,089) of tokens.

Upper bound: If the dictionaries cover all light-verbs and deverbal nouns, the collection covers 24.1% (492,737 / 2,044,387) of tokens.

In the experiment, our model generated all the possible paraphrases when a given verb was polysemous with multiple entries in the T-LCS dictionary. As a result, the model generated 822 paraphrases from the 735 input sentences, at least one for each input. We then classified the resultant paraphrases as correct and incorrect by comparing them with the gold-standard, where we ignored ordering of syntactic cases, and obtained 624 correct and 198 incorrect paraphrases Recall, precision, and F-measure ($\alpha = 0.5$) were 0.878 (624 / 711), 0.759 (624 / 822), and 0.814, respectively.

As the baseline, we employed a statistical language model developed in [5]. Among all the combinations of the voice and syntactic cases, the baseline model selects the one that has the highest probability. Although the model is trained on a large amount of data, the generated expression often falls out of the vocabulary. In such a case, the probability cannot be calculated, and the model outputs nothing for the given sentence. As a result of an application of this baseline model to the same set of input sentences,

we obtained 320 correct and 215 incorrect paraphrases (Recall: 0.450 (320 / 711), Precision: 0.598 (320 / 535), and F-measure: 0.514). The significant improvement indicates that our lexical-semantics-based account benefited on the decisions we considered.

The language model can also be complementary used to our LCS-based paraphrase generation. By filtering implausible paraphrases out, 66 incorrect and 15 correct paraphrases were filtered, and the performance was further improved (Recall: 0.857, Precision: 0.822, and F-measure: 0.839).

5.2 Discussion

Although the performance has room for further improvement, we think the performance is reasonably high under the current stage of the T-LCS dictionary. In other words, the tendency of errors does not so differ from our expectation. As we expected in Section 2.2, the ambiguity of dative case "*ni*" (c.f. (5)) occupied the largest portion of errors (78 / 198). This was because the case assignment was performed by a rule instead of semantic parsing. Each rule in our model has been created relying on a set of linguistic tests used in the theory of LCS and our linguistic intuition on handling LCS. However, the rule set was not sufficiently sophisticated, so that led to 59 errors. Equally, 30 errors occurred due to the immature typology of the T-LCS dictionary.

We consider the improvement of the LCS typology as the primal issue, because our transformation rules depend on it. For the moment, we have the following two suggestions. First, more variety of semantic roles should be handled step by step. For example, we need to handle the object of "*eikyou-suru* (to affect)," which is marked by not accusative but dative. Second, the necessity of "Source" is inconsistent. Verbs such as "*hairu* (to enter)" do not require this argument ("BECOME BE AT") , while some other verbs, such as "*ukeru* (to receive)," explicitly require it ("MOVE FROM TO"). The telicity of "MOVE FROM TO" should also be discussed. With such a feedback from the application and an extensive investigation into lexicology, we have to enhance the typology, and enlarge the dictionary preserving its consistency.

6 Related work

The paraphrases associated with LVCs are not idiosyncratic to Japanese but also appear commonly in other languages such as English, French, and Spanish [13, 7, 4] as shown in (3) and (4). Our approach raises an interesting issue of whether the paraphrasing of LVCs can be modeled in an analogous way across languages.

Iordanskaja *et al*. [7] proposed a set of paraphrasing rules including one for LVC paraphrasing based on the Meaning-Text Theory introduced by [13]. The model seemed to properly handle LVC paraphrasing, because their rules were described according to the deep semantic analysis and heavily relied on what were called lexical functions, such as lexical derivation (e.g., $S_0(\mathit{affect}) = \mathit{effect}$) and light-verb generation (e.g., $\mathit{Oper}_1(\mathit{attempt}) = \mathit{make}$). To take this approach, however, a vast amount of lexical knowledge to form each lexical function is required, because they only virtually specify all the choices relevant to LVC paraphrasing for every combination of deverbal noun and light-verb individually. In contrast, our approach is to employ lexical semantics

to provide a general account of those classes of choices, and thus contributes to the knowledge development in terms of reducing human-labor and preserving consistency.

Kaji *et al*. [10] proposed a paraphrase generation model which utilized an monolingual dictionary for human. Given an input LVC, their model paraphrases it referring to the glosses of both the deverbal noun and light-verb, and a manually assigned semantic feature of the light-verb. Their model looks robust due to the availability of resource. However, their model fails to explain the difference between examples (7) and (9) in the voice selection, because it selects the voice based only on the light-verb irrespective of the deverbal noun: the light-verb "*ukeru* (to receive)" is always mapped to the passive voice.

(9) s. *musuko-ga* *kare-no* *hanashi-ni* ***kandou-o*** ***uke**-ta*.
son-NOM his-GEN talk-DAT impression-ACC to receive-PAST
My son was given a good impression by his talk.

t. *musuko-ga* *kare-no* *hanashi-ni* ***kandou-shi**-ta*.
son-NOM his-GEN talk-DAT to be impressed-PAST
My son was impressed by his talk.

In their model, the target expression is restricted only to the LVC itself (c.f., Figure 1). Hence, their model is unable to reassign the case particles as we saw in example (6).

There is another trend in the research of paraphrase generation: i.e., the automatic paraphrase acquisition from existing lexical resources such as ordinary dictionaries, parallel/comparable corpora, and non-parallel corpora. This type of approach may be able to reduce the cost of resource development. However, there are drawbacks that must be overcome before they can work practically. First, automatic methods require large amounts of training data. The issue is how to collect enough large size of data at low cost. Second, automatically extracted knowledge tends to be rather noisy, requiring manual correction and maintenance. In contrast, our approach, which focuses on the regularity underlying paraphrases, is a complementary avenue to develop and maintain knowledge resources that cover a sufficiently wide range of paraphrases.

Previous case studies [14, 18, 11] have employed some syntactic properties of verbs to constrain syntactic transformations in paraphrase generation: e.g. subject agentivity, aspectual property, passivizability, and causativizability. Several classifications of verbs have also been proposed [12, 15] based on various types of verb alternation and syntactic case patterns. In contrast, the theory of lexical semantics integrates syntactic and semantic properties including those above, and gives a perspective to formalize and maintain the syntactic and semantic properties of words.

7 Conclusion

In this paper, we explored what sorts of lexical properties encoded in LCS can explain the regularity underlying paraphrases. Based on an existing LCS dictionary, we built an LCS-based paraphrase generation model, and conducted an empirical experiment on paraphrasing of LVC. The experiment confirmed that the proposed model was capable of generating paraphrases accurately in terms of selecting the voice and reassigning the syntactic cases, and revealed potential difficulties that we have to overcome toward a

practical use of our lexical-semantics-based account. To make our model more accurate, we need further discussion on (i) the enhancement of the T-LCS dictionary with feedback from experiments, (ii) the LCS transformation algorithm, and (iii) the semantic parsing. Another goal is to practically clarify what extent can be done by LCS for other classes of paraphrase, such as those exemplified in Section 1.

References

1. C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics* (*COLING-ACL*), pages 86–90, 1998.
2. X. Carreras and L. Màrques. Introduction to the CoNLL-2004 shared task: semantic role labeling. In *Proceedings of 8th Conference on Natural Language Learning* (*CoNLL*), pages 89–97, 2004.
3. B. J. Dorr. Large-scale dictionary construction for foreign language tutoring and interlingual machine translation. *Machine Translation*, 12(4):271–322, 1997.
4. M. Dras. *Tree adjoining grammar and the reluctant paraphrasing of text*. Ph.D. thesis, Division of Information and Communication Science, Macquarie University, 1999.
5. A. Fujita, K. Inui, and Y. Matsumoto. Detection of incorrect case assignments in automatically generated paraphrases of Japanese sentences. In *Proceedings of the 1st International Joint Conference on Natural Language Processing* (*IJCNLP*), pages 14–21, 2004.
6. D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.
7. L. Iordanskaja, R. Kittredge, and A. Polguère. Lexical selection and paraphrase in a meaning-text generation model. In C. L. Paris, W. R. Swartout, and W. C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 293–312. Kluwer Academic Publishers, 1991.
8. R. Jackendoff. *Semantic structures*. The MIT Press, 1990.
9. T. Kageyama. *Verb semantics*. Kurosio Publishers, 1996. (in Japanese).
10. N. Kaji and S. Kurohashi. Recognition and paraphrasing of periphrastic and overlapping verb phrases. In *Proceedings of the 4th International Conference on Language Resources and Evaluation* (*LREC*) *Workshop on Methodologies and Evaluation of Multiword Units in Real-world Application*, 2004.
11. K. Kondo, S. Sato, and M. Okumura. Paraphrasing by case alternation. *IPSJ Journal*, 42(3):465–477, 2001. (in Japanese).
12. B. Levin. *English verb classes and alternations: a preliminary investigation*. Chicago Press, 1993.
13. I. Mel'čuk and A. Polguère. A formal lexicon in meaning-text theory (or how to do lexica with words). *Computational Linguistics*, 13(3-4):261–275, 1987.
14. S. Muraki. *Various aspects of Japanese verbs*. Hitsuji Syobo, 1991. (in Japanese).
15. A. Oishi and Y. Matsumoto. Detecting the organization of semantic subclasses of Japanese verbs. *International Journal of Corpus Linguistics*, 2(1):65–89, 1997.
16. M. Palmer, D. Gildea, and P. Kingsbury. The Proposition Bank: an annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.
17. J. Pustejovsky. *The generative lexicon*. The MIT Press, 1995.
18. S. Sato. Automatic paraphrase of technical papers' titles. *IPSJ Journal*, 40(7):2937–2945, 1999. (in Japanese).
19. K. Takeuchi, K. Kageura, and T. Koyama. An LCS-based approach for analyzing Japanese compound nouns with deverbal heads. In *Proceedings of the 2nd International Workshop on Computational Terminology* (*CompuTerm*), pages 64–70, 2002.