Normalization and Similarity Recognition of

Complex Predicate Phrases

Based on Linguistically-Motivated Evidence

Tomoko Izumi

January 2014

ABSTRACT

The need for deep semantic understanding of language is growing rapidly due to the increasing amount of textual information being stored, including blogs, tweets, and spoken dialogue data. Among various linguistic expressions, understanding *predicate phrases* is especially important because predicates express the propositional meaning of a sentence. For example, predicate expressions such as *can't install*, *would like to buy* and *don't know* express complaints, compliments, and questions, which all convey crucial information for systems such as opinion mining and automatic dialogue/QA systems. Similarly, identifying semantically similar predicate phrases such as "*consumes* memory" and "*eats* memory" can drastically improve the system performance in various applications including information retrieval, text mining etc.

While predicates provide valuable information, it is difficult to correctly capture predicate meaning. This is due to the variations in predicate expressions. Predicates are multi-word/morpheme expressions, and unlike nouns, they can convey not only information about an event but also discourse information, such as politeness. For example, the meaning of "want to buy" can be expressed by "*wanna buy*," "*would like to make a purchase*," and sometimes simply by "*want to get*."

In order to correctly understand the meaning of predicates, a system needs to deal with these surface variations, which can be divided into morphological variations, syntactic variations, and semantic variations. First, in agglutinating languages such as Japanese and Korean, predicates appear in a concatenated string of different morphemes, which express tense, modality, negation and discourse information (morphological variations). Second, several predicates appear in the complex form of Light Verb Constructions (LVC), such as *give a try* (syntactic variations). Last,

predicate meanings are polysemous and their meanings differ depending on context (semantic variations). These variations are all related to the linguistic properties of predicates, so simply ignoring them triggers the extraction of erroneous predicate meaning.

In this thesis, we propose the normalization and similarity recognition of complex predicate phrases based on linguistically-motivated evidence. Chapter 1 starts by introducing this thesis and summarizes the problems that occur when interpreting predicate phrases as part of natural language processing.

Focusing on morphological variations, Chapter 2 introduces a novel normalization technique that paraphrases complex functional expressions into simplified forms that retain just the crucial meaning of the predicate. The paraphrasing rules that result are based on linguistic theories in syntax and semantics, and achieve the high accuracy of 79.7% while the differences in functional expressions are reduced by up to 66.7%.

Chapter 3 discusses syntactic variations in predicates, namely Light Verb Constructions. An analysis of the linguistic properties of light verbs allows us to create paraphrasing patterns that map 151 different light verbs into 10 simple forms. Of these 10 forms, 7 convert complex noun-particle-verb structures into simple predicative forms. By constructing a list of 923 examples for ambiguous light verbs, we show that we can correctly distinguish real LVCs from those in which the light verbs were actually functioning as a main verb. The results of experiments indicate that our paraphrasing rules offer high accuracy. Furthermore, both normalization techniques for functional expressions and light verb expressions provide the promising result of effectiveness in the predicate extraction task examined here, a simple text mining application.

Chapter 4 introduces the most challenging task of solving semantic variations in different predicates, namely identifying semantically similar predicates phrases. Using different linguistic levels of features for recognizing synonymous and antonymous predicates, we succeed in identifying predicate phrases, even those that are synonymous even in certain contexts, with the high F-score of 0.87. Moreover, the proposed method shows the promising result of automatically obtaining semantically similar predicate phrases from raw text corpora, indicating the possibility of the automatic construction of predicate thesauri, a resource essential for understanding the meaning of different predicates.

By normalizing morphological and syntactic variations in complex predicates and automatically recognizing semantically similar predicate phrases, we make it possible to understand various predicate expressions based on their similarity in meaning. We believe that this will improve the overall performance of natural language processing tasks on diverse textual data, the emerging source of valuable knowledge.

ACKNOWLEDGMENTS

I would like to express my special gratitude to Prof. Sadao Kurohashi, Dr. Tomohide Shibata, and Dr. Daisuke Kawahara. Particularly, I would like to thank Prof. Kurohashi for his helpful suggestions, thorough feedback, and continuous encouragement on completing my PhD project, and Dr. Shibata for his generous support, invaluable advice, and warm encouragement throughout my graduate work at Kyoto University.

I would like to express my appreciation to the members of my PhD committee, Prof. Toru Ishida and Prof. Tatsuya Kawahara, for their insightful comments and suggestions. I am also grateful to Dr. Kenji Imamura, Prof. Genichiro Kikui, Prof. Satoshi Sato, and Dr. Atsushi Fujita who have given me valuable advice since my first year in the field of engineering.

I owe a special thanks to my supervisors at NTT, Kuniko Saito, Hisako Asano, and Yoshihiro Matsuo, for their continuous encouragement and support for the completion of my graduate work. I would also like to thank all the members at Kurohashi-Kawahara Lab. for their warm encouragement throughout my graduate work, and Mike Blackburn for his outstanding help with my PhD thesis.

Finally, I would like to express my deepest gratitude to my mentor, Prof. Yoshifumi Sato, to my parents, Susumu Izumi and Etsuko Izumi, and to my husband, Takashi Mitsui who always supported and encouraged me to complete my graduate work at Kyoto University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- ACC Accusative case
- BL Baseline
- c Content word
- COMP Completive aspect
- CONT Continuous aspect
- COP Copular verb
- F F-score
- f Function word
- Funcs Functional expressions
- LCS Lexical concept structures
- LVC Light verb construction
- LV Light verb
- NLP Natural language processing
- NOM Nominative case
- P Particle
- PAtr Predicate attributes
- Prec Precision
- Pred Predicate
- Rec Recall
- SemLabels Semantic labels
- SFP Sentence final particles

CHAPTER 1

INTRODUCTION

1.1. Need for Deep Semantic Understanding of Predicate Phrases

Demand is growing for the deep semantic understanding of language so as to handle and utilize the increasing amount of textual information that is becoming accessible, such as newspapers, blogs, tweets, and spoken dialogue data. Among various linguistic expressions that need to be understood, *predicate phrases* are especially important because predicates express the propositional meaning of a sentence, the essential part that describes *what is happening* in a text. For example, predicate expressions such as *can't install*, *would like to buy* and *don't know* express complaints, compliments, and questions, which all convey crucial information for systems such as opinion mining and automatic dialogue/QA systems.

In order to fully understand the meaning of these diverse predicate phrases, the most fundamental task for Natural Language Processing (NLP) is to recognize their similarity in meaning (recognition of synonymous predicates). In information retrieval, identifying semantically similar predicate phrases such as the pair "*consumes* memory" and "*eats* memory" is crucial to improve overall system performance, while in text mining, summing up expressions with the same meaning drastically affects the overall quality of text analysis.

Although predicates provide valuable information, it is difficult to correctly identify their semantic similarity. This is due to variations in predicate expressions. Predicates are multi-word/morpheme expressions, and unlike nouns, they can convey not only information about an event (i.e., *what is happening*) but also discourse

information, such as politeness. For example, the meaning of "want to buy" can be expressed by "*wanna buy*," "*would like to make a purchase*," and sometimes simply by "*want to get*." Simply using the surface strings would only result in a great number of predicate expressions with low frequencies while only using the head word of a predicate (e.g., *buy* and *get*), the approach adopted by many current text mining and information retrieval systems, would yield the extraction of false predicate meaning.

In order to construct an algorithm that can understand the semantic similarity of different predicates, it is essential to analyze the linguistic properties of predicates because each element in a predicate has its own linguistic function. In linguistics, predicates are claimed to express the meaning of an event (Portner, 2005). The meaning of an event is further elaborated by elements such as tense (aspect), modality, and negation. Tense (aspect) expresses the time in (at/for) which an event occurs, such as *went* for past tense. Modality affects the factuality status of an event (Narrog, 2005), such as *might* and *must* in "might occur" and "must occur." Negation reverses the value of an event (e.g., *did not go*). Discourse information such as politeness can be also expressed in predicates (e.g., *wanna* vs. *would like to*).

In actual NLP applications, we claim that sustaining the predicate meaning of "what is happening" is crucial. That is, identifying when the event occurred/occurs (tense), and making a clear distinction of whether the event indeed occurred/occurs (negation and modality) are crucial. A similar claim is made in Inui et al. (2008) in which they emphasize the importance of determining whether what is written is factual or not.

In this thesis, we construct algorithms for identifying semantically similar predicate phrases based on linguistically-motivated evidence, meaning the rules and features applied are based on a theoretically-sound analysis of linguistics. Our

proposed methods consist of *normalization* and *similarity recognition.* By normalization, we mean reducing differences in surface forms of predicates while sustaining their meaning. By similarity recognition, we mean automatically identifying whether different predicates express the same event with the same factuality status.

The normalization and similarity recognition proposed in this thesis are related to three fundamental aspects of predicate variations, namely morphological variations, syntactic variations, and semantic variations, all of which explain why NLP systems have difficulty in analyzing predicate meaning. First, predicates produce a great number of morphological variants (morphological variations). Second, several predicates appear in syntactically complex structures (syntactic variations). Lastly and most challengingly, predicate phrases are polysemous, having multiple meanings, and different predicates can express the same meaning depending on context (semantic variations). By solving these problems, we make it possible to understand the widest variety of predicate expressions.

1.2. Morphological Variations of Functional Expressions

Predicate phrases are multi-word/morpheme expressions. They are constructed by a combination of a content word (e.g., *buy, purchase, get*), which provides the propositional meaning of an event, and functional expressions (e.g., *want to, would like to, wanna*), which add information of how the event is perceived by the speaker (Narrog, 2005; Portner, 2005). In agglutinating languages such as Japanese and Korean, a sequence of different morphemes forms a *functional expression*. The following is an example (NOM for nominalizer).

(1) kaiyakushi -tai -n -desu -kedo

cancel -want -NOM -COP -but

"want to cancel."

The propositional meaning is expressed by the head *kaiyakushi* "cancel" while the speaker's desire is expressed by *tai* "want." The sequence of morphemes *n-desu-kedo* merely conveys a discourse hedge. The same meaning can be expressed by the following predicates, in which only the functional expressions vary (COMP for a completive aspect marker)

(2) kaiyakushi -chai -tai -no -desu -ga

cancel COMP want NOM COP but

(3) kaiyakushi -tee -n -da -kedo

cancel -want -NOM -COP -but

The morpheme strings of *tai-n-desu-kedo*, *chai-tai-no-desu-ga* and *tee-n-da-kedo* all express a desire to cancel. As shown, in Japanese, functional expressions produce a great number of morphological variants, some of which convey important information about an event while others simply convey discourse information.

Several studies attempted the semantic understanding of functional expressions. Matsuyoshi and Sato (2008) paraphrase functional expressions using a hierarchically organized dictionary (Matsuyoshi et al., 2007). Shudo et al. construct rewriting rules of modality expressions, such as *beki-dewa-nai* "should not."

However, once we focus on applications such as text mining, we need a broader sense of "semantic similarity" than paraphrasing. That is, various expressions need to be grouped if they express the same event; information unrelated to the eventual meaning such as discourse politeness should be ignored. Previous studies on paraphrasing functional expressions (e.g., Matsuyoshi and Sato, 2008) kept even

discourse level information. This is useful for many applications including language generations but prevents the morphological variations in functional expressions from being reduced.

In order to reduce these morphological variations, this thesis proposes normalization of functional expressions in Japanese predicates (Chapter 2). Following the truth-value approach of an *event* denoted by predicates in the field of formal semantics (e.g., Chierchia and Mcconnel-Ginet, 2000; Portner, 2005), we categorize functional expressions into those that affect the meaning of an *event* and those that are merely used for discourse purposes. By deleting unnecessary functional expressions, we succeed in simplifying predicates while retaining the meaning of the event expressed by the predicate. This normalization of functional expressions is found to improve the performance of the *biased predicate extraction task*, which is examined here as a simple text mining application.

1.3. Syntactic Variations of Complex Predicates

Several predicates are expressed by a more complex form of light verb construction (LVC; Oku, 1990; Muraki, 1991; Stevenson et al., 2004), such as *give a try* and *make a cough.* In LVC, the meaning of the predicate is conveyed by the noun, and the verb itself simply functions as the verbalization of the noun. The following is an example of a Japanese LVC (ACC stands for the accusative case marker).

(4) kaiyaku -o -okonai -tai -no -desu -ga
cancellation -ACC -conduct -want -NOM -COP -but
"want to make a cancellation"

In LVC, recognizing the noun that is the head of the predicate is crucial because the verb itself does not convey propositional meaning. The verb is there to verbalize the

noun. Some light verbs also provide information such as passive voice, so correctly capturing the function of light verbs is also needed.

Furthermore, several light verbs show ambiguity, functioning as a light verb only when combined with a certain noun. An example is "give" in *give a try* and *give a copy,* the *give* in the former verbalizes *try* while the latter functions as a main verb. We need to correctly disambiguate light verbs from main verbs when understanding the meaning of a predicate.

Several studies tackled the paraphrasing of LVCs. Oku (1990) normalized Japanese LVCs into simplified verbal predicates as a preprocessing step for machine translation. The study aimed at improving translation quality, so translatable expressions are often not the target of normalization regardless of their similarities in meaning. Fujita et al. (2004) paraphrase LVC sentences by making the LVC into a simplified verbal predicate and transforming the case information of arguments based on a dictionary of lexical conceptual structure (LCS: Jackendoff, 1992; Takeuchi et al., 2006); however, the overage of the dictionary was insufficient. In order to fully handle the syntactic variations of LVCs, one needs a language resource that can cover various LVC structures including those with ambiguity.

In order to solve these problems raised by LVCs, this thesis proposes a paraphrase of LVC into a simplified verbal expression (Chapter 3). Based on a thorough analysis of linguistic properties in Japanese LVCs, we construct 10 paraphrasing patterns that normalize the complex structures of LVC into simplified verbal predicates. We use a large collection of blogs and newspapers to construct a comprehensive dictionary of noun-verb pairs for LVC disambiguation. By correctly disambiguating LVCs from regular verb-object structures, we succeed in reducing the syntactic variations of LVCs and improve the recall of predicate extraction.

1.4. Semantic Variations in Predicate Meanings

Predicates are often polysemous, and thus show a great number of semantic ambiguities. This is the reason why some of the predicates become synonymous only in a certain context (e.g., "break" and "ignore" in *break the rule* and *ignore the rule*). Unlike functional expressions or light verb constructions, which are consisted of closed class words, semantic ambiguities in predicates are too diverse to permit rule-based methods to be successful. In order to deal with the polysemous nature of predicates, statistically based metrics have been introduced. One of the commonly used methods is a distributional similarity measure (Curran, 2004; Dagan et al., 1999; Lin, 1998; Shibata and Kurohashi, 2010; Yih & Qazvinian, 2012).

Shibata and Kurohashi (2010) construct a vector model from a gigantic web corpus of 69 billion sentences in order to calculate the similarities between predicate-argument structures, such as *keiki-ga-hiekomu* "business gets cold feet" vs. *keiki-ga-akka* "business gets worse." Mitchell and Lapata (2010) propose composition based vector models in order to measure the similarity of phrases including object-verb structures such as *require-attention* vs. *need-treatment*.

The most notable feature of any distributional similarity metric is its unsupervised nature. It can construct vector models from raw corpora and there is no need for constructing human annotated data and/or language resources. However, as has been pointed out by several studies (Lin et al., 2003; Shibata & Kurohashi, 2010; Yih et al., 2012), a similarity based metric tends to simply indicate semantic relatedness; that is, it assigns high scores to words not only in synonym relations but also other semantic relations such as antonyms, associations etc. In order to make a clear distinction between semantically similar phrases and semantically *opposite* or *associated* phrases, one needs a finer grained algorithm.

In order to correctly identify semantically similar predicate phrases, this thesis proposes the automatic recognition of semantically similar predicate phrases based on linguistically-motivated features (Chapter 4). By combining an argument with a predicate we construct an algorithm that can recognize synonymous predicate-argument pairs, including those that become synonymous only in certain contexts. By analyzing linguistic evidence for synonymous predicates, we combine distributional similarity measures and different linguistic features extracted from definition sentences, predicate attributes, and functional expressions. Furthermore, we add linguistic properties that are peculiar to antonym relations and succeed in correctly distinguishing semantically similar predicates (synonymous predicates) from semantically *opposite* predicates (antonymous predicates). Using the algorithm, we also conduct the task of extracting synonymous predicate-argument pairs from a blog corpus and show the promising result of the automatic determination of semantically similar predicate phrases.

1.5. Contributions of Thesis

This thesis proposes a novel approach to identifying semantically similar predicate phrases in Japanese based on linguistically-motivated evidence. The contributions of this thesis are divided into two parts; *normalization* of complex predicate structures and *similarity recognition* of different predicates. The normalization simplifies predicate expressions while retaining the crucial meaning of predicates. The underlying principles of normalization rules are based on a solid linguistic analysis of syntax and semantics, which can be said universal. Our normalization rules themselves are applicable to not only Japanese but also other agglutinating languages such as Korean.

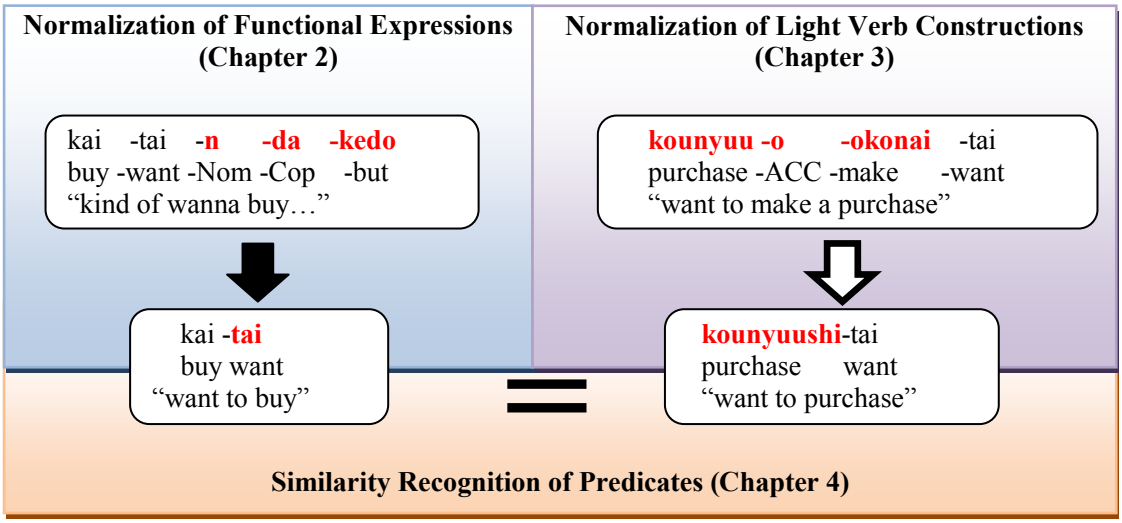

Figure 1.1. *Normalization and similarity recognition of predicates*

The similarity recognition proposal automatically classifies whether two different predicates express the same *event*. The linguistically-motivated features described herein reflect universal properties of synonymous predicates from different linguistic levels, ranging from lexical-encyclopedic level to discourse level, and can be applied to various languages. The normalization and similarity recognition proposed in this thesis are depicted in Figure 1.1. By normalizing morphological and syntactic variations in complex predicates and automatically recognizing semantically similar predicate phrases, we make it possible to understand various predicate expressions based on their similarity in meaning. We believe that this will improve the overall performance of NLP tasks on diverse textual data, the dominant emerging source of valuable knowledge.

The remainder of this thesis is organized as follows. Chapter 2 proposes the normalization of functional expressions in Japanese predicates. Chapter 3 proposes the paraphrasing of LVCs into simplified verbal expressions. Chapter 4 proposes the

supervised classification of synonymous predicates based on linguistically-motivated evidence. Chapter 5 concludes the thesis.

CHAPTER 2

NORMALIZING COMPLEX FUNCTIONAL EXPRESSIONS IN JAPANESE PREDICATES: LINGUISTICALLY-DIRECTED RULE-BASED PARAPHRASING AND ITS APPLICATION

2.1. Background

The need for text mining systems such as opinion mining and sentiment analysis is growing rapidly due to the increasing amount of textual information, especially consumer generated media and call center data. These systems must offer deep semantic analysis of the target language. Nasukawa (2009) claims that in the domain of Voice Of Customer (VOC) analysis, a practical system must be able to distinguish among complaints, compliments, and questions. Inui et al. (2008) emphasize the importance of determining whether what is written is factual or not in the domain of experience mining (Factuality Analysis). These studies often claim that the simple bag-of-word technique is insufficient for deep semantic analysis (Inui et al., 2008, p. 318; Nasukawa, 2009, p. 1).

For example, in the domain of VOC analysis, expressions such as *would like to* as in "*would like to buy"* and *can't* as in *"can't install"* are key expressions in detecting the customer's needs and complaints, and extracting only a single content word ("buy" and "install" in the example above) would fail to capture this information.

Similarly, recognizing tense marking and the existence of a modal auxiliary can be crucial to detecting whether what is described by the predicate actually happened or is simply supposed (e.g., purchas*ed* vs. *might* purchase). In order to extract subtle but semantically essential information from the input text, it is crucial to consider both the content word *and* the functional expressions (Nasukawa, 2001).

Few studies have dealt extensively with functional expressions for use in Japanese Natural Language Processing (NLP) systems (e.g., Tanabe et al., 2001; Matsuyoshi and Sato, 2006, 2008). This is due to the fact that functional expressions such as *would like to* and *might have been* are syntactically complicated and semantically abstract and so are poorly handled by NLP systems. For example, the expression *would like to* has meaning similar to *want to* and *wanna*, but the system cannot recognize their similarities from just the surface forms. This is especially the case in Japanese.

In Japanese, functional expressions appear in the form of suffixes or auxiliary verbs that follow the content word. This sequence of a content word (*c* for short) plus several functional expressions (*f* for short) forms a *predicate* in Japanese (COMP for completive aspect marker, NOM for nominalizer, COP for copular verb).

(5)	kat	-tyai	-takat	-ta	-n	-da
	buy	-COMP	-want	-PAST	-NOM	-COP
	c	$-f_1$	$-f_2$	$-f_3$	$-f_4$	$-f_5$

"(I) wanted to buy (it)."

The meaning of "want to" is expressed by -*takat* (f_2) and the past tense is expressed by -*ta* (f_3). The other functional expressions, -*tyai* (f_1), -*n* (f_4), and -*da* (f_5), only slightly

alter the formality of expressing "wanted to buy," as there is no direct English translation. Rather, these expressions are used for discourse purposes, such as emphasizing the action itself and wrapping (or softening) the speaker's comment (Maynard, 1997; Tsujimura, 2007). Therefore, (1) expresses the same fact as (6).

(6) kai -takat -ta
buy -want -PAST
"(I) wanted to buy (it)."

As shown above, in Japanese, sentential predicates are multi-morpheme expressions that consist of two different types of functional expressions; one influences the factual meaning of the predicate (f_2 and f_3) while the other is merely used for discourse purposes and does not alter the factual status of the predicate (f_1, f_4 and f_5). Once one extracts a predicate phrase with functional expressions, however, the number of differences in surface forms increases drastically regardless of their similarities in meaning as shown in (1) and (2). This increase in surface forms complicates NLP systems, especially information extraction systems including text mining, because they are unable to recognize that these seemingly different predicates actually express the same *fact.*

In this chapter, we introduce a novel *normalization* technique that paraphrases complex functional expressions in Japanese into simplified natural language forms. The term *normalize* is used here to refer to the procedure of unifying the surface variations of predicates that express similar meanings. By focusing on the extraction of predicates that express the same *event*, we define functional expressions that influence *the factuality of predicative meaning*. The normalization system reduces the

differences in surface forms of predicates while retaining the factual status of the information. This is made possible by our linguistically-directed paraphrasing rules.

This chapter is organized as follows. In Section 2.2, we provide related work on Japanese functional expressions in NLP systems as well as problems that need to be solved. Section 2.3 introduces the linguistic theories on which our paraphrasing rules are constructed. Section 2.4 describes the experiments conducted on our normalization system. Section 2.5 discusses the results and applicability of our normalization system to text mining systems. The last section is the conclusion. Throughout this thesis, we use the term *functional expression* to indicate not only a single function word but also compounds (e.g., *would like to*).

2.2. Related Works on Functional Expressions and Problems

Shudo et al. (2004) construct semantic rules for functional expressions and use them in order to find whether two different predicates have the same meaning. Matsuyoshi et al. (2006, 2007) and Matsuyoshi and Sato (2008) construct an exhaustive dictionary of functional expressions, which are hierarchically organized, and use it to generate paraphrases of functional expressions.

Although these studies provide useful insights and resources for NLP systems, if the intention is to extract and group predicates expressing the same event, we find there are still problems that need to be solved. We focus here on the following two key problems.

The first problem is that many functional expressions are unnecessary for deciding the factuality of a predicate.

(7) yabure -tesimat -ta -no -dearu

rip -COMP -PAST -NOM -COP

c $-f_1$ $-f_2$ $-f_3$ $-f_4$

"(something) ripped."

(7) can be simply paraphrased as (8)

(8) yabure -ta

rip -PAST

c $-f_1$

In actual NLP applications such as text mining, it is essential that the system recognizes that (7) and (8) express the same event of something "*ripped*." In order to achieve this, the system needs to recognize *-tesimat, -no,* and *-dearu* as unnecessary (f_1, f_3, $f_4 \rightarrow \emptyset$). Previous studies that focus on the paraphrasing of one functional expression to another ($f \rightarrow f'$) cannot solve this problem.

The second problem is that we sometimes need to *add* certain functional expressions in order to retain the meaning of a predicate ($\emptyset \rightarrow f$).

(9) (Hawai-ni) $\underline{_{P1}\text{iki}}$, $\underline{_{P2}\text{nonbirisi}}$ $\underline{\text{-takat}}$ $\underline{\text{-ta}}$

(Hawaii-to) go relax -want -PAST

c_1 c_2 $-f_1$ $-f_2$

"I wanted to go to Hawaii and relax."

Example (9) is in a coordinate structure, and two verbal predicates, *iki* (P1) "go" and *nonbirisi-takat-ta* (P2) "wanted to relax," are coordinated.

As the English translation indicates, the first predicate has the factual meaning of *iki-takat-ta* "wanted to go," which implies that the speaker was *not* able to go to Hawaii. If the first predicate was extracted and analyzed as *iku,* the base (present) form of "go," then this would result in faulty predicate extraction, indicating the erroneous fact of going to Hawaii in the future (present tense in Japanese expresses a future event). In this case, we need to *add* the functional expressions *takat* "want" and *ta* (the past tense marker), to the first verbal predicate.

As shown above, there are two problems that need to be solved.

- Several functional expressions are necessary for sustaining the meaning of the *event* expressed by a predicate while others barely alter the meaning ($f \rightarrow \emptyset$).
- Several predicates in coordinate structures lack necessary functional expressions at the surface level ($\emptyset \rightarrow f$). This results in an incorrect interpretation of the predicate meaning if only the surface form of the predicate is extracted.

These problems, caused by variations in functional expressions, have also been reported in the field of machine translation. Shirai et al. (1993) constructed rewriting rules to convert Japanese functional expressions that express tense and modality into pseudo-linguistic forms that are translated into English. However, the expressions that they covered are very limited (only expressions such as *yotei-da* "be planning to" and *tokoro-da* "be going to" were included in the rules.). Oku (1990) also constructed a rule to rewrite Japanese predicate phrases into simplified forms. However, he only treats light verb constructions, such as *sihai-o-ukeru* "to be under control (literally,

receive control)," and provides no rule to simplify functional expressions in predicate phrases. As shown, neither Oku (1990) nor Shirai et al. (1993) constructed rewriting rules that broadly cover different types of functional expressions, including those that are merely used for discourse purposes. This is because the two studies target newspaper articles, which tend to have fewer variations in functional expressions than blogs and conversational style texts.

On the other hand, the study by Lee et al. (2006) translated Korean conversational style texts into English. They constructed a rule that deleted Korean function words that were untranslatable into English before translation and achieved an improvement in overall translation quality. However, they simply used POS tags in deciding which expressions to delete, so it is not clear whether crucial information was properly sustained. Furthermore, they failed to offer any rules that could add necessary functional expressions to intermediate predicates although Korean shows a similar tendency of lacking necessary functional expressions in a coordinate structure (Yoon (1994)). This would result in a serious loss in the information needed for correct English translation.

As shown above, although there are studies that focus on paraphrasing functional expressions, they still leave many issues unanswered. In this study, based on syntactic and semantic theories in linguistics, we construct paraphrasing rules that broadly cover various functional expressions in Japanese predicates and solve the problems by *normalizing* complex functional expressions into syntactically simple but semantically rich forms.

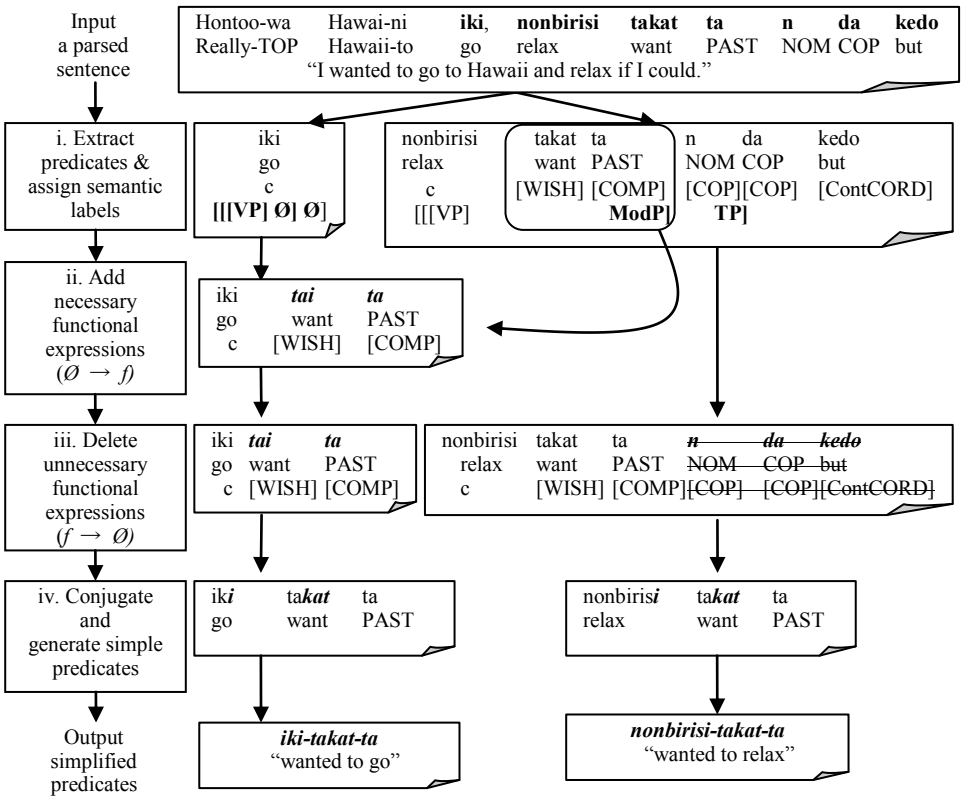

Figure 2.1. *Flow of normalization.*

2.3. Construction of Paraphrasing Rules

The overall flow of our normalizing system is depicted in Figure 2.1. The system works as follows.

i. Given a parsed sentence as an input, it extracts a predicate(s) and assigns a semantic label to each functional expression based on Matsuyoshi et al. (2006, 2007).

ii. As for an intermediate predicate, necessary functional expressions are added if missing (*Ø →f)*.

iii. From each predicate, delete unnecessary functional expressions that do not alter the factual meaning of the predicate (*f →Ø*).

iv. Conjugate each element and generate a simplified predicate.

There are two fundamental questions that we need to answer to accomplish this system.

A) *What are UNNECCESARY functional expressions (at least for NLP applications)*, *i.e., which ones do not alter the meaning of the event expressed by a predicate?*

B) *How do we know which functional expressions are missing and so should be added?*

We answer these questions by combining what is needed in our NLP applications and what is discussed in linguistic theories. We first answer Question A.

2.3.1. Categorization of Functional Expressions

As discussed in Section 1 and in Inui et al. (2008), actual NLP applications must be able to recognize whether two seemingly different predicates express the same *fact*. This emphasis on factuality is similar to the truth-value approach of an *event* denoted by predicates as discussed in the field of formal semantics (e.g., Chierchia and Mcconnel-Ginet, 2000; Portner, 2005). Although an extensive investigation of these theories is beyond the scope of this paper, one can see that expressions such as *tense (aspect)*, *negation* as well as *modality*, are often discussed in relation to the meaning of an *event* (Partee et al., 1990; Portner, 2005; Narrog, 2005).

- **Tense (Aspect):** Expresses the time in (at/for) which an event occurred.[1]
- **Negation:** Reverses the truth-value of an event.
- **Modality**: Provides information such as possibility, obligation, and the speaker's eagerness with regard to an event and relate it to what is true in reality.

The above three categories are indeed useful in explaining the examples discussed above.

(10)	kat	-tyai	-takat	-ta	-n	-da
	buy	-COMP	-want	-PAST	-NOM	-COP
		aspect	*modality*	*tense (aspect)*		

(11)	kai	-takat	-ta
	buy	-want	-PAST
		modality	*tense (aspect)*

"wanted to buy"

The predicate "*kat-tyai-takat-ta-n-da*" in (10) and "*kai-takat-ta*" in (11) express the same event because they share the same tense (past), negation (none), and modality (want). Although (10) has the completive aspect marker *-tyai* while (11) does not, they still express the same fact. This is because the Japanese past tense marker *-ta* also

1 Throughout this paper, we treat Tense and Aspect as one because Japanese only has the past tense marker *ta* which also functions as an aspect marker (Nakau, 1976; Tsujimura, 2007).

expresses the completive aspect. The information expressed by *-tyai* in (10) is redundant and so unnecessary.

On the other hand, the predicate "*iku*" in (5) and "*iki-takat-ta*," which conveys the actual meaning of the predicate, express a different fact because they establish a different tense (present vs. past) and different modality (none vs. want).

As shown, once we examine the semantic functions of functional expressions, we can see that the factual information in a predicate is influenced by tense (aspect), negation, and modality. Therefore, the answer to Question A is that the necessary functional expressions are those that belong to *tense (aspect), negation,* and *modality*. Furthermore, if there are several functional expressions that have the same semantic function, retaining one of them is sufficient.

2.3.2. Adding Necessary Functional Expressions

The other question that we need to answer is how we can find which functional expressions are missing when normalizing predicates in a coordinate structure (e.g., (9)). This shortfall occurs when a predicate appears in the middle of a sentence (henceforth, *intermediate predicates*). We solve this based on a detailed analysis of the syntactic structure of predicates.

In coordinate structures, several *equivalent* phrases are coordinated by conjunctions such as *and, but,* and *or*. If a predicate is coordinated with another predicate, these two predicates must share the same syntactic level. Therefore, the structure in (5) is depicted as follows (What TP and ModP stand for will be discussed later).

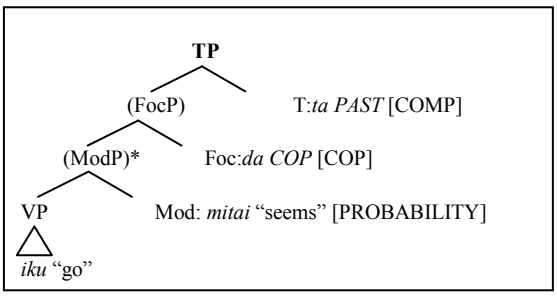

Figure 2.2: *Structure of a predicate.*

[$_{TP}$	[$_{ModP}$	[**$_{VP}$**	(Hawai-ni) iki]	[**$_{VP}$**	nonbirisi]	takat]	ta]
[$_{TP}$	[$_{ModP}$	[$_{VP}$	(Hawaii-to) go]	[$_{VP}$	relax]	want]	PAST]

This is the reason why the first predicate *iki* should be paraphrased as *iki-takat-ta* "wanted to go." It needs to be tagged with the modality expression *-takat* "want to" and the past tense marker *-ta*, which seem to be attached to only the last predicate.

This procedure of adding necessary functional expressions to the intermediate predicate is not as simple as it seems, however.

(12) $_{P1}$nemutai -mitai -de $_{P2}$kaeri -tagat -tei -ta

sleepy seems -COP go home want CONT PAST

"He seemed sleepy and wanted to go home."

In (12), the first predicate *nemutai-mitai-de* "seem to be sleepy" should be paraphrased as *nemutai-mitai-dat-ta*, "seem*ed* to be sleepy," in which only the functional expression indicating *past* is required. The other functional expressions such as *tagat* "want," and the aspect marker *te-i* (CONTinuation) should not be added (*nemutai-mitai-de-tagat* (want)-*te-i* (CONT)-*ta* (PAST) is completely ungrammatical).

Furthermore, the intermediate predicate in the following example does not allow any functional expressions to be added.

(13) (imawa) P1yasui-ga (mukasiwa) P2takakat -ta
(today) inexpensive-but (in old days) expensive PAST
"(They) are inexpensive (today), (but) used to be very expensive (in the old days.)"

In (13), the first predicate *yasui* "inexpensive" should not be paraphrased as *yasukat-ta* "was inexpensive" since this would result in the ungrammatical predicate of "* (they) *were* inexpensive (today)."

As shown in (8) and (9), in order to add necessary functional expressions to an intermediate predicate, one needs to solve the following problem.

- *Which functional expressions should be added to which intermediate predicate*?

We address this problem by turning to the *incompleteness* of the syntactic structure of a predicate.

Studies such as Rizzi (1999) and Cinque (2006) proposed detailed functional phrases such as a TopP, *Topic Phrase*, in order to fully describe the syntactic structures of a language. We adopt this idea and construct a phrase structure of Japanese predicates that borrows from the functional phrases of *Tense Phrase* (TP), *Modality Phrase* (ModP), and *Focus Phrase* (FocP) (Figure 2.2).

ModP, *Modality Phrase*, is where modality expressions can appear.[2] FocP, *Focus Phrase*, is the phrase where the copula *da* appears. This phrase is needed because several modality expressions syntactically need the copula *da* in either the following or preceding position (Kato, 2007). The existence of FocP also indicates that the modality expressions within the phrase are complete (no more modality phrase is attached). TP, *Tense Phrase*, is where the tense marker appears.[3]

As discussed, we assume that predicates are coordinated at one of the functional phrase levels in Figure 2.2. Functional expressions that need to be added are, therefore, those of the *outer* phrases of the target phrase.

For example, if the target phrase has *da*, the head of FocP, then it only needs the past tense marker to be added, which is located above the FocP (i.e., TP). This explains the paraphrasing pattern of (8). Therefore, by looking at which functional expressions are held by the target predicate, one can see that the functional expressions to be added are those that belong to phrases above the target phrase.

Furthermore, as is shown in Figure 2.2, the predicate can be said to be complete if there is a TP. This is because, as often described in syntactic theories (e.g., Adger, 2003), a sentence can be said to be a phrase with tense (i.e., TP). In other words, if a predicate has tense, it can stand alone as a sentence.

---

[2] The structure of Figure 2.2. is recursive. A modality expression can appear after a TP. Also, more than one ModP can appear although ModP and FocP are optional.

[3] Note that this structure is constructed for the purpose of Normalization; other functional projections such as NegP (negation phrase) will not be discussed although we assume they must exist.

2.3.3. Implementing Normalization

In this final subsection, we describe how we actually implemented our theoretical observations in our normalization system.

CATEGORIZE functional expressions

First, we divided functional expressions listed in Matsuyoshi et al. (2006, 2007), yielding a total of about 17,000 functional expressions with 96 different semantic labels,[4] into those that belong to our syntactic and semantic categories and those that do not. The semantic labels in Matsuyoshi et al. (2006, 2007) were constructed in order to group semantically similar expressions. For example, the functional expressions that show the speaker's wish were labeled as *ganbou* "wish" while the functional expressions that express the completive aspect were labeled as *kanryo* "completion." For those that belong to our syntactic and semantic categories, appropriate syntactic and semantic categories were assigned. The categorization used abstract semantic labels, such as "completion," "probability," and "wish."

Then, we further divided those that did not belong to our syntactic and semantic categories into *Deletables* and *Undeletables*. *Deletables* are those that do not alter the meaning of an event and are, therefore, unnecessary. *Undeletables* are those that are a part of the content words, and so cannot be deleted (e.g., *kurai* [degree] "about" as in *1-man-en-kurai-da* "is *about* ten thousand yen"). The results of our categorization are

[4] We added 7 semantic labels and 643 entries to Matsuyoshi et al. (2007) to increase the coverage of the dictionary for blog texts. We first measured the coverage of the original version of Matsuyoshi et al. (2007) against blog texts. We then selected functional expressions that often appeared in the texts but were not included in the original dictionary. We added frequently appeared expressions and increased the coverage from 95.3% to 97.1%.

detailed in Table 2.1. The assignment of semantic labels to functional expressions was conducted by Imamura et al. (2011)' s semantic label tagger, which selects the best sequence of semantic labels based on a discriminative model.

Based on the categorization of semantic labels as well as surface forms of functional expressions, our system works as follows;

ADD necessary functional expressions

A-1: Examine whether the intermediate predicate has the tense marker *ta*. If it lacks -*ta* and it is in the form of gerundive or in the form followed by the conjunction -*te* or -*de*, then go to Step A-2. Otherwise, go to D-1.

A-2: Based on the semantic label of the intermediate predicate, decide which level of syntactic phrase the predicate projects. Add functional expressions from the following predicate that belongs to outer phrases.

DELETE unnecessary functional expressions

D-1: Delete all functional expressions categorized as *Deletables*.

D-2: Leave only one functional expression if there are several identical semantic labels. For those categorized as *Negation*, however, delete all if the number of negations is even. Otherwise, leave one.

D-3: Delete those categorized as *Focus* if they do not follow or precede a functional expression categorized as *Modality*.

D-4: Check the syntactic necessity of *Focus* based on trigram scores and select the candidate with the highest score.

We added D-4 in order to judge the syntactic necessity of *Focus* because not all functional expressions categorized as *Modality* syntactically need *Focus.* We used

Table 2.1. *Syntactic and semantic categorization of semantic labels.*

Syntactic	Semantic	Semantic Labels
T if the surface is *ta*	*Tense (Aspect)*	completion, continuation, succession, simultaneity, avoidance, resultative (*teoku*-form), in the middle, leaving (as it is), tendency, trial, experience, habit, continuation (from), continuation (toward)
	Negation (Negated Modality)	negation, leaving (as it is), prohibition, inevitability, impossibility, improbability, meaninglessness, unneccessity, negated intention
Mod	*Modality*	probability, wish, interrogation, permission, obligation, intension, request, persuasion, invitation, possibility, comparison, obligation (in conjunctive form)
Foc	*Focus*	copular, nominalization, apposition
	Deletables	politeness, do a favor of (*kureru*-form), do a favor of (*ageru*-form), hearsay, balanced, resultative (*kotoninaru*-form), in addition to, reason, contrastive conjunction, interjection, grumble, coordinate conjunction, subordinate conjunction, unexpectedness, restrictive, excessive, continuation (from), continuation (toward), trial, experience, habit
	Undeletables	degree, endpoint, evidence, perspective (wa), perspective (mo), ratio, criterion, starting point, circumstance, state, disregarding, interrelation, target, instrument, definition, range, unrestricted, unbalanced, position, synchronous, restricted coordination, contrastive subordination, purpose, repetition, causative state, contrastive, appropriateness, situation, topic, parallel, recipient, goal, subjective, emphasis

SRILM (Stolcke, 2002) to construct a trigram model, which calculates trigram scores of candidates with and without *Focus*. The candidate with the highest score is selected. POS tags of each word and the surface form of each functional expression are used to calculate the trigram score.

GENERATE simple predicates

Last, conjugate all elements and generate simplified surface forms of predicates.

The above procedures of CATEGORIZE, ADD, and DELETE are our linguistically-directed paraphrasing.

By adopting this idea, we judge the completeness of a predicate by the existence of tense. If there is tense, then no functional expression has to be added because the predicate is syntactically complete. Because Japanese marks past tense by the past

tense marker -*ta*, we use -*ta* to detect the existence of tense. However, Japanese has no explicit *present* tense marker; the base form of a verb is also its present form. We solve this based on the coordinate conjunction that follows a predicate. As discussed in Minami (1993), the finite state and the type of conjunction are related; some conjunctions follow tensed phrases while others follow infinitival phrases. By investigating various coordinate conjunctions in Japanese, we find that predicates in gerundive form (also called *renyoukei*) and those coordinated by -*te/de* as in (9) and (12) cannot directly co-occur with the tense marker -*ta*, meaning they are tenseless. These are the predicates that are syntactically incomplete, so we add functional expressions that belong to the outer phrases of the target predicate as in (9) and (12).

As shown, the answer to Question B is that we only add functional expressions to *incomplete* predicates, where judgment is based on the existence/absence of tense. The appropriate functional expressions to be added are those of outer phrases of the target phrase.

2.4. Experiments of Normalizing Functional Expressions

In order to evaluate the accuracy of our paraphrasing rules as well as the impact of our normalization on text mining system performance, we conducted three experiments. Expriment 2.1 measures the paraphrasing accuracies against human-annotated data. Experiment 2.2 measures the rate of reduction in surface differences obtained by our normalization system. Experiment 2.3 examines the impact of our normalization system on a text mining application.

2.4.1. Experiment 2.1: Evaluating Paraphrasing Accuracy

2.4.1.1. Constructing paraphrase data

We selected 2,000 sentences from newspaper and blog articles in which two or more predicates were coordinated.[5] We manually extracted predicates ($c_{1..}c_m$-f_1-f_2..f_n). Half of them were those in which the last predicate had *three* or more functional expressions ($n \geq 3$). We then asked one annotator with a linguistic background to paraphrase each predicate into the simplest form possible while retaining the meaning of the event.[6] We asked another annotator, who also has a background in linguistics, to check whether the paraphrased predicates made by the first annotator followed our criterion, and if not, resolve this discrepancy with the first annotator to settle on one correct paraphrase. 424 out of 4,939 predicates (8.5%) were judged by the 2nd annotator as *not following the criterion* and were re-paraphrased. This means that the accuracy of 91.5% is the gold-standard of our task. Out of 2,000 sentences in the data, 400 were used for constructing the rules as development sets (Closed) while 1,600 were used as test sets (Open).

2.4.1.2. Procedure

We evaluated the accuracy of our paraphrasing system as follows. First, we excluded instances that had tokenization errors and those that were judged as

---

[5] We use *Mainichi Newspapers* from the year 2000 and blog articles from September 2004 to December 2005, which were crawled from the web.

[6] We asked whether functional expressions should be deleted from or add to each predicate when paraphrasing. Only the surface forms (and not semantic labels) were used for annotation.

inappropriate as a predicate.[7] We also manually assigned correct semantic labels to these predicates. A total of 1,501 intermediate predicates (287 for development and 1,214 for test) and 1,958 last predicates (391 for development and 1,567 for test) were used in the evaluation of our paraphrasing rules.

The accuracy was measured based on *the exact match* in surface forms with the manually constructed paraphrases. For comparison, we used the following baseline methods. The last baseline method (Delete on POS) was used for comparing Lee et al. (2006)'s method of preprocessing Korean-English translation to our proposed method.

- No Add/Delete (BL 1): Do not add/delete any functional expression.
- Simp-Add (BL 2): Simply add *all* functional expressions that the intermediate predicate does not have from the following predicate.
- Delete on POS (BL 3): Delete functional expressions based on their POS tags. Following Lee et al. (2006), delete words with the POS tag *zyosi* "particles."[8]

---

[7] In Japanese, a gerundive form of a verb is sometimes used as a postposition. The annotators excluded these examples as "not-paraphrasable."

[8] Lee et al. (2006) deleted Korean function words that belong to case particles, auxiliary particles and final particles. Due to the differences between Korean and Japanese, we were not able to find POS tags that exactly match those of Lee et al. (2006). Therefore, we used the POS tags *zyosi* "particles." Function words with the POS "particles" include case particles, final and interjectory particles, binding particles, conjunctive particles, coordinate particles, adverbial particles, nominal particles, adverbializers and particles-of-special-function.

Table 2.2. *Result of experiment 2.1 (Rules Only).*

** p< 0.01

	Normalization	(BL1) No Add/Delete	(BL2) Simp Add	(BL3) Delete on POS
Open (Intermediate)	**77.5% (941/1214)****	**57.8% (702/1214)**	**32.8% (398/1214)**	**32.0% (389/1214)**
Closed (Intermediate)	82.6% (237/287)**	62.0% (178/287)	35.2% (101/287)	37.3% (107/287)
Open (Last)	**81.4% (1275/1567)****	**51.2% (802/1567)**	n.a	**35.7% (559/1567)**
Closed (Last)	87.5% (342/391)**	48.1% (188/391)	n.a.	37.9% (148/391)

Table 2.3. *Result of experiment 2.1 (Overall accuracies).*

Last (OPEN)	Overall Accuracy
Correct	**78.6%** (1257/1600)
Wrong	17.0% (273/1600)
PRED Extraction Error	4.4% (70/1600)

Intermediate (OPEN)	Overall Accuracy
Correct	**46.9%** (906/1931)
Wrong	12.1% (234/1931)
PRED Extraction Error	41.0% (791/1931)

2.4.1.3. Results of Experiment 2.1

Table 2.2 indicates the results. Our paraphrasing rules achieved the high accuracy of 77.5% (Intermediate (predicates)) and 81.4% (Last (predicates)) in Open (against the test set) and 82.6% (Intermediate) and 87.5% (Last) in Closed (against the development set). These values are quite high compared to the baseline methods (No Add/Delete (open), 57.8%; Simp-Add (open), 32.8%; Delete on POS (open), 32.0%). The differences between the proposed method and the baseline methods are all statistically significant (**$p < 0.01$).[9]

[9] We conducted a sign test to compare the results of our proposed method to those of the baseline methods.

We also measured the overall accuracy of our normalization system, which means all the procedures in Section 3.1 were automatically implemented. We used MeCab[10] as the tokenizer and POS tagger and CaboCha[11] as the dependency parser. For predicate extraction and semantic label tagging, we used Imamura et al. (2011)'s automatic predicate extractor and semantic label tagger, which extracts a predicate phrase and assigns one of the semantic labels of Matsuyoshi et al. (2006, 2007) to each functional expression of the predicate. The overall accuracy of Imamura et al. (2011) is 95.8%. The accuracy of extracting predicates and assigning correct semantic labels is 86.3%. In order to extract intermediate predicates in coordinate structures, we created heuristic rules based on dependency information.

The overall accuracy is listed in Table 2.3. Only the results against the test set (1,600 sentences) are shown. We divided the results into three categories; *PRED Extraction Error, Correct,* and *Wrong*. *PRED Extraction Error* indicates that the system failed to extract a target predicate, which is caused by Imamura et al. (2011)'s predicate extractor and our heuristic rules to extract intermediate predicates. *Correct* indicates that the automatically normalized functional expression matched the human annotation. *Wrong* indicates that the normalized forms output by the system and by the human annotators do not match at the surface level. The *Wrong* instances were caused by two factors; errors in our normalizing rules and errors in Imamura et al. (2011)'s semantic label tagger.

The overall accuracies were 78.6% for Last and 46.9% for Intermediate. The accuracy for Intermediate seems low. This is because we set a rather strict criterion for

[10] Mecab (http://mecab.sourceforge.net/)

[11] CaboCha (http://cabocha.sourceforge.net/)

extracting intermediate predicates in coordinate structures, resulting in the decrease in recall. The accuracy for Intermediate is lower in Blogs (42.9%) than in News (51.2%), indicating the difficulties of extracting correct predicates from texts with informal style. For 774 instances out of the 791 PRED Extraction Errors, the system simply failed to recognize and extract them as a target predicate.

2.4.2. Experiment 2.2: The Rate of Reduction in Surface Differences

2.4.2.1. Data Sets

Next, we examined the reduction rate of differences in surface forms of predicates before and after normalization. We used two data sets.

- *News*: One-year collection of newspaper articles (*Mainichi Newspaper 2003,* 392,865 sentences)
- *Blogs*: Two-week collection of blog articles (*April 1st-14th 2007,* 427,474 sentences)

Our normalizer paraphrased both intermediate and last predicates in these data sets. A total of 478,922 predicates were normalized in News and 479,695 in Blogs. We counted the number of differences in *type* in predicates (i.e., the sequence of $c_1..c_m$-$f_1..f_n$ is counted as one) and the number of differences in *type* in functional expressions (i.e., a sequence of $f_1..$-f_n is counted as one) before and after the normalization. We use the term ORG (ORiGinal) to indicate "predicates/functional expressions *before* normalization" and NORMED to indicate "predicates/functional expressions *after* normalization."

Table 2.4. *Results of experiment 2.2 (News and Blogs)*

News (478,922 Predicates)	# of Surface Differences (TYPE)	Reduction Rate
ORG (Predicates)	111,043	**21.5%**
NORMED (Predicates)	87,168	
ORG (Functional Expressions)	10,572	**57.5%**
NORMED (Functional Expressions)	4,497	

Blogs (479,695Predicates)	# of Surface Differences (TYPE)	Reduction Rate
ORG (Predicates)	191,323	**30.7%**
NORMED (Predicates)	132,620	
ORG (Functional Expressions)	34,971	**66.7%**
NORMED (Functional Expressions)	11,650	

2.4.2.2. Results of Experiment 2.2

Table 2.4 indicates the number of types in predicates as well as functional expressions in News and Blogs before and after normalization and the reduction rates. The reduction rates were calculated by the following equation.

$$\text{Reduction Rate } = 1 - \frac{\text{\# of Differences in NORMED}}{\text{\# of Differences in ORG}} \quad [1]$$

As shown, our normalizer succeeded in reducing the differences in surface forms of predicates by up to 21.5% in News and 30.7% in Blogs. This indicates that as many as 21.5% of the predicates in News and 30.7% in Blogs would have been wrongly recognized as "expressing different meanings" without our normalization. The differences in functional expressions were drastically reduced, by up to 57.5% in News and 66.7% in Blogs.

2.4.3. Experiment 2.3: Impact on a Text Mining Application

Lastly, we examined the impact of our normalization system on a text mining application. As has been discussed, using only the head word of a predicate for text mining systems such as VOC analysis and Factuality Analysis is insufficient because it

might lead to an erroneous conclusion. For example, the predicates *"can't open," "want to open"* and *"opened*," all of which express different facts, would be incorrectly analyzed as *"open."* On the other hand, if the system simply uses the surface forms of Japanese predicates, it would also fail to correctly count up the frequencies of the predicates. This also leads to an incorrect analysis. By normalizing predicates based on their meaning, we can expect to avoid these problems.

2.4.3.1. Task Description

In order to evaluate the effect of our normalization system on an NLP application, we set up the simple text mining task of extracting predicate phrases from two different groups of data sets and finding the bias in predicate distribution towards a particular group (henceforth, *biased predicate extraction task*). The goal of this task is to find phrases that are characteristic to certain data sets.

We conducted the biased predicate extraction task based on head words of predicates (HEAD), original surface forms of predicates (ORG), and normalized predicates (NORMED). We expect that the use of HEAD over-merges different predicates while the use of ORG miscalculates the frequencies of predicates. The use of NORMED will correctly count the frequencies of predicates based on their factual meanings.

2.4.3.2. Data Sets

We used data from *Yahoo! Chiebukuro,*[12] the Japanese version of *Yahoo! Answers*. The data consists of questions and answers posted to *Yahoo! Chiebukuro,* a user-oriented Q&A site. In this experiment, we compared two different pairs of data

[12] http://www.nii.ac.jp/cscenter/idr/yahoo/tdc/chiebukuro_e.html

sets. One is the data in questions of *Internet* category and we compare the data to questions of *Relationships* category (i.e., we did not use the answers). We selected these two groups because the questions posted to these two categories will differ in content and so we can clearly observe the effect of our normalization system on the biased predicate extraction task.[13] We expect expressions such as "I *can't open* a zip file" and "I *want to use* wireless internet connection at home" will occur in *Internet* while expressions such as "I *can't get along with* my new boss" will occur in *Relationships*. The other pair is the data in questions of *Diet* category and we compare the data to questions of *Relationships* category. We selected *Diet* because questions posted to this category often ask what to *do* and/or report what *is happening/happened*, which makes it easier to see the effect of our predicate normalization. We expect expressions such as "I *want to lose* 10 pounds in a month" and "I *gained* 10 pounds" will occur in *Diet*. We used the questions posted in June, 2004.[14]

Pair 1:

- Target 1- Questions in Internet posted in June, 2004 (2,565 questions)
- Comparison 1- Questions in Relationships posted in June, 2004 (3,561 questions)

[13] In the actual text analysis, it is often the case that one compares two similar groups (e.g., smartphones vs. tablets) and finds features that are peculiar to one group and not to the other. Comparing two obviously different groups might bring little valuable information. However, because the goal of the current task is to see the effect of our normalization and not to evaluate the value of the found result itself, we chose groups whose contents were different.

[14] We removed questions that did not have predicate phrases.

Pair 2:

- Target 2- Questions in Diet posted in June, 2004 (621 questions)
- Comparison 2- Questions in Relationships posted in June, 2004 (3,561 questions)

2.4.3.3. Procedure

The procedure of biased predicate extraction task is listed below. In order to calculate bias in a predicate distribution, we used a chi-square (χ^2) test because a χ^2 test gives a score for distributional differences, making it easy to observe the results.

- For each group, count the number of question posts in which each predicate occurs. Consider each post of question as one document and count the document frequency (df) of the predicate. For example, if a predicate *hiraku* "open" occurs in 100 question posts in *Internet* and 5 question posts in *Relationships*, the df of *hiraku* is 100 in *Internet* and 5 in *Relationships*.
- For each predicate, conduct a χ^2 test to compare the distribution of the predicate between the two groups. We used the critical value of 6.635 with the probability $p < 0.01$ of rejecting the null hypothesis (i.e., "the predicate appears at the similar distributional frequency in both groups.").[15] If the χ^2 score is above 6.635, the predicate has a tendency to appear often in *Internet* compared to *Relationships.* Note that the χ^2 test is conducted on every predicate.
- Select prediates whose χ^2 score is above 6.635.

[15] The critical value is 6.635 because we compared two groups (i.e., the degrees of freedom is one).

Table 2.5. *Extracted predicates of the head "use" (Pair 1).*

HEAD	χ^2*score*	NORMED	χ^2*score*	ORG	χ^2*score*
tukau "use"	92.2	***tukat-te-iru* "using"**	64.5	*tukat-tei-masu* "using (polite)"	26.0
				tukat-te-iru-no-desu-ga "using-NOM-COP (polite)-but"	18.1
				tukat-tei-masu-ga "using (polite)-but"	6.9
		***tukai-tai* "want to use"**	12.5	*tukai-tai-no-desu-ga* "want to use-NOM-COP (polite)-but"	6.9
				tukat-te "use (gerundive form)"	15.0

Table 2.6. *Extracted predicates of the head "open" (Pair 1)*

HEAD	χ^2*score*	NORMED	χ^2*score*	ORG	χ^2*score*
hiraku "open"	38.3	***hiraku* "open"**	12.2	none	n.a.
		***hiraka-nai* "can't open"**	11.1		

Table 2.7. *Extracted predicates of the head "lose" (Pair 2)*

HEAD	χ^2*score*	NORMED	χ^2*score*	ORG	χ^2*score*
heru "lose"	62.0	***hera-nai* "can't lose"**	23.0	none	n.a.
		***het-te-iru* "losing (progressive)"**	17.2	none	n.a.
		***heru* "lose"**	11.5	none	n.a.
		***het-ta* "lost"**	9.9	*heri-masi-ta* "lost (polite)"	11.5

Table 2.8. *Extracted predicates of the head "stop" (Pair 2)*

HEAD	χ^2*score*	NORMED	χ^2*score*	ORG	χ^2*score*
n.a	n.a.	***yame-rare-nai* "can't stop"**	16.8	none	n.a.

2.4.3.4. Results of Experiment 2.3

Tables 2.5 and 2.6 show several extracted predicates whose χ^2 scores were above the threshold in Pair 1 (i.e., predicates whose distributions were biased towards

Internet) and Tables 2.7 and 2.8 show those in Pair 2 (i.e., predicates whose distributions were biased towards *Diet*). The head word of the predicates in each table is the same ("use" in Table 2.5, "open" in Table 2.6, "lose" in Table 2.7, and "stop" in Table 2.8). HEAD represents the result of biased predicate extraction task in which only the head of a predicate is used. ORG represents the result in which we used the original forms of predicates and NORMED shows the result of normalized predicates.

Table 2.5 indicates that in HEAD, the different predicates *tukat-te-iru* "using" and *tukai-tai* "want to use" were all merged into *tukau* "use." Similarly, Table 2.6 reveals that *hiraka-nai* "can't open" and *hiraku* "open," which express opposite meaning, were also merged into *hiraku* "open" in HEAD. This oversimplifies the characteristics of *Internet* group, and so fails to obtain the important fact of "wanting to use something" or "not being able to open something." Table 2.7 also shows that *hera-nai* "can't lose," *het-te-iru* "losing (progressive)," *he-ru* "lose" and *het-ta* "lost" were all merged into *heru* "lose." This loses an important distinction between "can't lose" and "lost" expressed in *Diet*.

The use of original surface forms also showed a problem. As shown in Table 2.5, the predicates *tukat-te-i-masu* "using (polite)," *tukat-te-iru-no-desu-ga* "using-NOM-COP (polite)-but," and *tukat-te-i-masu-ga* "using (polite)-but," all of which express the same event of "using," were counted separately. This obscures the characteristic of *Internet* (i.e., the χ^2scores of these predicates became lower). Not only did the use of ORG obscure the result, but also it failed to obtain important predicates. As shown in Table 2.6, no predicate with the head word "open" was extracted in the ORG. This is because the predicates with the head "open" appeared in various surface forms and their frequencies were miscalculated. This results in failing to extract the predicate phrases *hiraku* "open" and *hiraka-nai* "can't open" from the ORG data. A

similar tendency is also observed in Tables 2.7 and 2.8. The use of ORG fails to extract predicates such as *hera-nai* "can't lose" and *yamerare-nai* "can't stop."

When normalized predicates (NORMED) were used, however, these problems of oversimplification and miscalculation seemed to disappear. As shown in Table 2.5, the predicates with the head "use" were extracted in two different forms of *tukat-te-iru* "using" and *tukai-tai* "want to use" from NORMED with the relatively high χ^2scores, indicating the effect of normalization. Furthermore, Table 2.6 shows that only from the NORMED data, was the predicate *hiraka-nai* "can't open" extracted. This is because the predicates in HEAD were all oversimplified, losing the meaning of "can't" while the predicates in ORG were not calculated correctly and so were not extracted. The predicate with the head "stop" in Table 2.8 was only extracted from NORMED. Indeed, a total of 22 different types of predicates in Pair 1 and 36 different types of predicates in Pair 2 were only extracted in NORMED (See Appendix for 22 predicates in Pair 1). In addition, no incomplete predicates which lack appropriate functional expressions were extracted from NORMED while the ORG data had some (e.g., *tukat-te* "use (gerundive form)" in Table 2.5).

2.5. Discussion

As shown, our normalization system can successfully generate simple predicates that contain only the functional expressions essential for retaining the factual meaning of the predicate. The predicates produced by our system had fewer variations in their surface forms while 81.4% (Last) and 77.5% (Intermediate) of them exactly matched the simplified predicates produced by human annotators, i.e., much better performance than the baseline systems.

Accuracies as a Predicate Paraphrase Generator

As the results of Experiment 2.1 show, the proposed system achieves high accuracy as a paraphrase generator because the paraphrasing rules are based on a solid analysis of linguistic theories in semantics and syntax. The quite low accuracy of the baseline methods, especially SimpAdd and Delete on POS further supports our claim that implementing linguistic theories in actual NLP applications can greatly improve system performance. Note that these theories on semantics and syntax are not language dependent; they can be applied to other languages.

Error analysis of Experiment 2.1 reveals that most of the errors were caused either because there were more simplified predicates available or because the generated predicates were ungrammatical. We tried to avoid the second problem by using trigram scores as a measure of normalized predicate grammaticality. However, the results indicate that we need a more sophisticated system to judge the grammaticality (or naturalness) of the output predicates.

Critical errors of our task are those that force normalization to output incorrect predicate meaning. We counted the number of these predicates in Last (Open) data. It was found that only 91 instances out of 1,567 (5.8%) could be considered as critical errors. Most of the critical errors happened when there were more than one negation phrase in the predicate. Recall that we simply delete all the negation phrases if the total number of negations is even and we leave only one if it is odd. Considering the semantic complexity of negations, this deletion rule is too simple and we need to construct more a sophisticated algorithm to deal with negations. Regardless of these errors, we can say that our normalization system achieves high accuracy as a simple paraphrase generator.

Impact as a Predicate Normalizer on an NLP Application

Experiment 2.2 revealed that our normalizing system reduced the differences in surface forms of functional expressions by up to 66.7%. This was achieved because we constructed deletion rules ($f \rightarrow \emptyset$) unlike previous studies (Shirai et al. 1993; Shudo et al., 2004; Matsuyoshi & Sato, 2008). Regardless of the domain of the texts, our paraphrasing system can compress the differences in functional expressions to a limited amount. This is especially important for systems such as VOC analysis and opinion mining from consumer generated media. In these domains, the surface forms of predicates vary greatly compared to a typical written text such as news articles, making it hard for the system to deal with these texts. Our normalization system can simplify them by reducing unnecessary elements.

Experiment 2.3 revealed that our normalization system has an important effect on the biased predicate extraction task, which we set as a simple text mining operation. By normalizing predicates, the system correctly counted the frequencies of predicates while retaining the crucial meaning of functional expressions. We also analyzed the results of Experiment 2.3 and found that 22 different types of predicate phrases in Pair 1 and 36 in Pair 2 were extracted from just the NORMED data. These include *todoka-nai* "haven't received," *kaisetu-si-tai* "want to set up" and *kie-ta* "vanished," all of which could be important expressions for detecting customers' needs and wants. By reducing the surface differences of predicates while retaining the crucial meaning, we succeeded in extracting crucial predicate phrases which would not be found if only the surface forms or the head words were used. This effect was observed regardless of the differences in categories (i.e., *Internet* and *Diet)*.

By comparing the extracted predicates between the *Internet* and *Diet* categories, we also found that predicate phrases on their own gave valuable information for

analyzing the user experience. For example, the expression "*can't lose"* in *Diet* is informative enough to analyze the failure of the user's diet. However, if one needs to extract users' specific needs and wants, not only the predicate information is necessary, but also its argument information. For example, in *Internet*, one might need to extract the expression *"can't install"* as well as its argument *"XX printer"* (i.e., "*can't install - XX printer*"). Our future work will be to combine information conveyed by a normalized predicate and by its arguments and observe the effect of correctly capturing users' needs and wants.

Usability of Paraphrase-based Normalization

Unlike the study of Brun and Hagège (2003), which produces a symbolic representation to normalize textual information, we generate simplified natural language paraphrases as normalized forms. One advantage of paraphrasing as a means of normalization is that it is application independent. Symbolic representations are often constructed in restricted form for use in a particular application as in Brun and Hagège (2003). On the other hand, paraphrased forms can be manipulated by various systems including text mining as in Experiment 2.3 as long as the systems process natural language. For example, we can apply our normalization to Japanese-English translation as a preprocessing step to reduce null alignments between Japanese words and English words. Our normalization focuses on the preservation of factual meaning as well as the grammatical correctness of simplified paraphrases. We can expect a lower risk of information loss than is possible with the previous method which simply deletes function words based on their POSs (Lee et al. (2006)), as was criticized by Hong et al. (2009). An investigation of our normalization as applied to machine translation is, however, future work.

Unlike the study by Inui et al. (2008), we did not include the meaning of *content* words in the normalization system. Therefore, our system is not capable of distinguishing information in which the semantic analysis of content words plays a crucial role (e.g., distinguishing a complaint such as "*can't install"* from a compliment such as *"can't wait for (getting the iPhone app soon)"*). However, this does not mean that our normalization is incapable of conducting deeper semantics analysis. Rather, combining the analysis of content words with functional expressions is important. As mentioned in Inui et al. (2008) as well as in Section 1, bag-of-words-based feature extraction, especially if only content words are used, is insufficient for conducting statistically-based deep semantic analysis. If normalized predicates were used instead of a single content word, we could expect an improvement in those statistically-based methods because each predicate holds important information about *fact.*

2.6. Conclusion of Chapter 2

In conclusion, we presented a novel normalization technique that paraphrases complex functional expressions in Japanese predicates into simplified natural language forms. Paraphrasing rules were constructed based on linguistic theories, and these rules generate paraphrases that, while retaining the crucial information of predicative meaning, are syntactically simple but semantically rich. By normalizing functional expressions in predicates, we succeeded in increasing the recall rates of predicate extraction tasks, which is crucial for text mining systems. The results of our study prove the usefulness of paraphrasing as a means to normalize various linguistic expressions, and provide an encouraging indication of its applicability to actual applications.

CHAPTER 3

PARAPHRASING JAPANESE LIGHT VERB CONSTRUCTIONS: TOWARDS THE NORMALIZATION OF COMPLEX PREDICATES

3.1. Background

Light verb constructions (LVCs) such as "give a try" are complex predicates in which the verbal noun (e.g., try; VN) and the light verb (e.g., give; LV) form a single semantic unit. Japanese LVCs are known for their rich variety. For example, the following three LVCs have the same meaning "get the first prize," regardless of their differences in surface forms.

(14) a. *yuushou*(VN) -*o*(Particle) -*togeru*(LV)

b. *yuushou*(VN) -*o*(Particle) -*hatasu*(LV)

c. *yuushou*(VN) -*o*(Particle) -*suru*(LV)

"get the first prize"

In addition to the above expressions, there is a single verb that has the same meaning, namely *yuushousuru* "(to) get the first prize."

Because of the significant surface variations, it is not easy to identify whether these different verbal expressions have *the same meaning*. This inability causes serious problems in natural language processing (NLP) systems, such as translation errors in machine translation (Wang and Ikeda, 2008), failure to detect correct predicates in

predicate-argument structure analysis, and an inability to group verbal predicates in text mining even though their meanings are similar.

To solve these problems, this paper proposes paraphrasing rules for Japanese LVCs that minimize the surface differences. These rules normalize Japanese LVCs into a very limited number of predicative forms while retaining several of the crucial syntactic/semantic functions of the light verbs.

The chapter is organized as follows. In Section 3.2, we provide a general introduction to the linguistic features of LVCs. We also discuss the peculiarities of Japanese LVCs and the problems caused by them. In Section 3.3, we introduce our paraphrasing rules for Japanese LVCs as well as a disambiguation of LVCs based on an example list. Section 3.4 details experiments and provides an evaluation of our approach. Section 3.5 discusses the results of the experiments and compares the proposal to related works. The last section is the conclusion of this chapter.

3.2. Linguistic Properties of LVCs and Problems Caused by LVCs

Light verb constructions (LVCs) such as “give a try” and “make a change” are a kind of multiword expression in which a verb and a noun form a single semantic unit. One of the peculiar properties of LVCs is that they function as a *single* predicate and the predicative meaning is conveyed mainly by the noun. The verb itself loses its dominant role as a main predicate and simply adds a vague syntactic/semantic function to the preceding noun (Butt, 2003). This is why they are called *light* verbs. For this reason, LVCs often have the same meaning as the verbalized form of the noun (e.g., “give a try” means “try”).

LVCs are found not only in English but also other languages including Japanese (Matsumoto, 1996), Korean (Han & Rambow, 2000) and Urdu (Butt, 2003). Japanese

LVCs are famous for their rich variety. For example, Muraki (1991) lists a total of 155 different verbs as light verbs. These 155 light verbs appear in the structure of verbal noun(VN)-case particle(P)-light verb(LV) (i.e., the case particle splits the verbal noun and the light verb). This VN-P-LV structure is problematic for NLP systems such as predicate-argument structure analyzers because the structure takes the same syntactic surface structure as the typical predicate-argument relation in Japanese, namely *noun-particle-verb.* If the system simply relies on the syntactic structure, then the LV would be wrongly detected as the main predicate.

(15) a. Typical Predicate-Argument Structure

fune(Noun) -o(P) ***-ukaberu(Verb)***

boat (Argument) ACC float (Predicate)

"float a boat"

b. The LVC (VN-P-LV) structure

nigawarai(Noun) -o(P) -ukaberu(Verb)

bitter smile ACC float

"smile bitterly"

In (15a), the main predicate is *ukaberu* "to float" and the argument (*theme*) of the predicate is *fune* "a boat." On the other hand, in (15b), the main predicate is the whole LVC, namely *nigawarai-o-ukaberu* "smile bitterly." However, due to the same syntactic structure, the system would wrongly detect the main predicate of (15b) as *ukaberu "to float."* Because the LVCs with the VN-P-LV structure appear at the rate of around 4.0% in newspaper articles, the existence of these LVCs significantly lowers system performance. Furthermore, as discussed in (Kaji & Kurohashi, 2004), not only for predicate-argument structure analyses but also for other NLP systems such as

information retrieval, QA systems, multi-document summarizations, and text preprocessors for speech-to-text synthesis, the simplification of the VN-P-LV structure is important.

As shown, the VN-P-LV structure of Japanese LVCs is problematic and needs to be correctly captured. One solution is to simplify the VN-P-LV structure. However, there are two problems that we need to solve when paraphrasing the VN-P-LV structure.

First, the LV of an LVC is not completely empty in terms of their semantic or syntactic functions. Like LVCs in other languages, Japanese LVCs often convey the same intrinsic meaning as the verbalized form of the VN. Therefore, it is possible to simply paraphrase an LVC by verbalizing the VN as in (14). However, light verbs in Japanese also add some syntactic/semantic information to the main predicate.

(16) a. *henkou(VN) -o(P) -okonau(LV)*
change ACC carry out
"change (literally, carry out the change)"

b. *henkou(VN) -o(P) -shiiru(LV)*
change ACC force
"force someone to change"

(16a) means "(to) change" and the agent of the VN is the subject. (16b), on the other hand, means "(to) force someone to change" and it is in the causative construction; the light verb *shiiru* expresses the causativeness of the action indicated by the predicate VN.

The second problem is ambiguity. Japanese LVs exhibit ambiguity; they function as a light verb or as a main verb (i.e., being 'heavy') depending on the VN.

Furthermore, several LVs have more than one syntactic/semantic function, so simply paraphrasing the LVs of all VN-P-LVs in the same manner is invalid.

(17) a. *adobaisu(VN)* *-o(P)* *-ataeru(LV)*
advice ACC give
"give advice"

b. *kinchou(VN)* *-o(P)* *-ataeru(LV)*
tension ACC give
"make someone nervous (literally, give a tension)"

c. *copii(VN)* *-o(P)* *-ataeru(LV)*
copy ACC give
"give someone a copy"

All the examples in (17) have the VN-P-LV structure with the same light verb, *ataeru* "give." Only the VNs are different. However, while (17a) can be simply paraphrased by verbalizing the VN (i.e., *adobaisusuru)*, (17b) should be paraphrased as *kinchous-aseru,* in which the causative inflectional morpheme *-aseru* is attached to the verbalized VN. Verbalizing the VN of (17c), in which *ataeru* functions as a heavy verb, is invalid.

Besides these ambiguities, several idiomatic expressions also take the form of LVC.

(18) *nanori(VN)* *-o(P)* *-ageru(LV)*
introducing one's name ACC raise
"join a competition (literally, raise one's name)"

Because the idiomatic meaning is not sustained if the LVC is transformed into a different form, verbalizing the VN results in a paraphrase error. As shown, those

so-called LVCs are indeed ambiguous and they can be paraphrased *only* when the LVs truly function as a light verb.

In sum, several of the properties of LVCs cause serious problems in NLP systems. We focus here on the complex surface structures of VN-P-LV, the impact of the syntactic/semantic functions of LVs on the meaning of the main predicate, and the ambiguity in terms of LV function. This study solves these problems by constructing a solid rule-based paraphrasing system. More specifically, the system solves the above problems based on the following strategies;

- Paraphrase complex VN-P-LV structures into very simple predicative forms.
- Conduct a full linguistic analysis of the syntactic/semantic functions of LVs and construct paraphrasing patterns[16] that sustain the crucial meaning of the LVs while removing elements that are unnecessary to the meaning of the predicate
- Construct a list of ambiguous LVCs and use the list to disambiguate an entered LVC.

In the following section, we provide a detailed architecture of our rule-based normalizing system.

[16] Throughout this paper, we use the term *paraphrasing patterns* to refer to both paraphrased *output* as well as *procedure* to generate the output.

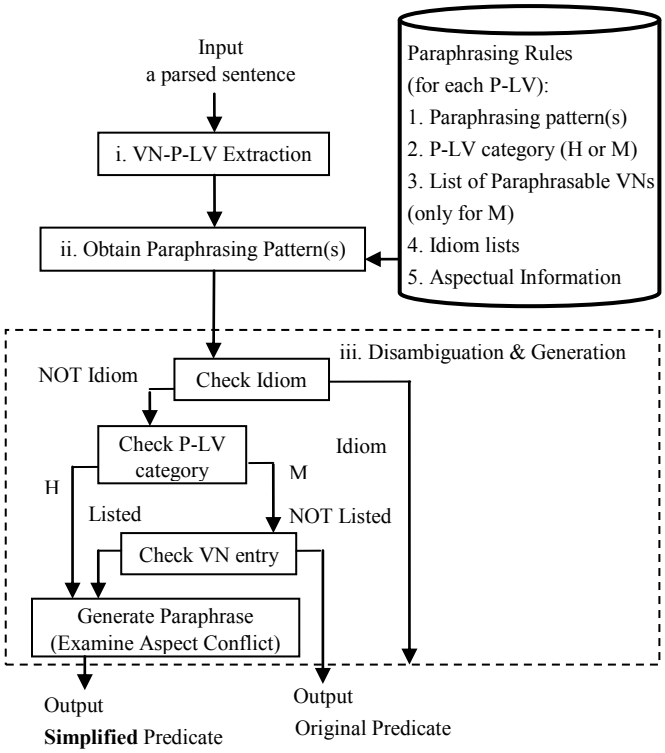

Figure 3.1. *Our normalization system.*

3.3. Forming LVC Paraphrasing Patterns and Constructing Paraphrasable LVC sets

This section introduces our rule-based LVC normalizing system. The overall flow of our system is described in Figure 1. The system works as follows.

i. From a parsed sentence, extract a VN-P-LV (e.g., *yuushou(VN)-o(P)-hatasu(LV)*).

ii. Use the P-LV to select the corresponding paraphrasing pattern(s) (e.g. *o(P)-hatasu(LV)* -> verbalize VN). Also, obtain the P-LV category (which indicates if the LV is ambiguous), paraphrasable VN lists, idiomatic VN lists, and aspectual information (all of which will be discussed later).

iii. Based on LVC disambiguation, those that meet our paraphrasing conditions are paraphrased into a single predicate form (e. g., *yuushou(VN)-o(P)-hatasu(LV) -> yuushousuru*). If the LV has aspectual information, the paraphrasing pattern that retains this information is selected (also discussed later).

3.3.1. *Fundamental Principles for Building Paraphrase Patterns: Analyzing Syntactic and Semantic Functions of LVs*

The crucial point of our paraphrasing system is how we construct paraphrasing patterns. One of our main goals is to decrease the differences in the surface forms of Japanese LVCs so that we can group verbal predicates that have similar meaning. However, LVCs that are clearly different must not be placed in the same group. The following is an example.

(19) a. *yuushou(VN)* *-o(P)* *-togeru(LV)*
the first prize ACC accomplish
"get the first prize"

b. *yuushou(VN)* *-o(P)* *-mezasu(LV)*
the first prize ACC aim at
"try to get the first prize"

As the English translations show, these two LVCs have different meaning. *Yuushou-o-togeru* means "get the first prize," while *yuushou-o-mezasu* means "**try to** get the first prize," indicating the speaker's eagerness to get the first prize.[17]

[17] This difference is seen more clearly if we change the tense interpretation of the LVCs into past. While *yuushou-o-togeta* (*ta* is the past tense marker in Japanese) indicates the *fact* of getting the first prize (that is, someone is now the first prize holder), *yuushou-o-mezashita* does not mean he or she *actually* got the first prize.

As shown, our paraphrasing patterns should not only be simple enough to integrate different LVs into the same surface form but also sophisticated enough to retain the syntactic/semantic differences of the LVs.

This raises the problem of *what differences in meaning can be conveyed by light verbs?* This is a tough question to answer because some light verbs contribute very subtle semantic meaning to the main predicate while others drastically change the meaning. Recall that our goal is not to find these subtle differences in meaning by different LVs. Rather, we aim to group different LVCs by paraphrasing them into a single predicate if they share similar meaning. For this purpose, we simply follow our intuition that "different predicates can be said to have the same meaning if they express the same *event*." In making paraphrasing patterns, we distinguish the following grammatical information among the various clues that denote events: *voice*, *modality* and, only when necessary, *aspect*.[18]

We assume this is reasonable because, as is discussed in Narrog (2005) and Portner (1995) modal expressions indicate an event that is different from reality. That is, modal expressions add information such as possibility (e.g., *may*), obligation (e.g., *must*) and eagerness (e.g., *want*) to the predicate, indicating that the event (with a modal) is *not factual* as opposed to that without a modal (factual). Therefore, by retaining the modal information conveyed by an LV, it becomes clear whether the predicate is expressing reality or not. This explains the difference in (19).

---

[18] Tense information also changes the meaning of *event*. We did not include tense simply because there is no such construction in which tense is conveyed by an LV. Tense is always conveyed by either the tense marker *ta* (past) or the verb ending of *–(r)u* (non-past) in Japanese.

Retaining the voice information expressed by an LV also makes sense because voice information plays a crucial role in detecting the semantic roles of their arguments. Recall that in (16) in Section 2, the causative information is expressed by the LV *shiiru*. The syntactic function of *shiiru* sets the argument structure of the LVC.

Following our principles, we have to sustain the aspectual information expressed by an LV. However, whether or not we should retain the aspectual information of the LV is not straightforward. As discussed in (Butt, 2003), light verbs often have, even across languages, the function of adding aspectual information to the main predicate. Japanese is not an exception.

(20) *shuufuku(VN)* *-o(P)* *-hajimeru(LV)*
repair ACC begin
"begin to repair"

In this case, the LV *hajimeru* "begin" expresses the beginning of the repairing action. However, (20) can be simply paraphrased as *shuufukusuru* "repair," the verbalized form of the VN *shuufuku*. The LV *hajimeru* need not be transformed into an inflectional morpheme unlike the *voice* or *modality* expressions discussed so far. This is possible because, in Japanese, the present tense of a verb has the function of indicating future/inceptive action (Tsujimura, 2007), so the aspectual information conveyed by the LV is redundant.

However, we still have to explicitly transform the aspectual information of the LV into an inflectional morpheme. This is necessary when the aspectual information of the LV conflicts with that of the main tense.
The following is an example.

(21) *shuufuku(VN)* *-o(P)* *-hajimeta(LV)*

repair ACC begin-PAST

"began to repair"

If one drops the aspectual information of the LV and simply verbalizes the VN in (21) as *shuufukushita*, this would force the paraphrased form to have a different event from the original LVC. While *shuufuku-o-hajimeta* "began to repair (something)" indicates the event that something is now in the process of being repaired, *shuufukushita* "repaired (something)" indicates that something has finished being repaired (i.e., it is not broken anymore). This happens because the Japanese past tense marker has the aspectual meaning of the completion of an action, meaning more like "it has been repaired." In this case, we must keep the aspectual information of the LV because dropping it would force the paraphrased form to express a different event from the original LVC.

The same pattern applies to LVs that convey aspectual information of "completion." These LVs can be dropped only when the tense of the main predicate is in the past form, which also expresses completion of the main predicate. Therefore, we can drop the aspectual information of an LV if and only if its aspectual interpretation agrees with that of the main predicate.

In sum, our paraphrasing patterns follow the following criteria. First, all the LVCs should be paraphrased into the simplest form. Only when LVs hold voice (syntactic) or modality (semantic) information, should we retain their function as in the form of verbal inflection. Last, we should keep the aspectual information conveyed by an LV only when there is a conflict between the aspectual information of the LV and that of the main predicate (inceptive vs. past tense or completion vs. present tense).

3.3.2. *Detecting Ambiguous LVCs via a List of Examples*

Another problem that we need to solve is LVC disambiguation. As discussed in Section 2, there are two types of ambiguity. One is the ambiguity of light or heavy usage (Type 1 ambiguity). The other is the ambiguity of the syntactic/semantic function of the light verb (Type 2 ambiguity). The below examples are taken from (17). A VN-P-LV can be paraphrased if and only if the LV indeed functions as a light verb.

(22) a. *adobaisu(VN)* *-o(P)* *-ataeru(LV)*
advice ACC give
-> *adobaisusuru* "to advise"

b. *kinchou(VN)* *-o(P)* *-ataeru(LV)*
tension ACC give
->*kinchous-aseru* "to make someone nervous"

c. *copii(VN)* *-o(P)* *-ataeru(LV)*
copy ACC give
->unable to paraphrase

As these examples show, we need to decide which VN-P-LV can be paraphrased (i.e., the LV is functioning as a light verb) and if so, into which pattern it should be paraphrased.

Recall that whether an LV is paraphrasable or not depends on its VN. Furthermore, not all LVs show ambiguity. LVs such as *suru* almost always function as a light verb no matter which VN they are combined with. The number of ambiguous LVs is limited and those ambiguous LVs function as a light verb only when they are combined with *certain types* of VNs.

For this reason, it is possible to disambiguate LVs by manually constructing a list of paraphrasable VNs for each ambiguous LV from large corpora. That is, for

ambiguous LVs, we only paraphrase a VN-P-LV if the VN is listed. We can also avoid wrong paraphrasing of idiomatic LVCs by listing these examples as *not-paraphrasable*.

In sum, our normalization system is constructed based on *paraphrasing patterns* and *conditions*. *Paraphrasing patterns* decide a simple predicate form for each LVC. *Conditions* check the validity of paraphrasing an entered LVC by checking whether the LVC is a real LVC (LVC Disambiguation) and whether the aspectual information can be sustained without the LV (Examine Aspect Conflict).

3.3.3. *Construction of Paraphrasing Patterns and List of Examples for Ambiguous LVs*

We constructed our paraphrasing patterns as well as a list of paraphrasable VNs for ambiguous LVs based on instances in the data extracted from large actual corpora. In order to make our paraphrasing patterns and example list as exhaustive as possible, not only did we use large amounts of data extracted from different domains (newspapers and blog articles), but we also examined as many instances as possible based on their frequencies. Out of 39,130 different types of VN-P-LV appeared in the data, we examined around 2,700 instances with 160 different P-LVs in order of frequency. By doing this, we can construct paraphrasing patterns and a list of examples that can cover frequently occurring expressions.

3.3.3.1. Resource Data

We used 12 years of newspaper articles *(Mainichi* newspaper 1991-2002; 5,583,644 sentences) as well as 1 year of blog articles (about 8,240,000 sentences) as the source for constructing the paraphrasing patterns and an example list for ambiguous LVs. By using newspaper and blog articles, we can expect to cover a broad

range of LVC expressions ranging from those in formal writing to those in casual writing.

As for light verbs, we used the light verbs listed in Fujita et al., (2004) and those in Muraki (1991).[19] Because we focus on LVCs with the VN-P-LV pattern, we split each entry into a unique pair of an LV and a particle (P-LV). The total number of P-LV entries was 160 in terms of type. As for VNs, we simply used a dictionary and POS tags of parsed sentences.

We extracted VN-P-LV instances from the newspaper and blog articles.[20] 223,822 instances from 5,583,644 sentences (4.0%) were extracted for the newspaper articles while 153,364 instances from 8,240,000 sentences (1.9%) were extracted from the blog articles.

3.3.3.2. Forming LVC Paraphrasing Patterns

We applied the following procedure to the extracted VN-P-LV instances in order to construct paraphrasing patterns as well as to divide them into unambiguous P-LVs and ambiguous ones.

i. Select the top 5 instances for each P-LV pair from the newspaper articles, and choose the top 5 instances that were *not* found in the newspaper articles from the blogs, if available (A maximum of 10 instances were examined).

---

[19] We excluded three entries listed in Fujita et al. (2004) and two in Muraki (1991) from our light verb dictionary. These verbs do not show the syntactic behaviors typical of light verbs discussed in Matsumoto (1996). These verbs are *kentousuru* "consider", *sizumu* "sink", *susumu* "proceed", *sasou* "invite", and *motomeru* "demand".

[20] We used ChaSen as a POS tagger (http://chasen-legacy.sourceforge.jp/) and CaboCha as a dependency parser (http://cabocha.sourceforge.net/).

ii. By following the paraphrasing rules in Section 3.1, make paraphrasing patterns for each P-LV by focusing on the sustainability of the meaning of the *event* denoted by the predicate.

iii. Categorize each P-LV as H (productivity as an LV is High) if all instances can be paraphrased in the same manner (P-LV(H)). If not, categorize the P-LV as M (productivity as an LV is Middle; P-LV(M)).

97 P-LVs were categorized as H (P-LV(H)) while 54 P-LVs were categorized as M (P-LV(M)). These P-LV(M)s are ambiguous P-LVs either because the LVs kept their "heavy" usage as a main verb (Type 1) or because they were paraphrased into different forms depending on the preceding VN (Type 2).

Table 3.1 indicates our paraphrasing patterns and the number of P-LVs that belongs to each pattern. Note that 7 out of 151 P-LVs were assigned two paraphrasing patterns because they are ambiguous P-LVs and are paraphrased into different forms depending on the preceding VN (Type 2 Ambiguity). This is why the accumulated number of P-LVs in Table 1 is 158 (151 + 7). Except for nine P-LV pairs, we were able to map 151 different P-LV pairs to 10 paraphrasing patterns.

Table 3.1. *10 Paraphrasing patterns for Japanese LVCs.*

Output Form	Type of paraphrasing pattern	# of P-LVs: Examples	Example of paraphrased predicate (VN = *adobaisu* "advice")
V-Inflection	1.Verbalize VN	111: *o-okonau* "do(formal)", *o-suru* "do"...	*adobaisuSURU* "to advise"
	2.Verbalize VN + passive	13: *o-ukeru* "receive", *o-atsumeru* "gather"…	*adobaisus-ARERU* "to be advised"
	3.Verbalize VN + causative	13: *o-shiiru* "force", *ni-toru* "steal"...	*adobaisus-ASERU* "to force someone to advise"
	4.Vervalize VN +causative passive	1: *o-kissuru* "suffer"	*adobaisus-ASERARERU* "to be forced to advise"
	5.Verbalize VN + modality (intention)	5: *o-hakaru* "plan", *o-mezasu* "aim"...	*adobaisushi-YOUTOSURU* "to intend to advise"
	6.Verbalize VN + causative + modality (intention)	1: *o-unagasu* "urge"	*adobaisus-ASEYOUTOSURU* "to intend to force someone to advise"
	7.Verbalize VN +modality (capability)	4: *ga-iku,* "go" *o-dekiru* "can"...	*adobaisu-DEKIRU//RERU* "can advise"
VN-P-LV	8.VN + NOM(P) + Existential LV	3: *ga-mirareru,* "be seen" *o-miru,* "see"...	*adobaisu-GA-ARU* "there is advise (agent-less)"
	9.VN + DAT(P) + Resultive LV	5: *o-maneku,* "cause" *ni-owaru* "end up with"...	*adobaisu-NI-NARU* "end up with advice"
	10.VN + ACC(P) + Completive LV	2: *o-uchikiru,* "cut off" *o-yameru* "stop"	*adobaisu-O-YAMERU* "stop advising"

The nine P-LV pairs were those that could not be simplified into a single predicate (i.e., V-Infl), nor were there any VN-P-LV patterns that could merge these nine P-LVs. The patterns constructed are as follows.

- 7 of the 10 patterns convert an LVC into a single predicate form, which eliminates the erroneous syntactic assessment of a light verb as a main predicate.

- Three paraphrasing patterns stayed in the form of VN-P-LV (Patterns 8-10), although all the P-LVs in each pattern are merged into one P-LV.

It was not possible to paraphrase P-LVs with Patterns 8-10 into a single predicative form. P-LVs in Patterns 8 and 9 were in LVCs with agentless constructions, such as *henkou(VN)-ga(P)-mirareru(LV),* literally translated as "a change is seen." As the translation indicates, the subject/agent of the verbal noun (i.e., "change") is neither explicitly mentioned nor can it be easily defined by the context. Because these constructions do not allow one to know who the agent of the main predicate is, there is no way to verbalize the VN since this requires mention of an explicit agent.

P-LVs in Pattern 10 have the aspectual meaning indicating termination of an action (terminative), similar to the English "stop." Unlike the inceptive or completive aspect, Japanese does not use any inflectional morphemes to indicate the terminative meaning of an action. Therefore, we decided to group these LVCs together by paraphrasing them into the same P-LV, namely *o-yameru.*

In addition to these 10 patterns, we made 2 additional patterns indicating the addition of aspectual auxiliary verbs, namely *hajimeru* "begin to" and *owaru* "finish to." These patterns are used only when there is a conflict between the aspectual information of the LV and that of the main predicate. This was done in order to avoid the paraphrase errors caused by conflict in tense/aspect interpretation as discussed above.

3.3.3.3. Constructing an Example List for Ambiguous P-LVs

Another resource that we need to construct is a list of paraphrasable VNs for 54 ambiguous P-LV(M)s. The procedure for constructing this list is as follows.

Table 3.2. *Number of P-LV categorized as H or M and number of VNs* listed.

P-LV Category	P-LV entries	VN listed	Idiomatic LVCs
H	97	n.a	6
M	47 (Type 1) 7 (Type 1 and Type 2)	923	48

i. For each P-LV(M), select VN-P-LV(M) instances that appeared at least 5 times from the newspaper and blog articles. If no example is available, select VN-P-LV(M) instances that appeared less than 5 times.

ii. Examine each instance and determine if it is paraphrasable. If so, decide into which pattern it should be paraphrased and add the VN to the list of P-LV(M).

The total number of instances that we examined was 1,954 (1,568 from the newspaper and 374 from the blog articles), and from those instances, we selected a total of 923 VNs for entry in our example list.

Besides them, we also added "Idiomatic LVCs," which have syntactic LVC patterns but function as idioms. Because they cannot be paraphrased due to their idiomatic nature, based on Sato (2007), we list these idiomatic LVCs as "exceptions for paraphrasing." Table 3.2 indicates the number of entities in our example list as well as that of idiomatic LVCs.

3.4. Experiments and Evaluations

We conducted two experiments. The first experiment measures the accuracy of our paraphrasing system as well as the coverage of our example list. The second experiment measures the effectiveness of our paraphrasing system as a predicate normalizer.

Table 3.3. *Number of paraphrased instances and types of P-LVs.*

Domain	# of VN-P-LV instances	# of P-LV types found
News	3.9% (15,443/392,865)	130
Blogs	1.9% (990/52,152)	80

Table 3.4. *Accuracy of paraphrasing rules (Newspaper)*

	Newspapers	# of instances in H and M	
		H	M
E-Preserved	**89.7%** (269/300)	223/269	46/269
E-Changed	4.0% (12/300)	**11/12**	1/12
Errors	6.3% (19/300)	13/19	6/19

Table 3.5. *Accuracy of paraphrasing rules (Blogs)*

	Blogs	# of instances in H and M	
		H	M
E-Preserved	**93.0 %** (279/300)	225/279	54/279
E-Changed	1.0 % (3/300)	**3/3**	0/3
Errors	6.0 % (18/300)	15/18	3/18

3.4.1. *Experiment 3.1: Accuracy and Coverage*

3.4.1.1. Measuring Accuracy

In order to test our LVC paraphrasing system, we used one-year of newspaper articles (*Mainichi* newspaper 2003; 392,865 sentences) and blog articles (52,152 sentences), neither which were used as resource data. We extracted VN-P-LV instances in the same way as we did in Section 3. Only those VN-P-LV instances that passed our paraphrasing rules were automatically paraphrased. 15,448 (3.9%) instances from the newspaper articles and 990 (1.9%) instances from the blog articles were paraphrased. The number of types of P-LVs found in the newspaper and the blog were 130 and 80, respectively (Table 3.3).

We measured the accuracy of the paraphrased instances as well as the coverage of our example list for P-LV(M)s. The accuracy was measured as follows.

i. Randomly select 300 paraphrased instances from each domain.

ii. If a paraphrased output was ungrammatical (e.g., verb conjugation error), count them as Errors.

iii. If the original VN-P-LV and its paraphrased predicate denote the same *event,* count them as E(vent)-Preserved. If not, count them as E(vent)-Changed.

The results shown in Table 3.4 indicate that 89.7% (93.0%) of the LVCs in Newspapers (Blogs) were paraphrased into a simple predicative form without changing the meaning of the event expressed by the original LVC. This indicates that our manually constructed paraphrasing patterns are valid enough to achieve high accuracy.

3.4.1.2. Measuring Coverage of the Example List

In order to measure the coverage of our example list, we took the following procedures.

i. From the same test data, extract all the VN-P-LV(M) instances in which the P-LVs were in category M.

ii. Divide them into two groups: those that were paraphrased (i.e., listed in the example list) and those that were *not* paraphrased (i.e., not listed in the example list). Measure the ratio of the two groups (ParaphRate vs. NOTParaphRate).

iii. Randomly select 300 instances from each group and determine whether they really are an LVC (ShouldBeLight) or not (ShouldBeHeavy)

The total number of VN-P-LV(M) instances extracted was 4,591 and the ratio of ParaphRate vs. NOTParaphRate was 67.9% vs. 32.1%. The distribution of correct LVC detection is listed in Table 3.6.

Table 3.6. *Measuring the coverage of example list*

	ShouldBeLight	ShouldBeHeavy
ParaphRate:**67.9%** (3,117/4,591)	**A: 98.3%** (295/300)	B: 1.7% (5/300)
NOTParaphRate:**32.1%** (1,474/4,591)	**C: 32.7%** (98/300)	D: 67.3% (202/300)

32.7% of those that were *not* paraphrased were actual LVCs and so should have been paraphrased. Although our example list did not cover these instances, the coverage of our list is still high. Based on [2] and Table 3.6, we calculated the coverage of our list as follows.

$$\text{Coverage} = \frac{\text{ParaphRate} \times \text{A}}{\text{ParaphRate} \times \text{A} + \text{NOTParaphRate} \times \text{C}} \quad [2]$$

The coverage of our manually constructed example list is approximated to be quite high at 0.86.

3.4.2. *Experiment 3.2: Impact on NLP Applications as a Predicate Normalizer*

The second experiment was conducted in order to investigate the impact of our paraphrasing system on NLP applications. As discussed earlier, one of the serious problems caused by LVCs lies the variety in surface forms. This makes it hard for NLP systems to recognize whether different LVCs are expressing the same meaning. Because this causes problems, especially to text mining systems, which often extract and group predicates for their analysis (e.g., Nasukawa & Nagao, 2001), we conducted a predicate extraction task which extracts and counts the number of extracted predicates. If our paraphrasing system works effectively as a normalizer of complex predicates, we can expect an increase in the number of extracted predicates.

3.4.2.1. Data

We used the 5-year set of newspaper articles (*Mainichi* newspaper 2003-2007; 2,117,105 sentences) that were not used for constructing the paraphrasing patterns discussed in Sections 3.3.2 and 3.3.3. We used two data sets: one which went through our paraphrasing system (Paraphrased data) and the other which did not (Original data).

As target predicates, we randomly select verbal nouns and automatically verbalized them into Patterns 1-7 in Table 3.1. We only used Patterns 1-7 because these are the patterns that convert the complex VN-P-LV structure of LVC into a single predicative form (i.e., V-Infl).

3.4.2.2. Measuring the Increase in Recall

For each target predicate, we extracted matched items from the Paraphrased data and from the Original data. We then compared the number of extracted items for each target predicate. We removed the predicates that appeared neither in the Paraphrased data nor in the Original data, and selected a total of 9,643 predicates as target predicates for our analysis.

In order to see the increase in the recall rate, we calculated the increase rate for each target predicate as follows:

$$\text{Increase Rate} = \frac{\text{\# of Extracted items in the Paraphrased data}}{\text{\# of Extracted items in the Original data}} \quad [3]$$

If Predicate A appears 5 times in the Original data and 10 times in the Paraphrased data, then the increase rate of Predicate A is 2.0.

Figure 3.2 shows the result. The y axis indicates the increase rate, ranging from 1.0 (NO CHANGE) to 5.0 and more (5.0-more). The x axis indicates the number of

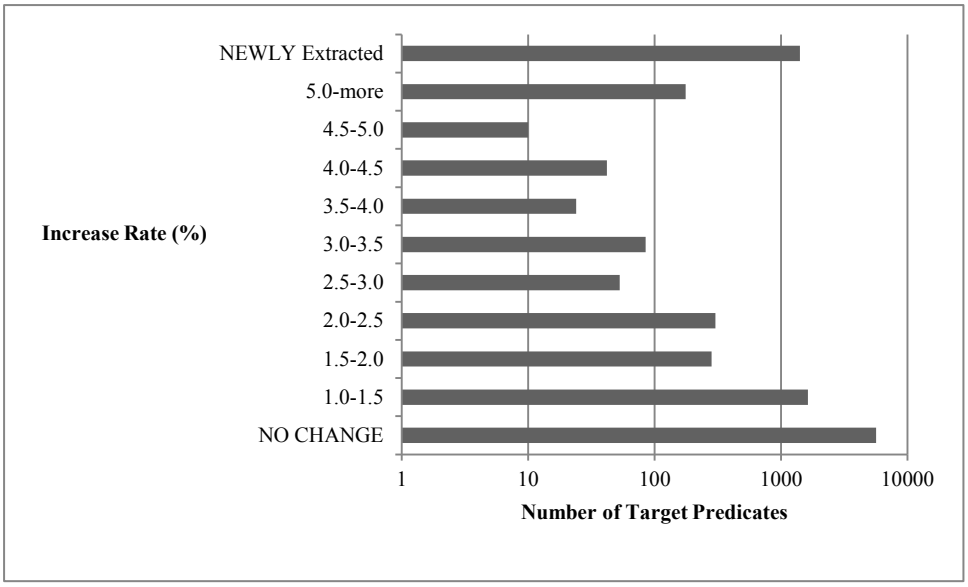

Figure 3.2. *Result of the predicate extraction task.*

Table 3.7. *The overall increase rate*

Total Number of Extracted Predicates		
Original data	Paraphrased data	Overall Increase Rate
447,224	498,249	1.11

target predicates categorized in terms of their increase rate. NEWLY Extracted indicates that a target predicate is extracted only from the Paraphrased data.

The result indicates that 41.6% (4,010 / 9,643) of the target predicates show an increase in recall after our paraphrasing. The average increase rate is 2.48. Although 58.4% (5,633 / 9,643) of the predicates did not show any increase in recall, these predicates tend to be low-frequency predicates (35.3% of them only appeared once). Notably, 14.6% (1,409 / 9,643) of the predicates were extracted only from the Paraphrased data, meaning that without our paraphrasing system, these predicates would not have been detected (NEWLY Extracted). The overall increase rate of this task is 1.11 (Table 3.7).

3.5. Discussion and Related Works

As the results show, various Japanese LVCs were paraphrased into very simple forms with high accuracy. A total of 132 different P-LV patterns appeared in the test data were reduced to 10 paraphrasing patterns, indicating that we succeeded in achieving our main goal of suppressing the surface differences caused by various LVCs. This is also shown in the results of Experiment 3.2 in which 41.6% of the predicates used in the experiment indicated an increase in recall (the average increase rate was 2.48).

This study covers a wider variety of LVs than previous studies (only *suru* "do" in Wang and Ikeda (2008) and only *make, take,* and *give* in Stevenson et al. (2004). Unlike the study of Fujita et al. (2004), which also covers various LVs (i.e., 40 types), our method does not require any additional language resources. Rather, our method can paraphrase different LVCs including ambiguous ones with high accuracy based on simple paraphrasing rules and an example list. Although we focused here only on paraphrasing complex predicates, Fujita et al. (2004) aimed at paraphrasing whole clauses including case marker transformation, our study is still valuable in that we succeeded in suppressing the surface differences of complex predicates, which is problematic for most NLP systems.

Most of the E-Changed errors (i.e., wrong paraphrasing) in Experiment 3.1 were caused by P-LV(H)s. These P-LV(H)s were actually ambiguous P-LVs whose type, light or heavy, depends on context. Recall that we categorized each P-LV as H or M based on the top 10 frequent instances. These errors could be avoided if we set a stricter criterion for categorizing H either by examining more instances or selecting instances at random.

Regardless of these errors, we achieved the high accuracy of around 90% in both domains. This is because we avoided paraphrasing errors by constructing a list of paraphrasable VNs for P-LV(M)s. This approach to detecting paraphrasable LVCs, which is based on an example list, is different from that in Fazly and Stevenson (2007), which used a statistical method to find LVCs. While statistical methods can be applied to any LVC, even if those that do not appear in the training data, ours only detects the LVC patterns that are listed. However, our system covers a broad range of ambiguous LVCs (the coverage is .86). This is because ambiguous P-LVs are those that function as an LVC only when they are combined with *certain* types of VNs and we could collect them with relatively high coverage by examining a 12-year collection of newspaper and a 1-year set of blog articles.

The results of Experiment 3.2 indicate that our paraphrasing system works effectively as a normalizer of complex predicates. For 41.6% of the target predicates used in the experiment, the average increase in the rate of recall was of 2.48. Furthermore, our paraphrasing system makes it possible to extract predicates even though they did not even appear in the Original data.

The system increases the overall increase rate for the predicate extraction task to 1.11. Although the overall increase rate seems relatively small, this is still valuable for the following two reasons. First, the experiment revealed that 41.6% of the verbal nouns appeared in the form of LVCs, and our paraphrasing system succeeded in extracting and merging them with their verbalized counterparts. Second, our study revealed the effectiveness of automatic paraphrase systems in NLP applications such as text mining. Unlike the study of (Brun & Hagège, 2003), which uses symbolic representations as the output of normalization, we directly use natural language paraphrases as the output of normalization. This reduces the need for the addition of

paraphrase generation, from artificial symbols to natural language, making it easy to apply our normalization system to various NLP applications including machine translation, predicate-argument structure analyzers, and summarization.

3.6. Conclusion of Chapter 3

In this chapter, we presented our novel rule-based paraphrasing system for Japanese LVCs that reduces the differences in the surface structures of verbal predicates. By analyzing the linguistic properties of Japanese LVCs, as well as examining a number of VN-P-LV instances from large corpora, we created patterns that allowed various LVCs with 151 different P-LVs to be paraphrased into 10 simple predicative forms. In addition to these patterns, we also made a list of 923 examples of ambiguous P-LVs. By using the list to disambiguate LVCs, we could paraphrase different LVCs into simple forms while still keeping the accuracy high. Our paraphrasing system works as a normalizer of complex predicates and increases the recall rate of systems such as text mining by reducing the differences in surface forms while retaining the original meaning.

Future work includes investigating the effect of our paraphrasing system on other NLP systems such as predicate-argument structure analyzers. We will also develop our paraphrasing patterns for not only LVCs but also other types of complex predicates to achieve a more sophisticated predicate normalization system.

CHAPTER 4

RECOGNIZING SEMANTICALLY SIMILAR PREDICATE PHRASES BASED ON LINGUISTICALLY-MOTIVATED FEATURES

4.1. Background

Identifying synonym and antonym relations between words and phrases is one of the fundamental tasks in Natural Language Processing (NLP). Understanding these semantic relations is crucial for realizing many NLP applications including QA systems, information retrieval, text mining etc. Among various word and phrasal relations, identifying the semantic relations between *predicates* is especially important because predicates convey the propositional meaning of a sentence. For example, identifying synonymous predicates such as *"can't repair X"* and *"unable to fix X"* is crucial for text mining systems.

However, it is hard to obtain a rich language resource that can completely cover the synonym-antonym relations of predicates in sufficient detail. This is because the meaning of a predicate varies depending on its context. For example, "ignore" and "break" can express the same meaning if they are combined with the argument "rule" (*break* the rule vs. *ignore* the rule).

In this chapter, we propose the supervised classification of synonymous predicates based on linguistically-motivated features. As features for recognizing semantically similar predicates, we use two different kinds of features; one for

recognizing synonyms and the other for recognizing antonyms. To support training and evaluation, we also construct a large human annotated set of predicate pairs for synonym-antonym relations. Accompanied by a noun and a predicate, the data consists of predicate-argument pairs such as "*consume*-memory (ACC)" vs. "*eat*-memory (ACC)"; the relations are categorized as synonyms, antonyms, or unrelated.

This chapter is organized as follows. In Section 4.2, we provide related work on the recognition of semantically equivalent predicates. Section 4.3 details our proposed method of automatic classification of synonymous predicates and Section 4.4 details the corpus constructed for this thesis. Section 4.5 details the experiment conducted and Section 4.6 discusses the results. Section 4.7 is the conclusion of this chapter.

4.2. Related Works

4.2.1. Paraphrasing Based on Dictionaries

Fujita et al. (2004) use Lexical Conceptual Structure (LCS: Jackendoff 1992; Takeuchi et al., 2006) in order to paraphrase light verb expressions such as *give an influence* into a simplified verbal predicate such as *influence* (e.g., "The rate of change in stock *gave an influence to/influenced* the exchange rates"). Similarly, Kaji and Kurohashi (2004) propose paraphrasing the complex predicate structure of "noun-particle-verb" to a simplified verbal predicate using linguistic clues extracted from definition sentences in a dictionary. They focus on identification and paraphrasing of LVCs (*periphrastic phrases* in their term) such as *hinan-o-abiru* "(literally), draw down blame on" and predicates with semantic redundancy (*overlapping phrases* in their term) such as *choking-o-tameru* "(literally), save my savings".

Matsuyoshi and Sato (2008) proposed the paraphrasing of functional expressions such as *yaru-shika-nai* "no choice but to do" to *yarazaru-o-enai* "must do" using a hierarchically organized dictionary of Japanese functional expressions (Matsuyoshi et al., 2007).

These studies of paraphrasing precisely recognize semantically equivalent phrases by using a linguistic resource such as an LCS dictionary. Because the previous studies focus on LVCs and functional expressions, whose variations are not as productive as content words due to their closed-class nature, the coverage of the language resource is not as critical. This is also shown in Chapter 3, in which we constructed a LVC dictionary of high coverage. However, once the focus shifts to predicates with a content word, coverage becomes critical. Several predicate pairs become only synonymous in a certain context (e.g., *break* the rule vs. *ignore* the rule), so one must have language resources that can fully cover these variations.

4.2.2. Distributional Similarities

Distributional similarities are vector based metrics that calculate semantic similarities between words/phrases (Curran 2004; Dagan et al. 1999; Lee 1999; Lin 1998). Following the *distributional hypothesis* (Firth, 1954), which claims that semantically similar words occur in similar contexts, distributional similarities calculate word similarities based on co-occurring words (i.e., contexts).

Szpektor and Dagan(2008) conducted the automatic acquisition of inference rules between predicates such as "X *takes a nap*" and "X *sleeps*" using the unary pattern of a predicate and a variable for calculating distributional similarities. Shibata and Kurohashi (2010) focused on predicates that become synonyms only when combined with a certain argument such as *hiekomu* "get cold" in *keiki-ga-hiekomu*

"Business gets cold feet" and *akka* "get worse" in *keiki-ga-akka* "Business gets worse". Using a predicate-argument structure as a unit, they drew on a huge data resource, 6.9 billion sentences from the Web, to construct vector models for predicates and predicate-argument.

The use of vector-based models can cover various expressions that might not be listed in a thesaurus such as WordNet. Furthermore, it does not require any human-annotated data. However, as has been pointed out by several studies (e.g., Lin et al., 2003; Shibata and Kurohashi, 2010; Yih et al., 2012), there is a problem with distributional similarities. Distributional metrics simply measure the basic associations between words; they cannot represent finer distinctions such as synonymy and antonymy. Lin et al. (2003) constructed a dictionary of antonyms using the pattern such as *either X or Y* and in order to filter out antonyms from distributionally similar words. However, these antonym patterns are characteristic of canonical antonyms (Jones et al., 2007), so cannot guarantee the coverage of the dictionary.

The methods described above are unable to distinguish between not only synonyms and antonyms, but also synonyms and sequential event relations. As has been pointed out by Shibata and Kurohashi (2010), predicates in a sequential event relation such as "put powder (to a brush)" and "apply powder (to your face)" also have high distributional similarities because they tend to share a similar context. The following is an example of surrounding words for *kona-o-toru* "get powder" and *kona-o-tsukeru* "apply powder."

(23) Example of surrounding words of *kona-o-toru* and *kona-o-tsukeru*

a. *kona-o-toru* "get powder"

***burashi-o-tsukau* "use a brush", *pafu-o-tsukau* "use a puff",**

hada-ni-noseru “apply to skin”, *fukuro-ni-ireru* “put in a bag”

b. *kona-o-tsukeru* “apply powder”

burashi-o-tsukau **“use a brush”,** *katachi-o-totonoeru* “adjust the shape”

pafu-o-tsukau **“use a puff”**

As shown, both predicate expressions share the same surrounding contexts of “use a brush” and “use a puff”.

Another problem of distributional similarities is ambiguity. Yih and Qazvinian (2012) combined three different vector models constructed from Wikipedia, snippets of a search engine and WordNet, and averaged distributional similarities calculated in order to suppress the effect of word sense ambiguities. They successfully disambiguate word meanings, such as jaguar as an automobile and that as an animal, by combining vector models constructed from those different language resources. However, because they simply average similarity scores, the problem of distinguishing synonymous words from antonyms remains.

4.2.3. Synonym Recognition Based on Supervised Classification

Hashimoto et al. (2011) proposed the automatic acquisition of paraphrases using supervised classification. Unlike previously introduced studies on paraphrasing, Hashimoto et al. (2011) automatically extract definition sentences from the Web and use those that express the same concept to acquire paraphrases. For example, they extract a pair of paraphrases “makes bone fragile” and “increases the risk of bone fracture” from sentences that define “Osteoporosis”. The identification of definition sentences and that of paraphrasing are conducted by a supervised classification method, which achieved high precision. However, because they use definition sentences, the

paraphrases so obtained are restricted to those that express a certain *concept;* paraphrases that are less likely to appear in definition sentences cannot be obtained (e.g., *eat-sandwich* and *devour-sandwich*).

Hagiwara (2008) performed the supervised acquisition of synonyms by directly using the contextual features used for calculating distributional similarities, in addition to syntactic relations between words. However, contextual features are indeed the same for both synonyms and antonyms, so whether the proposed method is effective for distinguishing synonyms from antonyms is not clear. Furthermore, for predicate phases, contextual features themselves cannot be a clue for detecting synonymous predicates; the important property is the *commonality* in context and not the context itself.

Turney (2008) proposed a unified approach of recognizing synonyms, antonyms and associations. He uses surrounding words as features for supervised classification. Although the method is unique in that it can classify different semantic relations by itself, if one focuses on the task of recognizing semantically equivalent phrases (i.e., synonyms), its overall performance is not as great as the algorithm specific to synonym recognition, as mentioned in Turney (2008).

Weisman et al. (2012) introduce the recognition of entailment relations between verbs such as *snore* and *sleep* using co-occurrence information calculated separately on sentence, paragraph and document levels. They thoroughly analyzed linguistic clues to detect entailment relations, and used distributional models calculated based on unique linguistic features such as verb classes and adverbs to achieved higher performance than previous methods. However, several of the linguistic features used are specific to English, so one cannot directly apply them to Japanese. Furthermore, they only focus on single verbs, so additional linguistic features are needed in order to correctly calculate semantic information of predicate phrases, including functional expressions.

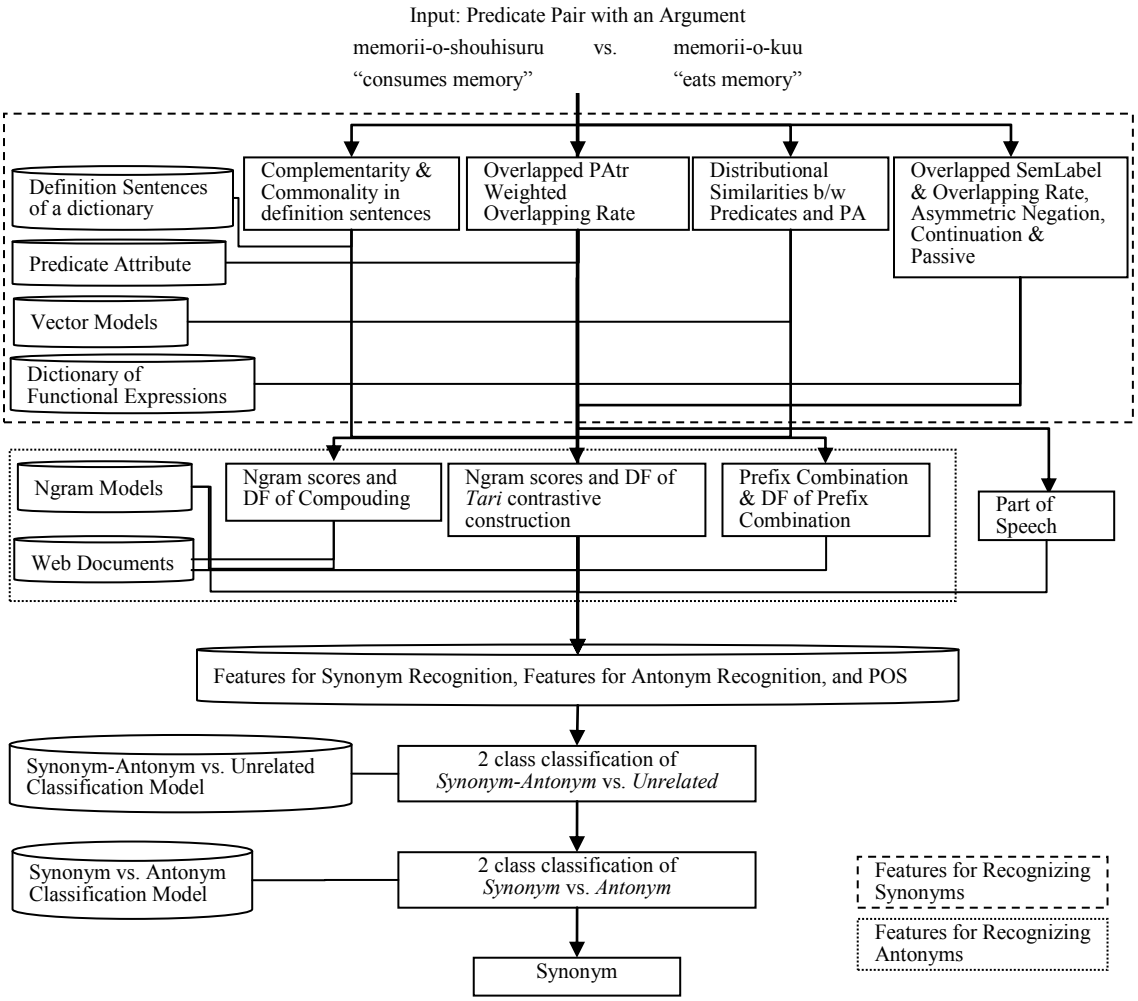

Figure 4.1. *The overall flow of synonym recognition.*

4.3. Proposed Method

We propose the supervised classification of synonymous predicates. In order to correctly recognize semantically equivalent predicates, we construct linguistically-motivated features that reflect the linguistic properties of synonyms and antonyms. These features are summarized in Table 4.1. The overall flow of our synonym recognition is depicted in Figure 4.1.

Table 4.1. *Linguistically-motivated features used in the proposed method*

Features		Description
Recognizing Synonyms	**Definition sentences in a dictionary**	-Binary features indicating whether a predicate appears in the definition sentences of the other predicate -Word overlap among definition sentences between predicate pairs
	Abstract predicate categories	-Predicate categories that the two predicates share - Ratio of overlap in predicate categories
	Distributional Similarities	-Distributional similarities between predicates -Distributional similarities between predicate-argument pairs - The difference between the distributional similarity of predicates and that of predicate-argument structures.
	Modality and Negation	-Modality and Negations that each predicate has - Asymmetric Occurrence of Negation, Continuation, and Passive -Ratio of overlap in Modality and Negations between two predicates
Recognizing Antonyms	**Compounding and the *tari* contrastive construction**	- The Frequency and Ngram scores of the compounding word of predicate pairs - The ngram score of the string in which two predicates are combined by the *tari* conjunct.
	Prefix combination	- The combination of the first character of antonym pair and its Ngram score and frequency.
POS		- Part-of-Speech of each predicate

Run	
Phonetics/Phonology	/r ʌ n/
Morphology	Verb
Syntax/Semantics	Number of Argument:1 Argument 1: Theme Argument 1: Animate
Semantics	+dynamic; -telic continuous directed motion undergone by < 1 >
Lexical-Encyclopedic information	motion involves rapid movement of legs, no continuous contact with ground
Associations (Context)	exercise, boredom, heart attacks

Figure 4.2. *Linguistic structure of the verb "run". (Ramchand, 2010, p.4).*

4.3.1. Features for Recognizing Synonyms

The meaning of a predicate is decided by the combination of different linguistic information as is done for the structure of verbs in Ramchand (2010) (Figure 4.2). We claim that in order to correctly recognize synonymous predicates, one needs to find properties at various linguistic levels; lexical-encyclopedic level, abstract semantic level, discourse level and modality.

4.3.1.1. Definition Sentences

In order to recognize synonymous predicates, we first need to understand the meaning of a predicate (Lexical-Encyclopedic information). As we turn to a dictionary when we encounter a word that we do not know, we use definition sentences from a dictionary for extracting lexical-encyclopedic information. The use of definition sentences for recognizing semantically equivalent phrases has been reported useful by several studies (e.g., Tsuchiya & Kurohashi, 2000; Fujita & Inui, 2001; Kaji et al., 2003)

Upon observing definition sentences of synonymous predicates, we find two important properties. One is that if two predicates are synonyms (e.g., "buy" and "purchase"), one (especially one with broader meaning) tends to occur in the definition sentence of the other. The following is an example of the definition of "purchase" extracted from *Longman Advanced American Dictionary.*

(24) Definition of "purchase": to *buy* something, especially something big or expensive

We call this feature as "complementarity in definition sentences" because one predicate complements the meaning of the other synonymous predicate. We use the binary feature of existence of complementarity in definition sentences.

The next property is that if two predicates are synonymous, their definition sentences are also similar. The following is an example of definition sentences of "high-priced" and "expensive".

(25) "high-priced": Costing a lot of money

(26) "expensive": Costing a lot of money

As shown, both definitions contain exactly the same wording. By extracting commonly used content words, we measure the similarity of definition sentences of two predicates. The following are lexical-encyclopedic features extracted from definition sentences.

- Complemantarity in definition sentences
- Commonality in the content words of two definition sentences.

4.3.1.2. Predicate Attributes

We claim that if two predicates express the same meaning, their abstract semantic class must be the same regardless of the differences in surface forms. For example, the following two synonymous predicates share the same predicate attribute in *Goi-Taikei* (Ikehara et al., 1999).

(27) Predicate Attributes of *kau* "buy" and *kounyuu-suru* "purchase"

Kau "buy" : [*Transfer in possession*], [*Action*]

Kounyu- suru "purchase": [*Transfer in possession*], [*Action*]

Both share the same predicate attributes of *Transfer in possessions* and *Action.*

We use *yougen zousei* "predicate attributes" in *Goi-Taikei* (Ikehara et al, 1999) as features at the semantic level. Because the predicate attributes are hierarchically organized as in Figure 4.3, we extract the following two features.

- Predicate attributes that two predicates share
- Ratio of overlap in predicate attributes

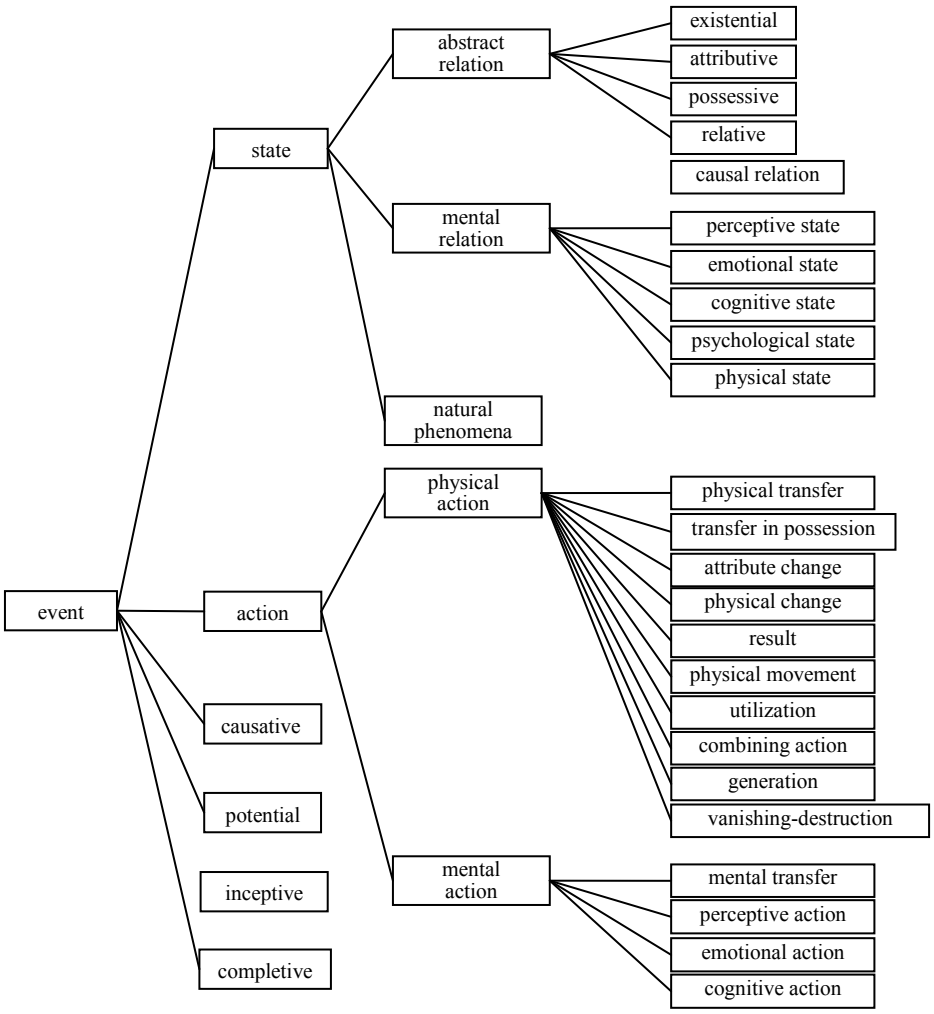

Figure 4.3. *Hierarchical predicate attributes in Goi-Tikei (Ikehara et al., 1999)*

More than one predicate attribute can be assigned to a predicate. Attributes at the lower levels of the hierarchy express more detailed properties than those at the upper level (e.g., "Action (level 2)" vs. "Transfer in Possession (level 4)"). The deeper the shared attribute is, the more similar the two predicates are, so a weighted overlap ratio in predicate attributes is used. The weights are decided heuristically. Level x indicates the level at *Goi-Taikei's* Predicate Attribute Hierarchy (the highest being 1 and the lowest 4). PAtr is for Predicate Attributes.

$$\text{Weighted Overlap of PAtr} = \frac{\begin{array}{c}(|\text{PAtr for Pred1 at Level 1} \cap \text{PAtr for Pred2 at Level1}|*1 \\ + (|\text{PAtr for Pred1 at Level2} \cap \text{PAtr for Pred2 at Level2}|)*1.5 \\ + (|\text{PAtr for Pred1 at Level3} \cap \text{PAtr for Pred2 at Level3}|)*2 \\ +(|\text{PAtr for Pred1 at Level4} \cap \text{PAtr for Pred2 at Level4}|)*2.5\end{array}}{(|\text{PAtr for Pred1} \cup \text{PAtr for Pred2}|)} \quad [4]$$

These features on predicate attributes can capture the similarity between predicates at the semantic level.

4.3.1.3. Distributional Similarities

Distributional similarities of predicates and predicate-argument pairs are used for syntactic and contextual features. We follow Shibata and Kurohashi (2010) and use vector models constructed to calculate the similarities between predicates and those of predicate-argument structures. Shibata and Kurohashi (2010) use predicate-argument structures such as *memorii-o-shouhi* "consume memory" and predicates alone such as *shouhi* as units for calculating distributional similarities. They use words in the dependency relations with the predicate-argument or predicate as features for vector models. The following is an example of the vector features used in Shibata and Kurohashi (2010). (*pre* refers to words that precede the target predicate-argument and *post* refers to those that follow the target predicate-argument.)

(28) Example of vector features used in Shibata and Kurohashi (2010)

Unit (u)	Features (f)
Predicate-Argument "consume-memory"	-Predicates that precede / follow the unit - boot up: *pre* - slowed down: *post*, freeze: *post*
Predicate "consume"	-Arguments and other predicates that are in the dependency relation. -[Arguments] calorie-ACC, fuel-ACC, memory-ACC -[Predicates] burn: *pre,* metabolize: *pre* digest: *post*

Following Curran (2004), *weight* and *measure* functions are used for calculating distributional similarities. *weight* calculates the informativeness of the relation between

the unit and its features based on frequencies. Shibata and Kurohashi (2010) define *weight* as;

- The weight function

$$\text{weight} = \begin{cases} 1 \text{ if MI}>0, \\ 0 \text{ (otherwise)} \end{cases} \quad [5]$$

$$\text{MI} = \log \frac{\text{P(u,f)}}{\text{P(u)P(f)}} \quad [6]$$

MI is calculated by the following formula. P(u) indicates the relative frequency of a unit (i.e., predicate-argument or predicate) and P(f) indicates the relative frequency of features against *u*. P(u, f) indicates their probability of co-occurrence.

For calculating distributional similarities, we use the averaged score of Jaccard and Simpson coefficient measures.

- The *measure* function

$$\text{measure} = \frac{1}{2}(\text{JACCARD} + \text{SIMPSON}) \quad [7]$$

$$\text{JACCARD} = \frac{|(\text{u1},*) \cap (\text{u2},*)|}{|(\text{u1},*) \cup (\text{u2},*)|} \quad [8]$$

$$\text{SIMPSON} = \frac{|(\text{u1},*) \cap (\text{u2},*)|}{\min(|\text{u1},*|,|\text{u2},*|)} \quad [9]$$

where

$$(\text{u},*) \equiv \{f | weight\ (u, f) = 1\} \quad [10]$$

Based on the above formula, we use the following three different kinds of distributional similarity scores as features for syntactic and contextual linguistic information.

- Distributional similarity between predicates
- Distributional similarity between predicate-argument structures

- The difference between the distributional similarity of predicates and that of predicate-argument structures.

4.3.1.4. Modality Information Expressed by Functional Expressions

As has been discussed throughout this paper, Japanese predicates consist of content words and functional expressions. The meaning of a predicate is influenced by not only its content words, but also functional expressions as we have been propounding throughout this thesis. In order to represent semantic information expressed by functional expressions, semantic labels selected by the normalizer in Chapter 2 are used as follows.

- Overlapped semantic labels (Semantic labels that both predicates have)
- Asymmetric Occurrence of Negation, Continuation, and Passive
- Overlap rate of Semantic labels

We set a special flag for the asymmetric constructions of negation, continuation and passive because these functional expressions drastically change the semantic meaning between predicates as in "tall" vs. "not tall." The overlap rate is calculated as follows (Funcs refers to functional expressions and SemLabels refers to Semantic Labels).

$$\text{Overlap in Funcs} = \frac{(|\text{SemLabels of Pred 1} \cap \text{SemLabels of Pred2}|)}{(|\text{SemLabels of Pred 1} \cup \text{SemLabels of Pred2}|)} \quad [11]$$

The features discussed so far represent our linguistically-motivated synonym recognition features at various linguistic levels.

4.3.2. Linguistic Features for Recognizing Antonyms

As has been pointed out by several studies (e.g., Lin et al., 2003; Yih et al., 2012), distinguishing synonymous phrases from antonymous ones is a challenge because most of the linguistic properties for synonyms are shared by antonyms. Antonymous phrases

share a similar context, and their abstract semantic class is also the same. In order to distinguish synonymous phrases from antonyms, one needs linguistic features that are peculiar to antonyms, which we will introduce here.

4.3.2.1. Compounding

As has been discussed in Murphy (2006), antonyms tend to appear in certain constructions. In Japanese, we observed that antonymous phrases tend to make a compound such as such as *uri-kai* (buy and sell) in which the conjunctive form of *uru* "sell" is combined with the conjunctive form of *kau* "buy".

(29) netto-de youfuku-o *uri-kai*-si-teiru
Internet-on outfits-ACC sell-buy-do-CONT(inuous)
"I buy and sell clothes on Internet."

(30) kurikaeshi ***nyuu-taiin***-si-teiru.
constantly in/out-do-CONT(inuous)
"She is constantly in and out of hospital."

The verb *nyuu-taiin-si-teiru* is constructed by combining the verb *nyuuin-suru* "be hospitalized" and the verb *taiin-suru* "leave the hospital". On the other hand, very few synonymous predicates are cast in the compound form.

(31) # Netto-de youfuku-o kai-kounyuu-si-teiru
Internet-on outfits-ACC buy-purchase-do-CONT
" I buy and purchase clothes on Internet."

By automatically generating a compound for each antonym pair, we use the following two features as compounding features.

- Document frequency (df) of the compound calculated from the web
- Ngram score calculated based on Japanese google ngram.

The higher frequency/score of the two compounds is used.

4.3.2.2. The *tari* Contrastive Construction

Similar to compounding, antonymous phrases tend to appear in the *tari* construction, which contrasts different events/actions.

(32) hon-o ut-tari kat-tari dekimasu.
book-ACC sell-*tari* buy-*tari* can
"You can sell and/or buy books here."

First, we automatically generate a string of Pred1-*tari* Pred2-*tari* "do/is Pred1 or do/is Pred2" and the other order of "Pred2-*tari* Pred1-*tari,"* and then use the following as the likelihood of the *tari* contrastive construction occurring. The higher score is selected.

- Ngram score of the string with the *tari*

4.3.2.3. Prefix Combination

Additionally, we use information of the *Kanji* (Chinese characters) in each predicate pair because we observe that *kanji* meaning is an important clue for detecting antonyms. We use the term *prefix* to refer to the first character of a predicate for the sake of description clarity.

(33) 入院 vs. 退院
"enter the hospital" "leave the hospital"

The kanji "入" expresses the action of entering, while the kanji "退" expresses leaving; both are antonyms. This is due to the construction of Japanese *kanji* words; the first

character tends to express the action, while the second tends to expresses the object (the *kanji* "院" means "clinic/hospital," which is the goal of the entering action).

On the other hand, synonymous predicates tend to share the same prefix. For example, the synonymous predicates *出演* "perform" and *出る* "appear" as in *perform/appear on stage* share the same *kanji* of "出." Furthermore, prefix combinations for antonym relations such as *入退* often function as a noun and/or a verb on their own while those for synonym relations do not. In order to represent these properties, the following prefix combination features are used. The prefix combination is constructed by combining the first character of each predicate. The higher ngram score/document frequency is selected.

- Prefix combination of predicate pairs
- Document frequency of prefix combination
- Ngram score of prefix combination
- Overlap Flag in prefix combination (indicating whether prefixes extracted from each predicate are the same)

The features compounding, the *tari* contrastive constructions and prefix combination discussed above represent our linguistically-motivated features peculiar to antonyms.

4.4. Constructing a Corpus of Japanese Predicates for Synonym/Antonym Relations

In order to evaluate the proposed method, we constructed a large human annotated set of predicate pairs for synonym-antonym relations. Because certain predicates become synonymous only with a particular argument (e.g., *break*-rule vs. *ignore*-rule), we assign a noun and an appropriate case marker to each predicate pair.

8.1 million sentences of web blogs were used to extract predicate-argument pairs. All predicates were normalized by our predicate normalizer of Chapter 2. Nouns used for an argument are the 700 most frequent nouns that are categorized as concrete nouns in *Goi-Taikei* (1999).

Predicate-argument pairs in the relations of synonyms, entailment, antonyms and unrelated were manually extracted by human annotators. In order to make the data as consistent as possible, we create several linguistic tests based on Chierchia and McConnel-Ginet (2000). For simplicity, we use the term Predicate A and Predicate B to refer to a predicate pair. Examples are listed in parenthesis. # indicates a semantically wrong sentence.

- Synonym (Mutual Entailment)

  Definition: Predicate A (*repair*) and Predicate B (*fix*) denote the same event. (If the event expressed by Predicate A is true, the event expressed by Predicate B is also true and vice versa.)

  Test: Negating only one of the predicates results in a contradictory fact (i.e., does not make sense).

  Example: # I *repaired* my pc, but I did *not fix* it.

- Entailment

  Definition: If the event denoted by Predicate A (*snore*) is true, the event denoted by Predicate B (*sleep*) is also true, but not vice versa.

  Test: Negating only Predicate B does not make sense. However, the opposite is possible.

  Example: # I *snored* last night, but I did *not sleep*.

  I *slept l*ast night, but I did *not snore.*

- Antonym

  Definition: If the event denoted by Predicate A (*long*) is true, the event denoted by Predicate B (*short*) must be false.

  Test: Predicate A and Predicate B cannot be combined by the conjunction "but" in a sentence.

  Example: # His legs are *long,* but they are *short.*

If a predicate-argument pair does not follow any of the tests above, it is categorized as *unrelated.*

The annotation was conducted as follows. The first annotator, who has a solid background in linguistics, extracted predicate-argument pairs for synonyms, entailment, antonyms and unrelated relations from the data and the first evaluator, who also has a solid background in linguistics, evaluated the predicate-argument pairs extracted by the first annotator. If the annotator and the evaluator disagreed, they discussed and selected the appropriate relation. After the first annotation and evaluation, two annotators checked the first data and modified it if it did not follow the criterion. The following shows the total number of predicate-argument pairs constructed and examples of the data, which was checked by the total of four annotators/evaluators.

(34) Synonymous Predicates (2,843 pairs)

Kuruma-ga-butsukat-teita	Kuruma-ga-shoutotushiteita
Car-Nom-bumped	Car-Nom-crashed
"A car was bumped."	"A car was crashed"
Basu-ga-hassyashita	Basu-ga-syuppatsushita
Bus-Nom-departed	Bus-Nom-started

“The bus departed.”	“The bus started.”

(35) Predicates in Entailment Relation (2,368 pairs)

Tokei-o-chekkusita	Tokei-o-mita
Watch-ACC-checked	Watch-ACC-looked
“(I) checked the watch.”	“(I) looked at the watch.”
Niwa-o-sannsakusita	Niwa-o-aruita
Garden-ACC-strolled	Garden-ACC-walked
“(I) strolled the garden.”	“(I) walked through the garden.”

(36) Antonyms (2,227 pairs)

Kuruma-ga-juutaishiteita	Kuruma-ga-nagareteita
Car-NOM-jammed	Car-NOM-running
“Cars were jammed.”	“Cars were running smoothly.”
Basu-o-orita	Basu-ni-notta
Bus-ACC-get off	Bus-ACC-get on
“(I) got off a bus”	“(I) got on a bus”

(37) Unrelated (4,948 pairs)

Kuruma-ga-butsukatteita	Kuruma-o-tomeru
Car-ACC-bumped	Car-ACC-stopped
“A car was bumped.”	“(I) stopped a car.”
Syokupan-ni-hasanda	Shokupan-ga-yaketa
Bread-between-put	Bread-Nom-made
“(I) put (s/th) in between bread”	“Bread was made”

4.5. Experiment

Using the data in Section 4.4, we evaluated our proposed method.

4.5.1. Resources

For extracting definition sentences in a dictionary and for extracting predicate attributes, *Gakken Japanese Dictionary (2nd Ed.)* of Kindaichi and Ikeda (1988) and *Goi-Taikei* of Ikehara et al. (1999) were used. In order to calculate distributional similarities between predicates and those between predicate-argument structures, the vector models used in Shibata and Kurohashi (2010), constructed from 6.9 billion sentences on the Web, were used. In order to reduce the influence of low frequent words, we only used features that occurred at least 10 times in the corpus. Semantic labels of predicates were extracted by our normalizer proposed in Chapter 2. For calculating ngram sores and frequency of occurrences for compounding and the *tari* contrastive construction, we used the Japanese google ngram and the document frequency (df) of the compound calculated from the web, respectively. In order to correctly evaluate the effect of our linguistically-motivated features, we only used predicates that were listed in the above language resources, which reduced the total number of predicate-argument pairs to 3,875. The following shows the total number of predicate-argument pairs used in training.

- Synonym (956 pairs)
- Entailment (669 pairs)
- Antonym (758 pairs)
- Unrelated (1120 pairs)

4.5.2. Training

372 pairs of predicate-argument structures were used to analyze linguistic features for recognizing synonymous predicates, and the rest (3,503 pairs) were used as a training set. For the current task, we grouped predicates for synonym and entailment relations together and used them as positive examples, while predicates in antonyms and unrelated relations were used as negative examples.

For training, we used LIBSVM (Chang & Lin, 2011) with radical basis function (RBF) kernel. Because synonym-antonym relations are similar compared to synonym-unrelated relations, we conducted two-step classification. First, the synonym and antonym classes were grouped together, and then the classification of synonym-antonym class against the unrelated class. Next, classification of synonymous predicates against antonymous predicates was conducted. Evaluation was by five-fold cross validation.

4.5.3. Baselines

Four different methods were compared. The first baseline uses the Japanese WordNet (Bond et al., 2009), one of the largest thesauruses. If the synonymous predicate pairs are listed in the synsets in WordNet, they are counted as correct. The other baselines are based solely on similarity measures. Baseline 2 (DistPAVerb-θ) uses distributional similarities alone and simply extract predicate pairs with the scores above a threshold. Following Yih and Qazvinian (2012), Baseline 3 (DistMultiAve-θ) averages distributional similarity scores calculated separately from the different language resources used in the proposed method (Web corpus, definition sentences in a dictionary, and predicate attributes in *Goi-Taikei*). Pairs whose similarity score exceeded a threshold were classified as synonymous predicate pairs. For Baselines 2

and 3, the threshold was decided based on the training corpus. Five-fold cross validation was conducted and the threshold that provided the best F-score for the training corpus was used.

Baseline 4 uses only distributional similarity scores as features for supervised classification. This baseline was structured to evaluate the effectiveness of our linguistically-motivated features.

- WordNet (BL1): If predicates pairs are in synset relations of WordNet, they are synonymous.
- DistPAVerb-θ (BL2): If the similarity score of predicate pairs is above the threshold, they are synonymous.
- DistMultiAve-θ (BL3): If the averaged similarity score obtained from different vector models is above the threshold, they are synonymous.
- DistPAVerb-SVM (BL4): Use distributional similarity scores as features for supervised classification of synonymous predicates.

The results are evaluated based on Precision (Prec), Recall (Rec) and F-score (F). Five-hold cross validation was performed and the averaged score used for comparison.

$$\text{Precision} = \frac{\#\ of\ True\ Synonymous\ Preds \cap \#\ of\ Preds\ Classified\ as\ Synonymous}{\#\ of\ Preds\ Classified\ as\ Synonymous} \quad [12]$$

$$\text{Recall} = \frac{\#\ of\ True\ Synonymous\ Preds \cap \#\ of\ Preds\ Classified\ as\ Synonymous}{\#\ of\ True\ Synonymous\ Preds} \quad [13]$$

$$\text{F} - \text{score} = \frac{2 * Prec * Rec}{(Prec + Rec)} \quad [14]$$

Table 4.2. *Results of experiment*

	Precision	Recall	F (** p < 0.01)
BL1(WordNet)	**0.873**	0.331	0.480**
BL2(DistPAVerb-θ)	0.621	0.778	0.688**
BL3(DistMultiAve-θ)	0.537	**0.946**	0.685**
BL4(DistPAVerb-SVM)	0.677	0.834	0.747**
Proposed	0.843	0.891	**0.866**
Synonyms Features Only	0.794	0.822	0.808**
Antonyms Features Only	0.645	0.807	0.717**

4.5.4. Results of Experiment

The results of the experiment are shown in Table 4.2. The proposed method achieved the highest F-score of 0.866 and its difference from the baseline methods is statistically significant.[21]

BL1 (WordNet) achieved the highest precision of 0.873, but the lowest recall of 0.311. BL3 (DistMultiAve-θ) provided the highest recall but the lowest precision. BL 2 (DistPAVerb-θ) achieved the low precision of 0.621. The use of supervised classification (BL 4, DistPAVerb-SVM) improved the overall F-score, but the proposed method offered an further improvement of more than 10 points. Although the features for recognizing synonyms on their own (Synonyms Features Only) achieved a higher F-score than the baselines (0.808 of F-score), the combination of synonyms features and antonym features improved the overall precision by up to 6 points.

[21] We conducted a *t-test* using F scores.

Table 4.3. *Results of ablation test*

		Acc	Synonyms			Antonyms			Unrelated		
			Prec	Rec	F	Prec	Rec	F	Prec	Rec	F
Proposed		84.9	0.843	0.891	0.866	0.736	0.644	0.686	0.806	0.802	0.804
Synonymous Features Only		76.2	0.794	0.822	0.808	0.712	0.518	0.599	0.743	0.840	0.788
Antonymous Features Only		66.0	0.645	0.807	0.717	0.767	0.551	0.637	0.641	0.522	0.574
w/o	Distributional Similarity	74.9	0.790	0.791	0.790	0.801	0.623	0.700	0.676	0.775	0.722
	Functional Expressions	78.3	0.807	0.847	0.826	0.772	0.562	0.649	0.760	0.840	0.798
	Definition Sentence	81.0	0.833	0.875	0.853	0.816	0.635	0.713	0.777	0.833	0.804
	Predicate Attributes	81.7	0.849	0.883	0.865	0.834	0.625	0.713	0.769	0.851	0.807
	Prefix	79.2	0.825	0.855	0.839	0.774	0.598	0.674	0.761	0.835	0.796
	Compouding	81.4	0.844	0.873	0.858	0.805	0.645	0.715	0.781	0.843	0.810
	Tari Construction	80.3	0.836	0.860	0.848	0.785	0.616	0.689	0.768	0.847	0.805
	POS	81.6	0.846	0.871	0.858	0.827	0.637	0.719	0.773	0.858	0.813

Table 4.4. *Features ordered by effectiveness*

	Classification of Synonymous Predicates	Classification of Antonymous Predicates
1	Distributional Similarity	Functional Expressions
2	Functional Expressions	Prefix
3	Prefix	Tari Construction
4	Tari Construction	Distributional Similarity
5	Definition Sentences	Definition Sentences
6	Compounding	Predicate Attributes
7	POS	Compounding
8	Predicate Attributes	POS

In order to analyze which feature was the most effective for the current task in detail, we conducted an ablation test (Table 4.3), in which each feature was removed in turn. We list the overall accuracy of three-class classification as well as precision, recall, and f-score of each class (synonym, antonym, unrelated) in order to analyze the effectiveness in detail. The results indicate that distributional similarities were the most effective for classifying synonymous predicates while functional expressions were the most effective for classifying antonymous predicates. The effectiveness of the features is ordered in Table 4.4.

4.6. Discussion

Although the use of WordNet yielded the highest precision, it suffered from low recall. The following are examples of synonymous predicates that WordNet could not find in synsets.

(38) memori -o -shouhisiteiru vs. memori -o -kutteiru
memory -ACC -comsume-CONT memory ACC -eat-CONT
"It's consuming memory." "It's eating memory."

(39) ranchi -o sumaseta vs. ranchi -o tabeta
lunch -ACC -done lunch -ACC -ate
"(I'm) done with lunch." "(I've) had lunch."

Predicates such as "eats" and "consume" become synonymous with the argument "memory." WordNet tends not to include predicates in synsets that become synonyms in a certain context, which degrades recall.

Since BL2, BL3, and BL4 relied on only distributional similarities they demonstrated the lowest precision although the use of supervised learning worked effectively to improve precision (BL4). The cause of low precision is due to the fact that distributional similarities gave higher scores to not only synonymous predicates but also those having some semantic associations. The following shows examples that are incorrectly recognized as synonyms by distributional similarity based methods.

(40) Basu -o -orita vs. Basu -ni -notta
Bus -ACC -get off Bus -DAT -get on
"got off the bus" "got on the bus"

(41) Kona -o -toru vs. Kona -o -tsukeru
powder -ACC -get powder -ACC -apply
"get powder" "apply powder"

The above predicate pairs are in an antonym relation and a sequential event relation, respectively. The distributional similarities failed to distinguish these relations and wrongly categorized them as synonymous.

The proposed method uses various linguistic features including those for recognizing synonyms and those for antonyms, and successfully classified synonymous predicates, which greatly improved the precision while achieving high recall. Features specifically introduced for antonyms were very effective and improved the overall F-score by up to 6 points. The following is an example that is correctly recognized as an antonym, which, without antonym features, would be wrongly categorized as synonyms (Synonymous Features Only).

(42) sake -ga -nakunatta vs. sake -ga -nokotteita
alcohol -NOM -gone alcohol -NOM - remained
"The alcohol was gone." "The alcohol remained."

The proposed method also correctly captures semantic information conveyed by functional expressions, and succeeded in understanding the complex semantic structure of predicate phrases. The following are predicate pairs that were correctly classified as synonyms by the proposed method.

(43) sozai -ga -tsukae-souda vs. sozai -ga -waruku-nai-kamoshirenai
material -NOM -seems useful material-NOM -might not be bad
"The material seems useful" "The material might not be that bad"

(44) pajama -o -mot-tei-nai vs. pajama -ga -nai
pajama -ACC -have-not pajama -NOM -not
"I don't have pajamas." "There are no pajamas."

Table 4.5. *List of extracted synonymous predicates*

dezitaru-ni-*kaeru* "*change to* a digital s/th"	dezitaru-ni-*ikousuru* "*swich to* a digital s/th"
ninki-ga-*joushousuru* "popularity *rises*"	ninki-o-*takameru* "*improve* popularity"
undou-ga-*kirai* "*hate* exercise"	undou-ga-*darui* "exercise *is dull*"
kyoku-o-*kaku* "*write* a song"	kyoku-o-*tsukuru* "*create* a song"
warai-ga-tae-nai "laughter never stops"	warai-ni-afureru "full of laughter"
oudishon-de-*ketteisuru* "*decided* by the audition"	ondeishon-de-*erabu* "*selected* by the audition"
sekai-ga-kagiri-nai "world *is infinite*"	sekai-ga-hiroi "world *is huge*"
risuningu-ga-wakan-nai "*don't understand* listening comprehension questions"	risuningu-ga-deki-nai "*can't solve* listening comprehension questions"
asa-kara-omoi "*feel heavy* since the morning"	asa-kara-sugure-nai "*not feel good* since the morning"
kyousou-ga-gekika "the competition *gets aggravated*"	kyousou-ga-hagesii "the competition is intense"

By retaining the crucial meaning of functional expressions such as modality and negation, we succeeded in dealing with complex semantic information of predicates.

An error analysis revealed that the proposed method often failed to classify synonymous predicates with idiomatic meanings.

(45) kane -ga -tobu vs. kane -ga -kakaru
money -NOM -files money -NOM -costs
"(literally), Money flies." "It costs money."

(46) fude -ga -omoi vs. fude -ga -susuma-nai
pen -NOM -heavy pen -NOM -go smoothly-not
"(literally), my pen is heavy." "(literally), my pen doesn't go smoothly."

The expression *kane-ga-tobu* metaphorically indicates the condition of money's rapid disappearance, and it means "to lose money." Both *fude-ga-omoi* and *fude-ga-susumanai* shows the state in which one, who is supposed to write something,

struggles with writing. These idiomatic expressions need more sophisticated rules of inference. One possible solution would be to use how these expressions are translated into a foreign language because these idiomatic expressions might be translated into the same phrase as direct word-to-word translation is avoided for idiomatic expressions. The analysis of idiomatic expressions and their translations is for future study.

In order to further evaluate the proposed method, we extracted synonymous predicate pairs from a raw corpus (3.3 years of blogs) and automatically classified them into synonyms, antonyms and unrelated. Examples of extracted synonymous predicates are shown in Table 4.5. A great many of the extracted predicates were synonymous expressions that an ordinary thesaurus would not have, such as *warai-ga-tae-nai* "laughter never stops" and *warai-ni-afureru* "full of laughter." We can say this shows the promise of the automatic acquisition of synonymous predicates.

4.7. Conclusion of Chapter 4

In Chapter 4, we proposed the supervised classification of synonymous predicates in Japanese. Using linguistically-motivated features for recognizing synonymous predicates, we succeeded in automatically classifying semantically equivalent predicate phrases. These predicates include not only typical synonymous expressions such as "buy" and "purchase," but also those that become synonymous only when combined with a certain argument. The proposed method yielded a drastic increase in recall compared to the simple method of using a thesaurus.

Furthermore, the proposed method correctly distinguished synonymous predicates from antonymous ones, which remains a challenge for distributional similarity based methods. By capturing the key meanings of functional expressions, the proposed method also successfully handled the complex semantic structure of

synonymous predicates. This is shown in automatically extracted predicate phrases, which demonstrated wide variations in expressions. We believe that the current study will shed light on automatic thesaurus acquisition, let alone increasing the overall performance of NLP systems such as text mining and information retrieval.

CHAPTER 5

CONCLUSION

In this thesis, we have proposed a novel approach for identifying semantic similarities between complex predicate phrases based on linguistically-motivated evidence, whose underlying principles are universal although several of clues are tuned specifically to the Japanese language.

Focusing on three fundamental problems for understanding predicate meanings, namely morphological variations, syntactic variations and semantic variations, we have proposed three distinct algorithms. The first proposes normalization of functional expressions in which only expressions that are crucial for eventual meanings are sustained. Using the paraphrasing rules constructed based on linguistic theories in syntax and semantics and deleting unnecessary expressions, we succeeded in sustaining the meaning of a predicate at the higher accuracy of 79.7% while improving the overall performance of predicate extraction task, making it possible to compute a complex meaning of functional expressions such as modality, tense and negation.

The second proposes paraphrasing of complex light verb constructions into a simplified verbal predicate. LVCs are complex in that a predicative meaning of predicate is conveyed by the noun and the verb itself merely works as verbalizing the noun. The existence of LVCs often causes incorrect identification of predicative meanings. By constructing 10 paraphrasing rules of simplifying LVCs as well as a comprehensive dictionary of noun-verb pairs for LVC disambiguation, we succeeded in simplifying complex LVC structures at the high accuracy (89.7% for newspapers, 93.0% for blogs) as well as improve the recall of the predicate phrase identification.

The last proposes a supervised classification of synonymous predicate phrases. Unlike functional expressions and LVCs, content words in predicates are polysemous, showing a great number of ambiguities in meaning. Previous studies of similarity-based metrics tend to simply assign similarity scores to semantically associated predicates and cannot precisely distinguish synonymous predicates from antonyms. By combining different features at various linguistic levels such as lexical-encyclopedic level, and abstract semantic levels, with those that are peculiar to semantically *opposite* phrases, we have succeeded in recognizing semantically similar predicate phrases at the high accuracy of 0.87, which satisfies the requirement for actual NLP applications. The classification model constructed further provides us with a promising result of automatic extraction of synonymous predicate phrases from raw data.

In conclusion, this thesis proposed a novel approach of understanding the semantic similarity of complex predicate phrases. All the methods introduced in this thesis provide us with a promising result of their usefulness in NLP applications. Deeply understanding predicate meanings will certainly lead to a robust improvement of natural language tasks, and we believe that this thesis will contribute to a deeper understanding of natural language from huge and diverse text data, which is an emerging valuable source of knowledge.

BIBLIOGRAPHY

Adger, D. (2003). *Core syntax: A minimalist approach.* New York: Oxford University Press.

Aizawa, A. (2008). Daikibo tekisuto koopasu-o mochiita go-no ruizido kenkyuu-ni kansuru kousatsu. [On calculating word similarity using large text corpora]. *Journal of Information Processing (IPSJ), vol. 49, 3,* 1426-1436.

Bond, F, Isahara, H, Fujita, S, Uchimoto, K, Kuribayashi, T and Kanzaki, K (2009). Enhancing the Japanese WordNet. *Proceedings of the 7th Workshop on Asian Language Resources, ACL-IJCNLP 2009,* 1-8.

Brun, C., and Hagège, C. (2003). Normalization and paraphrasing using symbolic methods. *Proceedings of the second international workshop on Paraphrasing: Paraphrase acquisition and applications (IWP)*, 41-48.

Butt, M (2003). The light verb jungle. *Harvard Working Papers in Linguistics. Vol.9,* 1-28.

Chang, C-C., and Lin, C-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2,* 27:1-27:27.

Chierchia, G., and McConnell-Ginet, S. (2000). *Meaning and grammar: An introduction to semantics (2nd ed.).* Cambridge, MA: The MIT press.

Cinque, G. (2006). *Restructuring and functional heads: The cartography of syntactic structures, Vol. 4*. New York: Oxford University Press.

Cruse, D. A., (1986). *Lexical Semantics.* New York: Cambridge University Press.

Curran, J.R. (2004). *From distributional to semantic similarity.* Doctoral dissertation, University of Edinburgh, Edinburgh, United Kingdom.

Dagan, I., Lee, L., and Pereira, F. C. N. (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning - Special issue on natural language learning, 34(1-3),* 43-69.

Dowty, D. (1979). *Word meaning and Montague grammar: The semantics of verbs and times in generative semantics and in Montague's PTQ.* Dordrecht: D. Reidel.

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

Fazly, A, and Stevenson, S. (2007). Distinguishing subtypes of multiword expressions using linguistically-motivated statistical measures. *Proceedings of the ACL Workshop on A Broader Perspective on Multiword Expressions,* 9-16.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. In J.R. Firth et al. (Eds.). *Studies in Linguistic Analysis, Special volume of the Philological Society,* pp. 1-32. Oxford: Blackwell.

Fujita, A., Furihata, K., Inui, K., Matsumoto, Y., and Takeuchi, K. (2004). Paraphrasing of Japanese light-verb constructions based on lexical conceptual structure. *Proceedings of Second ACL Workshop on Multiword Expressions: Integrating Processing,* 9-16.

Fujita, A., and Inui, K. (2001) Gosyakubun-o riyoushita futsuumeishi-no dougainengo-eno iikae [Paraphrasing nouns of the same concept using definition sentences.]. *Proceedings of the 7th annual meaning of the association for natural language processing,* 331-334.

Group Jamacy (2008). *Kyoosito gakusyuusyano tameno nihongo bunkei jiten [Dictionary of Japanese grammar constructions for teachers and learns of Japanese]*. Tokyo: Kuroshio.

Hagiwara, M. (2008). A supervised learning approach to automatic synonym identification based on distributional features. *Proceedings of the ACL-08: HLT Student Research Workshop (Companion Volume),* 1-6.

Han, C-H., and Rambow, O. (2000). The Sino-Korean light verb construction and lexical argument structure. *Proceedings of the 5th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 5),* 221-226.

Harabagiu, S., Hickl, A., and Lacatusu, F. (2006). Negation, contrast, and contradiction in text processing. *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-06).* 755-762.

Hashimoto, C., Torisawa, K., De Saeger, S., Kazama, J., and Kurohashi, S. (2011). Extracting paraphrases from definition sentences on the web. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics,* 1087-1097.

Haugh, M. (2008). Utterance-final conjunctive particles and implicature in Japanese conversation. *Pragmatics 18 (3),* 425-451.

Hong, G., Lee, S-W., Rim, H-C. (2009). Bridging morpho-syntactic gap between source and target sentences for English-Korean statistical machine translation. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers,* 233-236.

Ikehara, S., Miyazaki, M., Shirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y., and Hayashi, Y. (1999). *Goi-Taikei [A Japanese lexicon]*. Tokyko:Iwanami.

Imamura, K., Izumi T., Kikui, G., and Sato, S. (2011). Jutsubu kinouhyougen-no imiraberu tagaa [Semantic label tagging to functional expressions in predicate phrases]. *Proceedings of the 17th Annual Meeting of the Association for Natural Language Processing,* 308-311.

Imamura, K., Saito, K., and Izumi, T. (2009). Discriminative approach to predicate-argument structure analysis with zero-anaphora resolution. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 85-88.

Inui, K., Abe, S., Hara, K., Morita, H., Sao, C., Eguchi, M., Sumida, A., Murakami, K., and Matsuyoshi, S. (2008). Experience mining: Building a large-scale database of personal experiences and opinions from web documents. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 1.,* 314-321.

Inui, K., and Fujita, A. (2004). Iikaegijutsu-nikansuru kenkyuudoukou [A survey on paraphrase generation and recognition]. *Journal of Natural Language Processing, 11 (5),* 151-198.

Izumi, T., Imamura, K., Kikui, G., and Sato, S. (2010). Standardizing complex functional expressions in Japanese predicates: Applying theoretically-based paraphrasing rules. *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications,* 63-71.

Jackendoff, R. S. (1992). *Semantic structures,* Cambridge, MA: The MIT press.

Jones, S., Paradis, C., Murphy, M. L., and Willners, C. (2007). Googling for 'opposites': A web-based study of antonym canonicity. *Corpora, 2 (2),* 129-154.

Kaji, N., and Kurohashi, S. (2004). Ugenhyogen-to Juufukuhyogen-no iikae [Recognition and paraphrasing of periphrastic phrases and overlapping phrases]. *Journal of Natural Language Processing, 11 (1)*, 83-106.

Kaji, N., Kawahara, D., Kurohashi, S., and Sato, S. (2003). Kakufureemu-no taiouzuke-nimotozuku yougen-no iikae [Predicate paraphrasing based on case frame alignment]. *Journal of Natural Language Processing, 10 (4),* 65-81.

Kato, S. (2007). Nihongo-no jutsubu-kouzou to kyoukaisei [Predicate complex structure and morphological boundaries in Japanese]. *The annual report on cultural science, Vol. 122(6)*. Hokkaido University Graduate School of Letters, Sapporo, Japan, 97-155.

Kindaichi, H. (1976). *Nihongodousi-no asupekuto [Aspect in Japanese verbs]*. Tokyo: Mugishoboo.

Kindaichi, H., and Ikeda, Y. (1988). *Gakken kokugo daijiten (2nd Ed.)[Gakken Japanese dictionary]*. Tokyo: Gakusyuu Kenkyuusha.

Lee, J., Lee, D., and Lee, G. G. (2006). Improving phrase-based Korean-English statistical machine translation. *Proceedings of the Ninth International Conference on Spoken Language Processing,* 753-756.

Lee, L. (1999). Measures of distributional similarity. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, 25-32.

Lin, D. (1998). An information-theoretic definition of similarity. *Proceedings of the Fifteenth International Conference on Machine Learning,* 296-394.

Lin, D., Zhao, S., Qin, L., and Zhou, M. (2003). Identifying synonyms among distributionally similar words. *Proceedings of the 18th International Joint conference on Artificial Intelligence (IJCAI-03),* 1492-1493.

Matsumoto, Y. (1996). A syntactic account of light verb phenomena in Japanese. *Journal of East Asian Linguistics, 5,* 107-149.

Matsuyoshi, S., and Sato, S. (2008). Automatic paraphrasing of Japanese functional expressions using a hierarchically organized dictionary. *Proceedings of the 3rd International Joint Conference on Natural Language Processing*, 691-696.

Matsuyoshi, S., Sato, S., and Utsuro, T. (2006). Compilation of a dictionary of Japanese functional expressions with hierarchical organization. *Proceedings of*

the 21st International Conference on Computer Processing of Oriental Languages, Lecture Notes in Computer Science, Vol. 4285, 395-402.

Matsuyoshi, S., Sato, S., and Utsuro, T. (2007). Nihongo kinouhyougenjisyono hensan [A dictionary of Japanese functional expressions with hierarchical organization]. *Journal of Natural Language Processing, Vol. 14 (5),* 123-146.

Maynard, S. K. (1997). *Japanese communication: Language and thought in context.* Honolulu, HI: University of Hawai'i Press.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM, Vol. 38, 11,* 39-41.

Minami, F. (1993). *Gendai nihongobunpou-no rinkaku [Introduction to modern Japanese grammar]*. Tokyo: Taishuukan.

Mitchell, J., and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive Science, Vol. 34, 8,* 1388-1429.

Mohammad, S., Dorr, B., and Hirst, G. (2008). Computing word-pair antonymy. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* 982-991.

Mohammad, S., Dorr, B., Hirst G., and Turney, P. (2013). Computing lexical contrast. *Computational Linguistics, 39(3),* 555-590.

Muraki, S. (1991). *Nihongo doushi-no syosou [Aspects of Japanese verbs.* Tokyo: Hitsuji Shobou.

Murphy, L. (2006). Antonym as lexical constructions; or, why paradigmatic construction is not an oxymoron. *Constructions, Special Volume 1,* 1-37.

Nakau, M. (1976). Tense, aspect, and modality. In Shibatani, M.(Ed.), *Syntax and semantics (Vol.5): Japanese generative grammar,* pp. 421-482, Boston, MA: Academic Press.

Narrog, H. (2005). On defining modality again. *Language Sciences, 27,* 165-192.

Nasukawa, T. (2001). Kooru sentaa-niokeru tekisuto mainingu [Text mining application for call centers]. *Journal of Japanese society for Artificial Intelligence, 16(2),* 219-225.

Nasukawa, T. (2009). Text analysis and knowledge mining. *Proceedings of 8th International Symposium on Natural Language Processing,* 1-2.

Nasukawa, T., and Nagano, T. (2001). Text analysis and knowledge mining system. *IBM Systems Journal, 40, 4,* 967-984.

Nitta, Y., and Masuoka, T. (1989). *Nihongo-no Modality [Modality in Japanese language]*. Tokyo: Kuroshio.

Oku, M. (1990). Nihonbun kaiseki-niokeru zyutugosoutouno kanyouteki hyougenno atukai [Analysis methods for Japanese idiomatic predicates]. *Transactions of Information Processing Society of Japan, Vol. 31(12),* 1727-1734.

Partee, B.H., Meulen, A., and Wall, R.E. (1990). *Mathematical methods in Linguistics.* Dordrecht, The Netherland: Kluwer.

Portner, P. (2005). *What is Meaning? Fundamentals of Formal Semantics.* Malden, MA: Blackwell.

Ramchand, G. C. (2010). *Verb meaning and the lexicon: A first-phase syntax.* New York: Cambridge University Press.

Rizzi, L. (1999). *On the position "Int(errogative)" in the left periphery of the clause.* Ms., Università di Siena.

Sato, S. (2007). Kihonkanyouku gosyu taisyouhyou no sakusei [Compilation of a comparative list of basic Japanese idioms from five sources]. *IPSJ SIG Technical Reports, Information Processing Society of Japan,* 1-6.

Shibata, T., and Kurohashi, S. (2010). Bunmyaku-ni izonshita jutsugono dougikankei kakutoku. [Context-dependent synonymous predicate acquisition]. *Information processing society of Japan, Special Interest Group of Natural Language Processing (IPSJ-SIGNL) Technical Report,* 1-6.

Shirai, S., Ikehara, S., and Kawaoka, T. (1993). Effects of automatic rewriting of source language within a Japanese to English MT system. *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation: MT in the Next Generation,* 226-239.

Shudo, K., Tanabe, T., Takahashi, M., and Yoshimura, K. (2004). MWEs as non-propositional content indicators. *Proceedings of second Association for Computational Linguistics Workshops on Multiword Expressions: Integrating Processing,* 32-39.

Spenader, J., and Stulp, G. (2007). Antonymy and contrast relations. *The 7th International Workshop on Computational Semantics,* 1-12.

Stevenson, S., Fazly, A., and North, R. (2004). Statistical measures of the semi-productivity of light verb constructions. *Proceedings of Second ACL Workshop on Multiword Expressions: Integrating Processing,* 1-8.

Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. *Proceedings of the 7th International conference on Spoken Language Processing,* 901-904.

Szpektor, I., and Dagan, I. (2008). Learning entailment rules for unary templates. *Proceedings of the 22nd International Conference on Computational Linguistics*, 849-856.

Takeuchi, K., Inui, K., and Fujita, A. (2006). Goigainenkouzou-nimotozuku nihongodousi-no tougo imitokusei-no kijutu [Description of syntactic and

semantic structures of Japanese verbs based on lexical conceptual structures]. In T. Kageyama (Ed.), *Lexicon Forum No. 2*, pp. 85-120, Tokyo: Hitsuji Shoboo.

Tanabe, T., Yoshimura, K., and Shudo, K. (2001). Modality expressions in Japanese and their automatic paraphrasing. *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium,* 507-512.

Tsuchiya, M., and Kurohashi, S. (2000). MDL genri-nimotozuku zisyoteigibun-no assyuku-to kyoutuusei-no hakken [Compression of description sentences and discovery of common features based on MDL principle]. *Information processing society of Japan, Special Interest Group of Natural Language Processing Technical Report,* 47-54.

Tsujimura, N. (2007). *An introduction to Japanese linguistics (2nd Ed.)*. Malden, MA: Blackwell.

Turney, P. (2008). A uniform approach to analogies, synonyms, antonyms and associations. *Proceedings of the 22nd International Conference on Computational Linguistics,* 905-912.

Turney, P., Littman, M., Bigham, J., and Shnayder, V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. *Proceedings of the International Conference on Recent Advances in Natural Language Processing,* 482-489.

Wang, Y., and Ikeda, T. (2008). Translation of the light verb constructions in Japanese-Chinese machine translation. *Advances in Natural Language Processing and Applications, Research in Computing Science, 33*, 139-150.

Weisman, H., Berant, J., Szpektor, I., and Dagan, I. (2012). Learning verb inference rules from linguistically-motivated evidence. *Proceedings of the 2012 Joint*

Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 194-204.

Yih, W-T., and Qazvinian, V. (2012). Measuring word relatedness using heterogeneous vector space models. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* 616-620.

Yih, W-T., Zweig, G., and Platt, J. (2012). Polarity inducing latent semantic analysis. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning,* 1212-1222.

Yokote, K., Tanaka, S., and Ishizuka, M. (2011). Effects of using simple semantic similarity on textual entailment recognition. *Proceedings of Text Analysis Conference PASCAL Recognizing Textual Entailment Challenges (RTE-7).*

Yoon, J-H. (1994). Korean verbal inflection and checking theory. In C. Philip and H. Harley Eds., *MIT Working Papers in Linguistics: The Morphology-Syntax Connection,* pp. 251-270, Cambridge, MA: Department of Linguistics, MIT.

APPENDIX

In this appendix, we provide a list of predicate phrases that were extracted only when normalized predicates were used in Pair 1 of Experiment 3.3.

NORMED	χ^2 *score*
wakat-ta "understood"	12.5
hiraka-nai "can't open"	12.5
kie-ta "vanished"	11.1
oku-rare-ta "was sent"	11.1
hyouzisa-re-ta "was displayed"	9.7
setuzokusi-te-iru "connecting"	9.7
kensakusi-ta "searched"	9.7
de-te-iru "being displayed"	*9.5*
riyousi-te-iru "using"	*8.5*
imi-ka "does this mean? (the verb *mean* plus the question marker *ka*)"	*8.5*
rakusatusi-ta "won a bid"	*8.3*
de-nai "not displayed"	*7.2*
*donoyounisuru-to-yoi-no-darou-ka "*how should I do?"	*6.9*
yat-ta "did (casual)"	*6.9*
riyousi-tai "want to use"	*6.9*
kaisetusi-tai "want to set up"	*6.9*
daunroodosi-ta "downloaded"	*6.9*
todoka-nai "haven't received"	*6.9*
setteisi-te-iru "setting up"	*6.9*
koto-darou-ka "might mean"	*6.9*
kae-ta "changed"	*6.9*
kat-ta "bought"	*6.9*

LIST OF PUBLICATIONS

Journals

Izumi, T., Shibata, T., Saito, K., Matsuo, Y., and Kurohashi, S. (2013). Recognizing semantically equivalent predicate phrases based on several linguistic clues [In Japanese]. *Journal of Natural Language Processing, Vol.20, 4,* 539-561.

Izumi, T., Imamura, K., Saito, K., Asami, T., Kikui, G., and Sato, S. (2013). Normalizing complex functional expressions in Japanese predicates: Linguistically-directed rule-based paraphrasing and its application. *ACM Transactions on Asian Language Information Processing (TALIP), Vol.12, No.3,*11:1-11:20.

Imamura, K., **Izumi, T.**, Sadamitsu, K., Saito, K., Kobashikawa, S., and Masataki, S. (2013). Morpheme conversion using discriminative models for connecting different morphological systems [In Japanese]. *IEICE Transactions on Information and Systems, Vol.J96-D, 1,* 239-249.

Izumi, T., Imamura, K., Kikui, G., Fujita A., and Sato, S. (2011) Paraphrasing Japanese light verb constructions: Towards the normalization of complex predicates. *International Journal of Computer Processing of Languages, Vol. 23, 2,* 147-167.

Izumi, T., and Sato, Y. (2008). Japanese referring expressions and Accessibility Theory : The examination of a cross-linguistic applicability of Accessibility Theory, *Journal of Hokkaido University of Education, Humanities and Social Sciences, 58,* 101-114

Book Chapter

Izumi, T. (2010). "Why can't you omit *ni*?" : The role of the Japanese case marker ni in interpretation of utterances [In Japanese]. M. Minami (Ed.). *Gengo-gaku to Nihongo-kyooiku VI,* pp. 47-63, Tokyo: Kuroshio.

International Conferences and Workshops

Imamura, K., **Izumi, T.**, Sadamitsu, K., Saito, S., Kobashikawa, S., and Masataki, K. (2012). Morpheme conversion for connecting speech recognizer and language analyzers in unsegmented languages. *Proceedings of Interspeech 2011,* 1405-1408.

Izumi, T., Imamura, K, Kikui, G., and Sato, S. (2010). Standardizing complex functional expressions in Japanese predicates: Applying theoretically-based paraphrasing rules. *Proceedings of the Workshop on Multiword Expressions: From theory to applications (MWE 2010),* 63-71.

Izumi, T., Imamura, K., Kikui, G., Fujita, A., and Sato S. (2010). Paraphrasing Japanese light verb constructions: Towards the normalization of complex predicates. *Proceedings of International Conference on Computer Processing of Oriental Languages 2010,* 55-62.

Imamura, K., **Izumi, T.**, and Saito, S. (2009). Discriminative approach to predicate-argument structure analysis with zero-anaphora resolution. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers,* 85-88.

Izumi, T. (2008). "Why can't you omit *ni*?" : The role of the Japanese case marker ni in interpretation of utterances [In Japanese]. *6th International Conference on Practical Linguistics of Japanese,* San Francisco State University.

Domestic Conferences

Izumi, T., Shibata, T., Saito, K., Matsuo, Y., and Kurohashi, S. (2013). Classifying antonym relations based on lexical and contextual information in Japanese [In Japanese]. *IPSJ SIG Technical Report, Vol.2013-NL-212,* No.1.

Izumi, T., Wong B., Saito, K., and Matsuo, Y. (2013). Acquiring synonymous predicates using bilingual corpora [In Japanese]. *Proceedings of the 19th Annual Meeting of the Association for Natural Language Processing,* 600-603.

Nakamura, H., **Izumi, T.**, Shibata, T., and Kurohashi, S. (2012). Identifying synonymous predicates based on distributional similarities of content words and function words. *Proceedings of the 18th Annual Meeting of the Association for Natural Language Processing,* 413-416.

Imamura K., **Izumi T.**, Kikui, G., and Sato, S. (2011). Semantic label tagging to functional expressions in predicate phrases [In Japanese]. *Proceedings of the 17th Annual Meeting of the Association for Natural Language Processing,* 308-311.

Izumi, T., Imamura, K., and Kikui, G. (2010). Complementing functional expressions to incomplete intermediate predicates in coordinate structures [In Japanese]. *Proceedings of the 16th Annual Meeting of the Association for Natural Language Processing,* 752-755.

Izumi, T., Imamura, K., Kikui, G., Fujita, A., and Sato, S. (2009). Paraphrasing light verb constructions towards the normalization of complex predicates [In Japanese]. *Proceedings of the 15th Annual Meeting of the Association for Natural Language Processing,* 264-267.

Fujita, A., Sato, S., **Izumi, T.**, Imamura, K., and Kikui, G. (2009). Example-based identification of paraphrases between light-verb construction and verb phrase [In

Japanese]. *Proceedings of the 15th Annual Meeting of the Association for Natural Language Processing,* 268-271.