

Framework of Semantic Role Assignment based on Extended Lexical Conceptual Structure: Comparison with VerbNet and FrameNet

Yuichiroh Matsubayashi[†] Yusuke Miyao[†] Akiko Aizawa[†]

[†], National Institute of Informatics, Japan
{y-matsu, yusuke, aizawa}@nii.ac.jp

Abstract

Widely accepted resources for semantic parsing, such as PropBank and FrameNet, are not perfect as a semantic role labeling framework. Their semantic roles are not strictly defined; therefore, their meanings and semantic characteristics are unclear. In addition, it is presupposed that a single semantic role is assigned to each syntactic argument. This is not necessarily true when we consider internal structures of verb semantics. We propose a new framework for semantic role annotation which solves these problems by extending the theory of lexical conceptual structure (LCS). By comparing our framework with that of existing resources, including VerbNet and FrameNet, we demonstrate that our extended LCS framework can give a formal definition of semantic role labels, and that multiple roles of arguments can be represented strictly and naturally.

1 Introduction

Recent developments of large semantic resources have accelerated empirical research on semantic processing (Màrquez et al., 2008). Specifically, corpora with semantic role annotations, such as PropBank (Kingsbury and Palmer, 2002) and FrameNet (Ruppenhofer et al., 2006), are indispensable resources for semantic role labeling. However, there are two topics we have to carefully take into consideration regarding role assignment frameworks: (1) clarity of semantic role meanings and (2) the constraint that a single semantic role is assigned to each syntactic argument.

While these resources are undoubtedly invaluable for empirical research on semantic process-

Sentence	[John]	threw	[a ball]	[from the window] .
Affection	Agent		Patient	
Movement	Source		Theme	Source/Path
PropBank	Arg0		Arg1	Arg2
VerbNet	Agent		Theme	Source
FrameNet	Agent		Theme	Source

Table 1: Examples of single role assignments with existing resources.

ing, current usage of semantic labels for SRL systems is questionable from a theoretical viewpoint. For example, most of the works on SRL have used PropBank's numerical role labels (Arg0 to Arg5). However, the meanings of these numbers depend on each verb in principle and PropBank does not expect semantic consistency, namely on Arg2 to Arg5. Moreover, Yi et al. (2007) explicitly showed that Arg2 to Arg5 are semantically inconsistent. The reason why such labels have been used in SRL systems is that verb-specific roles generally have a small number of instances and are not suitable for learning. However, it is necessary to avoid using inconsistent labels since those labels confuse machine learners and can be a cause of low accuracy in automatic processing. In addition, clarity of the definition of roles are particularly important for users to rationally know how to use each role in their applications. For this reasons, well-organized and generalized labels grounded in linguistic characteristics are needed in practice. Semantic roles of FrameNet and VerbNet (Kipper et al., 2000) are used more consistently to some extent, but the definition of the roles is not given in a formal manner and their semantic characteristics are unclear.

Another somewhat related problem of existing annotation frameworks is that it is presupposed

that a single semantic role is assigned to each syntactic argument.¹In fact, one syntactic argument can play multiple roles in the event (or events) expressed by a verb. For example, Table 1 shows a sentence containing the verb “throw” and semantic roles assigned to its arguments in each framework. The table shows that each framework assigns a single role, such as Arg0 and Agent, to each syntactic argument. However, we can acquire information from this sentence that John is an agent of the throwing event (the “Affection” row), as well as a source of the movement event of the ball (the “Movement” row). Existing frameworks of assigning single roles simply ignore such information that verbs inherently have in their semantics. We believe that giving a clear definition of multiple argument roles would be beneficial not only as a theoretical framework but also for practical applications that require detailed meanings derived from secondary roles.

This issue is also related to fragmentation and the unclear definition of semantic roles in these frameworks. As we exemplify in this paper, multiple semantic characteristics are conflated in a single role label in these resources due to the manner of single-role assignment. This means that semantic roles of existing resources are not monolithic and inherently not mutually independent, but they share some semantic characteristics.

The aim of this paper is more on theoretical discussion for role-labeling frameworks rather than introducing a new resource. We developed a framework of verb lexical semantics, which is an extension of the lexical conceptual structure (LCS) theory, and compare it with other existing frameworks which are used in VerbNet and FrameNet, as an annotation scheme of SRL. LCS is a decomposition-based approach to verb semantics and describes a meaning by composing a set of primitive predicates. The advantage of this approach is that primitive predicates and their compositions are formally defined. As a result, we can give a strict definition of semantic roles by grounding them to lexical semantic structures of verbs. In fact, we define semantic roles as argument slots in primitive predicates. With this ap-

¹To be precise, FrameNet permits multiple-role assignment, while it does not perform this systematically as we show in Table 1. It mostly defines a single role label for a corresponding syntactic argument, that plays multiple roles in several sub-events in a verb.

proach, we demonstrate that some sort of semantic characteristics that VerbNet and FrameNet informally/implicitly describe in their roles can be given formal definitions and that multiple argument roles can be represented strictly and naturally by extending the LCS theory.

In the first half of this paper, we define our extended LCS framework and describe how it gives a formal definition of roles and solves the problem of multiple roles. In the latter half, we discuss the analysis of the empirical data we collected for 60 Japanese verbs and also discuss theoretical relationships with the frameworks of existing resources. We discuss in detail the relationships between our role labels and VerbNet’s thematic roles. We also describe the relationship between our framework and FrameNet, with regards to the definitions of the relationships between semantic frames.

2 Related works

There have been several attempts in linguistics to assign multiple semantic properties to one argument. Gruber (1965) demonstrated the dispensability of the constraint that an argument takes only one semantic role, with some concrete examples. Rozwadowska (1988) suggested an approach of feature decomposition for semantic roles using her three features of *change*, *cause*, and *sentient*, and defined typical thematic roles by combining these features. This approach made it possible for us to classify semantic properties across thematic roles. However, Levin and Rappaport Hovav (2005) argued that the number of combinations using defined features is usually larger than the actual number of possible combinations; therefore, feature decomposition approaches should predict possible feature combinations.

Culicover and Wilkins (1984) divided their roles into two groups, action and perceptual roles, and explained that dual assignment of roles always involves one role from each set. Jackendoff (1990) proposed an LCS framework for representing the meaning of a verb by using several primitive predicates. Jackendoff also stated that an LCS represents two *tiers* in its structure, *action tier* and *thematic tier*, which are similar to Culicover and Wilkins’s two sets. Essentially, these two approaches distinguished roles related to action and change, and successfully restricted com-

$$\left[\text{cause}(\text{affect}(i,j), \text{go}(j, \left[\begin{array}{l} \text{from}(\text{locate}(\text{in}(i))) \\ \text{forward}(\text{locate}(\text{at}(k))) \\ \text{toward}(\text{locate}(\text{at}(l))) \end{array} \right])) \right]$$

Figure 1: LCS of the verb *throw*.

binations of roles by taking a role from each set.

Dorr (1997) created an LCS-based lexical resource as an interlingual representation for machine translation. This framework was also used for text generation (Habash et al., 2003). However, the problem of multiple-role assignment was not completely solved on the resource. As a comparison of different semantic structures, Dorr (2001) and Hajičová and Kučerová (2002) analyzed the connection between LCS and PropBank roles, and showed that the mapping between LCS and PropBank roles was many to many correspondence and roles can map only by comparing a whole argument structure of a verb. Habash and Dorr (2001) tried to map LCS structures into thematic roles by using their thematic hierarchy.

3 Multiple role expression using lexical conceptual structure

Lexical conceptual structure is an approach to describe a generalized structure of an event or state represented by a verb. A meaning of a verb is represented as a structure composed of several primitive predicates. For example, the LCS structure for the verb “throw” is shown in Figure 1 and includes the predicates *cause*, *affect*, *go*, *from*, *forward*, *toward*, *locate*, *in*, and *at*. The arguments of primitive predicates are filled by core arguments of the verb. This type of decomposition approach enables us to represent a case that one syntactic argument fills multiple slots in the structure. In Figure 1, the argument *i* appears twice in the structure: as the first argument of *affect* and the argument in *from*.

The primitives are designed to represent a full or partial *action-change-state* chain, which consists of a state, a change in or maintaining of a state, or an action that changes/maintains a state. Table 2 shows primitives that play important roles to represent that chain. Some primitives embed other primitives as their arguments and the semantics of the entire structure of an LCS structure is calculated according to the definition of each primitive. For instance, the LCS structure in Fig-

Predicates	Semantic Functions
$\text{state}(x, y)$	First argument is in state specified by second argument.
$\text{cause}(x, y)$	Action in first argument causes change specified in second argument.
$\text{act}(x)$	First argument affects itself.
$\text{affect}(x, y)$	First argument affects second argument.
$\text{react}(x, y)$	First argument affects itself, due to the effect from second argument.
$\text{go}(x, y)$	First argument changes according to the path described in the second argument.
$\text{from}(x)$	Starting point of certain change event.
$\text{forward}(x)$	Direction of starting point.
$\text{via}(x)$	Pass point of certain change event.
$\text{toward}(x)$	Direction of end point.
$\text{to}(x)$	End point of certain change event.
$\text{along}(x)$	Linear-shaped path of change event.

Table 2: Major primitive predicates and their semantic functions.

ure 1 represents the action changing the state of *j*. The inner structure of the second argument of *go* represents the path of the change.

The overall definition of our extended LCS framework is shown in Figure 2.² Basically, our definition is based on Jackendoff’s LCS framework (1990), but performed some simplifications and added extensions. The modification is performed in order to increase strictness and generality of representation and also a coverage for various verbs appearing in a corpus. The main differences between the two LCS frameworks are as follows. In our extended LCS framework, (i) the possible combinations of *cause*, *act*, *affect*, *react*, and *go* are clearly restricted, (ii) multiple actions or changes in an event can be described by introducing a *combination* function (*comb* for short), (iii) *GO*, *STAY* and *INCH* in Jackendoff’s theory are incorporated into one function *go*, and (iv) most of the *change-of-state* events are represented as a metaphor using a spatial transition.

The idea of a *comb* function comes from a natural extension of Jackendoff’s *EXCH* function. In our case, *comb* is not limited to describing a counter-transfer of the main event but can describe subordinate events occurring in relation to the main event.³ We can also describe multiple

²Here we omitted the attributes taken by each predicate, in order to simplify the explanation. We also omitted an explanation for lower level primitives, such as STATE and PLACE groups, which are not necessarily important for the topic of this paper.

³In our extended LCS theory, we can describe multiple

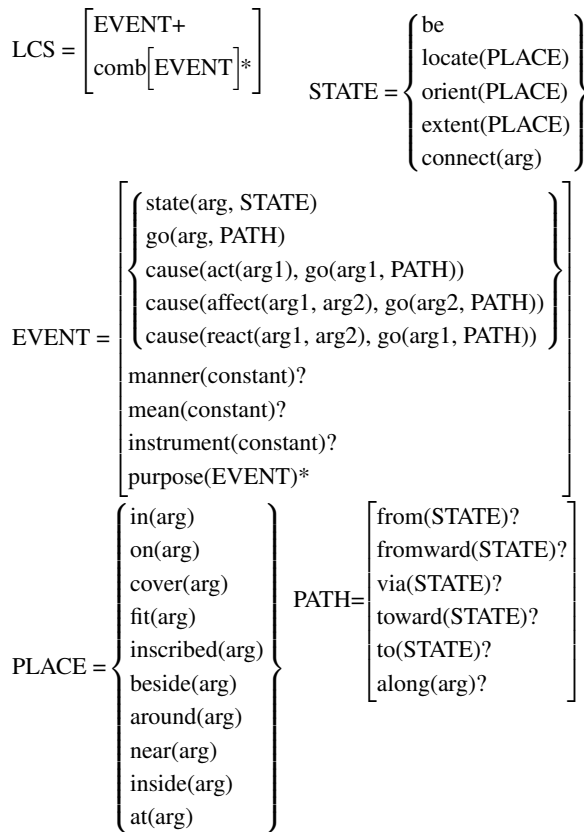


Figure 2: Description system of our LCS. Operators +, *, ? follow the basic regular expression syntax. {} represents a choice of the elements.

main events if the agent does more than two actions simultaneously and all the actions are the focus (e.g., John exchanges A with B). This extension is simple, but essential for creating LCS structures of predicates appearing in actual data. In our development of 60 Japanese predicates (verb and verbal noun) frequently appearing in Kyoto University Text Corpus (KTC) (Kurohashi and Nagao, 1997), 37.6% of the frames included multiple events. By using the *comb* function, we can express complicated events with predicate decomposition and prevent missing (multiple) roles.

A key point for associating LCS framework with the existing frameworks of semantic roles is that each primitive predicate of LCS represents a fundamental function in semantics. The func-

events in the semantic structure of a verb. However, generally, a verb focuses on one of those events and this makes a semantic variation among verbs such as buy, sell, and pay as well as difference of syntactic behavior of the arguments. Therefore, focused event should be distinguished from the others as lexical information. We expressed focused events as main formulae (formulae that are not surrounded by a *comb* function).

Role	Description
Protagonist	Entity which is viewpoint of verb.
Theme	Entity in which its state or change of state is mentioned.
State	Current state of certain entity.
Actor	Entity which performs action that changes/maintains its state.
Effector	Entity which performs action that changes/maintains a state of another entity.
Patient	Entity which is changed/maintained its state by another entity.
Stimulus	Entity which is cause of the action.
Source	Starting point of certain change event.
Source_dir	Direction of starting point.
Middle	Pass point of certain change event.
Goal	End point of certain change event.
Goal_dir	Direction of end point.
Route	Linear-shaped path of certain change event.

Table 3: Semantic role list for proposing extended LCS framework.

tions of the arguments of the primitive predicates can be explained using generalized semantic roles such as typical thematic roles. In order to simply represent the semantic functions of the arguments in the LCS primitives or make it easier to compare our extended LCS framework with other SRL frameworks, we define a semantic role set that corresponds to the semantic functions of the primitive predicates in the LCS structure (Table 3). We employed role names similarly to typical thematic roles in order to easily compare the role sets, but the definition is different. Also, due to the increase of the generality of LCS representation, we obtained clearer definition to explain a correspondence between LCS primitives and typical thematic roles than the Jackendoff's predicates. Note that the core semantic information of a verb represented by a LCS framework is embodied directly in its LCS structure and the information decreases if the structure is mapped to the semantic roles. The mapping is just for contrasting thematic roles. Each role is given an obvious meaning and designed to fit to the upper-level primitives of the LCS structure, which are the arguments of EVENT and PATH functions. In Table 4, we can see that these roles correspond almost one-to-one to the primitive arguments. One special role is *Protagonist*, which does not match an argument of a specific primitive. The *Protagonist* is assigned to the first argument in the main formula to distinguish that formula from the sub formulae. There are 13 defined roles, and

Predicate	1st arg	2nd arg
state	Theme	State
act	Actor	–
affect	Effector	Patient
react	Actor	Stimulus
go	Theme	PATH
from	Source	–
forward	Source_dir	–
via	Middle	–
toward	Goal_dir	–
to	Goal	–
along	Route	–

Table 4: Correspondence between semantic roles and arguments of LCS primitives

this number is comparatively smaller than that in VerbNet. The discussion with regard to this number is described in the next section.

Essentially, the semantic functions of the arguments in LCS primitives are similar to those of traditional, or basic, thematic roles. However, there are two important differences. Our extended LCS framework principally guarantees that the primitive predicates do not contain any information concerning (i) selectional preference and (ii) complex structural relation of arguments. Primitives are designed to purely represent a function in an *action-change-state* chain, thus the information of selectional preference is annotated to a different layer; specifically, it is directly annotated to core arguments (e.g., we can annotate *i* with *sel-Pref(animate ∨ organization)* in Figure 1). Also, the semantic function is already decomposed and the structural relation among the arguments is represented as a structure of primitives in LCS representation. Therefore, each argument slot of the primitive predicates does not include complicated meanings and represents a primitive semantic property which is highly functional. These characteristics are necessary to ensure clarity of the semantic role meanings. We believe that even though there surely exists a certain type of complex semantic role, it is reasonable to represent that role based on decomposed properties.

In order to show an instance of our extended LCS theory, we constructed a dictionary of LCS structures for 60 Japanese verbs (including event nouns) using our extended LCS framework. The 60 verbs were the most frequent verbs in KTC after excluding 100 most frequent ones.⁴ We cre-

⁴We omitted top 100 verbs since these most frequent ones

Role	Single	Multiple	Grow (%)
Theme	21	108	414
State	1	1	0
Actor	12	13	8.3
Effector	73	92	26
Patient	77	79	2.5
Stimulus	0	0	0
Source	11	44	300
Source_dir	4	4	0
Middle	1	8	700
Goal	42	81	93
Goal_dir	2	3	50
Route	2	2	0
w/o Theme	225	327	45
Total	246	435	77

Table 5: Number of appearances of each role

ated the dictionary looking at the instances of the target verbs in KTC. To increase the coverage of senses and case frames, we also consulted the online Japanese dictionary *Digital Daijisen*⁵ and Kyoto university case frames (Kawahara and Kurohashi, 2006) which is a compilation of case frames automatically acquired from a huge web corpus. There were 97 constructed frames in the dictionary.

Then we analyzed how many roles are additionally assigned by permitting multiple role assignment (see Table 5). The numbers of assigned roles for single role are calculated by counting roles that appear first for each target argument in the structure. Table 5 shows that the total number of assigned roles is 1.77 times larger than single-role assignment. The main reason is an increase in *Theme*. For single-role assignment, *Theme*, in our sense, in action verbs is always duplicated with *Actor/Patient*. On the other hand, LCS strictly divides a function for action and change; therefore the duplicated *Theme* is correctly annotated. Moreover, we obtained a 45% increase even when we did not count duplicated *Theme*. Most of increase are a result from the increase in *Source* and *Goal*. For example, *Effectors* of transmission verbs are also annotated with a *Source*, and *Effectors* of movement verbs are sometimes annotated with *Source* or *Goal*.

contain a phonogram form (Hiragana form) of a certain verb written with Kanji characters, and that phonogram form generally has a huge ambiguity because many different verbs have same pronunciation in Japanese.

⁵Available at <http://dictionary.goo.ne.jp/jn/>.

Resource	Frame-independent	# of roles
LCS	yes	13
VerbNet (v3.1)	yes	30
FrameNet (r1.4)	no	8884

Table 6: Number of roles in each resource.

4 Comparison with other resources

4.1 Number of semantic roles

The number of roles is related to the number of semantic properties represented in a framework and to the generality of that property. Table 6 lists the number of semantic roles defined in our extended LCS framework, VerbNet and FrameNet.

There are two ways to define semantic roles. One is *frame specific*, where the definition of each role depends on a specific lexical entry and such a role is never used in the other frames. The other is *frame independent*, which is to construct roles whose semantic function is generalized across all verbs. The number of roles in FrameNet is comparatively large because it defines roles in a frame-specific way. FrameNet respects individual meanings of arguments rather than generality of roles.

Compared with VerbNet, the number of roles defined in our extended LCS framework is less than half. However, this fact does not mean that the representation ability of our framework is lower than VerbNet. We manually checked and listed a corresponding representation in our extended LCS framework for each thematic role in VerbNet in Table 6. This table does not provide a perfect or complete mapping between the roles in these two frameworks because the mappings are not based on annotated data. However, we can roughly say that the VerbNet roles combine three types of information, a function of the argument in the action-change-state chain, selectional preference, and structural information of arguments, which are in different layers in LCS representation. VerbNet has many roles whose functions in the action-change-state chain are duplicated. For example, *Destination*, *Recipient*, and *Beneficiary* have the same property *end-state* (*Goal* in LCS) of a changing event. The difference between such roles comes from a specific sub-type of a changing event (possession), selectional preference, and structural information among the arguments. By distinguishing such roles, VerbNet roles may take

into account specific syntactic behaviors of certain semantic roles. Packing such complex information to semantic roles is useful for analyzing argument realization. However, from the viewpoint of semantic representation, the clarity for semantic properties provided using a predicate decomposition approach is beneficial. The 13 roles for the LCS approach is sufficient for obtaining a function in the *action-change-state* chain. In our LCS framework, selectional preference can be assigned to arguments in an individual verb or verb class level instead of role labels themselves to maintain generality of semantic functions. In addition, our extended LCS framework can easily separate complex structural information from role labels because LCS directly represents a structure among the arguments. We can calculate the information from the LCS structure instead of coding it into role labels. As a result, our extended LCS framework maintains generality of roles and the number of roles is smaller than other frameworks.

4.2 Clarity of role meanings

We showed that an approach of predicate decomposition used in LCS theory clarified role meanings assigned to syntactic arguments. Moreover, LCS achieves high generality of roles by separating selectional preference or structural information from role labels. The complex meaning of one syntactic argument is represented by multiple appearances of the argument in an LCS structure. For example, we show an LCS structure and a frame in VerbNet with regard to the verb “buy” in Figure 3. The LCS structure consists of four formulae. The first one is the main formula and the others are sub-formulae that represent co-occurring actions. The semantic-role-like representation of the structure is given in Table 4: $i = \{\text{Protagonist, Effector, Source, Goal}\}$, $j = \{\text{Patient, Theme}\}$, $k = \{\text{Effector, Source, Goal}\}$, and $l = \{\text{Patient, Theme}\}$. Selectional preference is annotated to each argument as i : $\text{selPref}(\text{animate} \vee \text{organization})$, j : $\text{selPref}(\text{any})$, k : $\text{selPref}(\text{animate} \vee \text{organization})$, and l : $\text{selPref}(\text{valuable_entity})$. If we want to represent the information, such as “*Source* of what?”, then we can extend the notation as $\text{Source}(j)$ to refer to a changing object.

On the other hand, VerbNet combines multiple types of information into a single role as mentioned above. Also, the meaning of some

VerbNet role (# of uses)	Representation in LCS
Actor (9), Actor1 (9), Actor2 (9)	Actor or Effector in symmetric formulas in the structure
Agent (212)	(Actor \vee Effector) \wedge Protagonist
Asset (6)	Theme \wedge Source of the change is (locate(in()) \wedge Protagonist) \wedge selPref(valuable entity)
Beneficiary (9)	(peripheral role \vee (Goal \wedge locate(in())) \wedge selPref(animate \vee organization) \wedge \neg (Actor \vee Effector) \wedge a transferred entity is something beneficial
Cause (21)	((Effector \wedge selPref(\neg animate \wedge \neg organization)) \vee Stimulus \vee peripheral role)
Destination (32)	Goal
Experiencer (24)	Actor of react()
Instrument (25)	((Effector \wedge selPref(\neg animate \wedge \neg organization)) \vee peripheral role)
Location (45)	(Theme \vee PATH roles \vee peripheral role) \wedge selPref(location)
Material (6)	Theme \vee Source of a change \wedge The Goal of the change is locate(fit()) \wedge the Goal fullfills selPref(physical_object)
Patient (59), Patient 1(11)	Patient \vee Theme
Patient2 (11)	(Source \vee Goal) \wedge connect()
Predicate (23)	Theme \vee (Goal \wedge locate(fit())) \vee peripheral role
Product (7)	Theme \vee (Goal \wedge locate(fit()) \wedge selPref(physical_object))
Proposition (11)	Theme
Recipient (33)	Goal \wedge locate(in()) \wedge selPref(animate \vee organization)
Source (34)	Source
Theme (162)	Theme
Theme1 (13), Theme2 (13)	Both of the two is Theme \vee Theme1 is Theme and Theme2 is State
Topic (18)	Theme \wedge selPref(knowledge \vee information)

Table 7: Relationship of roles between VerbNet and our LCS framework. VerbNet roles that appears more than five times in frame definition are analyzed. Each relationship shown here is only a partial and consistent part of the complete correspondence table. Note that complete table of mapping highly depends on each lexical entry (or verb class). Here, locate(in()) generally means possession or recognizing.

roles depends more on selectional preference or the structure of the arguments than a primitive function in the *action-change-state* chain. Such VerbNet roles are used for several different functions depending on verbs and their alternations, and it is therefore difficult to capture decomposed properties from the role label without having specific lexical knowledge. Moreover, some semantic functions, such as Mary is a *Goal* of the money in Figure 3, are completely discarded from the representation at the level of role labels.

There is another representation related to the argument meanings in VerbNet. This representation is a type of predicate decomposition using its original set of predicates, which are referred to as *semantic predicates*. For example, the verb “buy” in Figure 3 has the predicates *has_possession*, *transfer* and *cost* for composing the meaning of its event structure. The thematic roles are fillers of the predicates’ arguments, thus the semantic predicates may implicitly provide additional functions to the roles and possibly represent multiple roles. Unfortunately, we cannot discover what each argument of the semantic predicates exactly means since the definition of each predicate is not

Example: “John bought a book from Mary for \$10.”

VerbNet: Agent \vee Theme {from} Source {for} Asset.
has_possession(start(E), Source, Theme),
has_possession(end(E), Agent, Theme),
transfer(during(E), Theme), cost(E, Asset)

LCS:

$$\left[\begin{array}{l} \text{cause}(\text{aff}(i:\text{John}, j:\text{a book}), \text{go}(j, [\text{to}(\text{loc}(\text{in}(i)))))) \\ \text{comb} \left[\begin{array}{l} \text{cause}(\text{aff}(i,l:\$10), \text{go}(l, \left[\begin{array}{l} \text{from}(\text{loc}(\text{in}(i))) \\ \text{to}(\text{loc}(\text{at}(k:\text{Mary}))) \end{array} \right])) \end{array} \right] \\ \text{comb} \left[\begin{array}{l} \text{cause}(\text{aff}(k,j), \text{go}(j, \left[\begin{array}{l} \text{from}(\text{loc}(\text{in}(k))) \\ \text{to}(\text{loc}(\text{at}(i))) \end{array} \right])) \end{array} \right] \\ \text{comb} \left[\begin{array}{l} \text{cause}(\text{aff}(k,l), \text{go}(l, [\text{to}(\text{loc}(\text{in}(k)))))) \end{array} \right] \end{array} \right]$$

Figure 3: Comparison between the semantic predicate representation and the LCS structure of the verb *buy*.

publicly available. A requirement for obtaining implicit semantic functions from these semantic predicates is clearly defining how the roles (or functions) are calculated from these complex relations of semantic predicates.

FrameNet does not use semantic roles generalized among all verbs or does not represent seman-

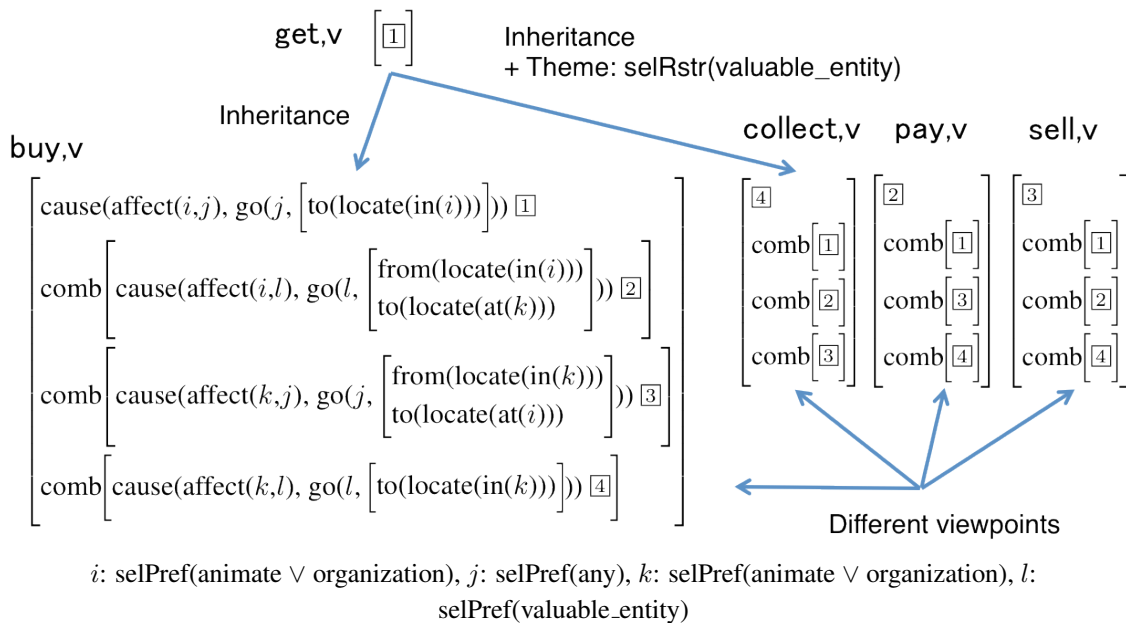


Figure 4: LCS of the verbs *get*, *buy*, *sell*, *pay*, and *collect* and their relationships calculated from the structures.

tic properties of roles using a predicate decomposition approach, but defines specific roles for each conceptual event/state to represent a specific background of the roles in the event/state. However, at the same time, FrameNet defines several types of parent-child relations between most of the frames and between their roles; therefore, we may say FrameNet implicitly describes a sort of decomposed property using roles in highly general or abstract frames and represents the inheritance of these semantic properties. One advantage of this approach is that the inheritance of a meaning between roles is controlled through the relations, which are carefully maintained by human efforts, and is not restricted by the representation ability of the decomposition system. On the other hand, the only way to represent generalized properties of a certain semantic role is enumerating all inherited roles by tracing ancestors. Also, a semantic relation between arguments in a certain frame, which is given by LCS structure and semantic predicates of VerbNet, is only defined by a natural language description for each frame in FrameNet. From a CL point of view, we consider that, at least, a certain level of formalization of semantic relation of arguments is important for utilize this information for application. LCS approach, or an approach using a well-defined predicate decomposition, can explicitly describe semantic properties and relationships between argu-

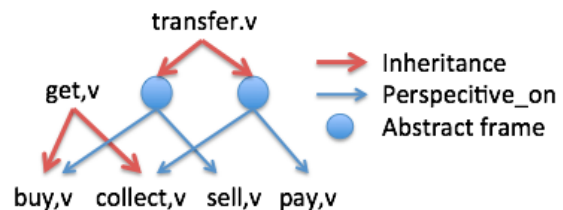


Figure 5: The frame relations among the verbs *get*, *buy*, *sell*, *pay*, and *collect* in FrameNet.

ments in a lexical structure. The primitive properties can be clearly defined, even though the representation ability is restricted under the generality of roles.

In addition, the frame-to-frame relations in FrameNet may be a useful resource for some application tasks such as paraphrasing and entailment. We argue that some types of relationships between frames are automatically calculated using the LCS approach. For example, one of the relations is based on an inclusion relation of two LCS structures. Figure 4 shows automatically calculated relations surrounding the verb “buy”. Note that we chose a sense related to a commercial transaction, which means a exchange of a goods and money, for each word in order to compare the resulted relation graph with that of FrameNet. We call relations among “buy”, “sell”, “pay” and “collect” as *different viewpoints* since

they contain exactly the same formulae, and the only difference is the main formula. The relation between “buy” and “get” is defined as *inheritance*; a part of the child structure exactly equals the parent structure. Interestingly, the relations surrounding the “buy” are similar to those in FrameNet (see Figure 5). We cannot describe all types of the relations we considered due to space limitations. However, the point is that these relationships are represented as rewriting rules between the two LCS representations and thus they are automatically calculated. Moreover, the grounds for relations maintain clarity based on concrete structural relations. A semantic relation construction of frames based on structural relationships is another possible application of LCS approaches that connects traditional LCS theories with resources representing a lexical network such as FrameNet.

4.3 Consistency on semantic structures

Constructing a LCS dictionary is generally a difficult work since LCS has a high flexibility for describing structures and different people tend to write different structures for a single verb. We maintained consistency of the dictionary by taking into account a similarity of the structures between the verbs that are in paraphrasing or entailment relations. This idea was inspired by automatic calculation of semantic relations of lexicon as we mentioned above. We created a LCS structure for each lexical entry as we can calculate semantic relations between related verbs and maintained high-level consistency among the verbs.

Using our extended LCS theory, we successfully created 97 frames for 60 predicates without any extra modification. From this result, we believe that our extended theory is stable to some extent. On the other hand, we found that an extra extension of the LCS theory is needed for some verbs to explain the different syntactic behaviors of one verb. For example, a condition for a certain syntactic behavior of a verb related to reciprocal alteration (see class 2.5 of Levin (Levin, 1993)) such as つながる (connect) and 統一 (integrate) cannot be explained without considering the number of entities in some arguments. Also, some verbs need to define an order of the internal events. For example, the Japanese verb 往復する (shuttle) means that going is a first action and coming back is a second action. These are not

the problems that are directly related to a semantic role annotation on that we focus in this paper, but we plan to solve these problems with further extensions.

5 Conclusion

We discussed the two problems in current labeling approaches for argument-structure analysis: the problems in clarity of role meanings and multiple-role assignment. By focusing on the fact that an approach of predicate decomposition is suitable for solving these problems, we proposed a new framework for semantic role assignment by extending Jackendoff’s LCS framework. The statistics of our LCS dictionary for 60 Japanese verbs showed that 37.6% of the created frames included multiple events and the number of assigned roles for one syntactic argument increased 77% from that in single-role assignment.

Compared to the other resources such as VerbNet and FrameNet, the role definitions in our extended LCS framework are clearer since the primitive predicates limit the meaning of each role to a function in the *action-change-state* chain. We also showed that LCS can separate three types of information, the functions represented by primitives, the selectional preference and structural relation of arguments, which are conflated in role labels in existing resources. As a potential of LCS, we demonstrated that several types of frame relations, which are similar to those in FrameNet, are automatically calculated using the structural relations between LCSs. We still must perform a thorough investigation for enumerating relations which can be represented in terms of rewriting rules for LCS structures. However, automatic construction of a consistent relation graph of semantic frames may be possible based on lexical structures.

We believe that this kind of decomposed analysis will accelerate both fundamental and application research on argument-structure analysis. As a future work, we plan to expand the dictionary and construct a corpus based on our LCS dictionary.

Acknowledgment

This work was partially supported by JSPS Grant-in-Aid for Scientific Research #22800078.

References

- P.W. Culicover and W.K. Wilkins. 1984. *Locality in linguistic theory*. Academic Press.
- Bonnie J. Dorr. 1997. Large-scale dictionary construction for foreign language tutoring and interlingual machine translation. *Machine Translation*, 12(4):271–322.
- Bonnie J. Dorr. 2001. Lcs database. http://www.umiacs.umd.edu/~bonnie/LCS_Database_Documentation.html.
- Jeffrey S Gruber. 1965. *Studies in lexical relations*. Ph.D. thesis, MIT.
- N. Habash and B. Dorr. 2001. Large scale language independent generation using thematic hierarchies. In *Proceedings of MT summit VIII*.
- N. Habash, B. Dorr, and D. Traum. 2003. Hybrid natural language generation from lexical conceptual structures. *Machine Translation*, 18(2):81–128.
- Eva Hajičová and Ivona Kučerová. 2002. Argument/valency structure in propbank, lcs database and prague dependency treebank: A comparative pilot study. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 846–851.
- Ray Jackendoff. 1990. *Semantic Structures*. The MIT Press.
- D. Kawahara and S. Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proceedings of LREC-2006*, pages 1344–1347.
- Paul Kingsbury and Martha Palmer. 2002. From Treebank to PropBank. In *Proceedings of LREC-2002*, pages 1989–1993.
- Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of the National Conference on Artificial Intelligence*, pages 691–696. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- Sadao Kurohashi and Makoto Nagao. 1997. Kyoto university text corpus project. *Proceedings of the Annual Conference of JSAL*, 11:58–61.
- Beth Levin and Malka Rappaport Hovav. 2005. *Argument realization*. Cambridge University Press.
- Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago Press.
- Lluís Màrquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. 2008. Semantic role labeling: an introduction to the special issue. *Computational linguistics*, 34(2):145–159.
- B. Rozwadowska. 1988. Thematic restrictions on derived nominals. In W Wlikins, editor, *Syntax and Semantics*, volume 21, pages 147–165. Academic Press.
- J. Ruppenhofer, M. Ellsworth, M.R.L. Petruck, C.R. Johnson, and J. Scheffczyk. 2006. FrameNet II: Extended Theory and Practice. *Berkeley FrameNet Release*, 1.
- Szu-ting Yi, Edward Loper, and Martha Palmer. 2007. Can semantic roles generalize across genres? In *Proceedings of HLT-NAACL 2007*, pages 548–555.