NAIST-IS-DD0261023

Doctoral Dissertation

Automatic Generation of Syntactically Well-formed and Semantically Appropriate Paraphrases

Atsushi Fujita

March 2005

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Atsushi Fujita

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Shunsuke Uemura	(Co-supervisor)
Professor Kiyohiro Shikano	(Co-supervisor)
Associate Professor Kentaro Inui	(Co-supervisor)

Automatic Generation of Syntactically Well-formed and Semantically Appropriate Paraphrases*

Atsushi Fujita

Abstract

Paraphrases of an expression are alternative linguistic expressions conveying the same information as the original. Technology for handling paraphrases has been attracting increasing attention due to its potential in a wide range of natural language processing applications; e.g., machine translation, information retrieval, question answering, summarization, authoring and revision support, and reading assistance. In this thesis, we focus on lexical and structural paraphrases in Japanese, such as lexical and phrasal replacement, verb alternation, and topicalization, which can be generated relying on linguistic knowledge only.

First, we address how to generate well-formed and appropriate paraphrases. One of the major problems is that it is practically impossible to take into account all sorts of semantic and discourse-related factors which affect the well-formedness and appropriateness of paraphrases. The knowledge, such as transformation rules, used for paraphrase generation tends to be underspecified, and thus would produce erroneous output. The revision process is introduced to detect and correct ill-formed and inappropriate candidates generated in the transfer stage. Within this framework, we first investigate what types of errors tend to occur in lexical and structural paraphrasing, and confirm the feasibility of our transfer-and-revision framework by revealing that most errors occur irrespective of classes of transformation rules. On the basis of another observation; that errors associated with case assignments form one of the major error types, we develop a model for detecting this type of error. The model utilizes a large collection of positive examples and a small collection of negative ones by combining supervised and unsupervised machine learning methods. Experimental results indicate that our model significantly outperforms conventional models.

The second issue is to develop a mechanism that is capable of covering a wide variety of paraphrases. One way of gaining the coverage of paraphrase generation is to exploit the systemicity underlying several classes of paraphrases, such as verb alternation and compound noun decomposition. To capture the semantic properties required for generating these classes of paraphrases, we utilize the Lexical Conceptual Structure (LCS). The framework represents verbs as semantic structures with focus of statement and relationships between semantic arguments and syntactic cases. We implement a paraphrase generation model which consists of a case assignment rule and a handful of LCS transformation rules, with particular focus on

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0261023, March 2005.

paraphrases of light-verb constructions (LVCs). An experiment demonstrates that our model generates paraphrases of LVCs accurately, showing advantages over conventional approaches to this class of paraphrases.

In this thesis we build up the following contributions. First, there has been no comprehensive investigation into errors across different paraphrase classes made so far, although various case studies have been done on paraphrase generation. Therefore, our attempt provides a basis for further research on handling transfer errors. We then address the most imperious type of errors occurring in paraphrase generation, and prove the feasibility of our over-generation plus filtering framework. What we argue here is how to use negative examples effectively. Finally, we propose a lexical-semantics-based account of a subset of lexical paraphrases. This is novel approach to represent paraphrases via a linguistically-motivated semantic representation.

Keywords:

Automatic Paraphrasing, Paraphrase Generation, Transfer Errors, Case Assignment, Lexical Conceptual Structure

Acknowledgments

First of all, I would like to express my sincere gratitude to Professor Yuji Matsumoto, my supervisor, for giving me a chance to study at the Computational Linguistics Laboratory (CL-Lab.) in Nara Institute of Science and Technology (NAIST). He gave me a lot of advice derived from his vast range of experiences, and taught me how to enjoy life in various aspects.

I am profoundly obliged to Associate Professor Kentaro Inui for his continuous encouragement throughout my research life. Since we had met in 1998 at Kyushu Institute of Technology (Kyutech), he gave me advice on numerous areas from his broad range of experiences, even during his 10-month stay in the United Kingdom. Discussion with him fostered my significant interest in this avenue of research; natural language generation and paraphrasing. I also learned various things from him, such as the proper attitude of a researcher and the delicateness of natural language expressions in human communication. I was quite fortunate to have the opportunity to study under him.

I would like to thank Professor Shunsuke Uemura and Professor Kiyohiro Shikano for reviewing this dissertation. They pointed out the inadequacy of explanations in the former version of this dissertation and gave me valuable comments to improve it.

I am deeply grateful to Assistant Professor Edson Tadashi Miyamoto (currently at University of Tsukuba), Assistant Professor Masashi Shimbo, and Assistant Professor Masayuki Asahara in the CL-Lab. They gave me a lot of helpful advice whenever I was struggling, as well as giving me brotherly encouragement.

During the absence of Associate Professor Inui, I had various opportunities to discuss with some active researchers. Associate Professor Satoshi Sato of Kyoto University and Assistant Professor Kazuhide Yamamoto of Nagaoka University of Technology invited me to their meeting on recent work encompassing paraphrasing. Dr. Koichi Takeuchi, Lecturer of Okayama University, and Dr. Kyo Kageura, Associate Professor of National Institute of Informatics, gave me valuable comments on the work in Chapter 5. Associate Professor Manabu Okumura of Tokyo Institute of Technology gave me a lot of valuable comments as well as encouraging me. I greatly appreciate their kindness in allowing me to join their high-level discussion.

I spent 3 happy years in the comfortable atmosphere of the CL-Lab. At laboratory meetings, students and research staffs gained insights into my work and gave me a lot of suggestions. Particularly, I am grateful to Dr. Hiroya Takamura, Dr. Taku Kudo, and Dr. Daichi Mochihashi for their comments from a statistical viewpoint. Miss Motoko Okawa, the secretary, was willing to assist all of my administrative needs. The first half of my research life, which I spent in Kyutech before coming to NAIST, was also quite enjoyable, because I have been a member of the original Inui-Lab. Dr. Hiroko Otsuka, Dr. Takashi Inui, Miss Satomi Yamamoto, and Miss Ryoko Tokuhisa have set good examples for my research life. Every activity with them led to invaluable experiences. Dr. Tetsuro Takahashi, Mr. Tomoya Iwakura, and Mr. Ryu Iida are my best colleagues and friends. Even now, they continue to discuss various things with me, and help me to stay on the right path. I would like to sincerely thank all the members of the CL-Lab. and the Inui-Lab.

During a 3-month stay at Xerox Research Centre Europe (XRCE), many people aided me in all respects. Dr. Kenji Yamada is the person who gave me a chance to work in the forefront of statistical machine translation. It was truly an incentive to work with him and the members of the machine learning team of the Content Analysis group, Dr. Pierre Isabelle, Dr. Eric Gaussier, Dr. Cyril Goutte, and Dr. Nicola Cancedda. Ph.D. candidates, Mr. Franck Guingne, Mr. Florent Nicart, and Mr. Aurélien Max, and administrators, Mrs. Angèle Buttin and Mrs. Karima Del Olmo constantly offered me support. Residents in *la Houille Blanche* welcomed me like family, and taught me the importance of relaxation in life. I appreciate their kindness.

I am indebted to Ian Fookes who offered his help to proofread this dissertation. His correction improved both this dissertation and my English skill.

Last but not least, I am sincerely thankful to my parents and maternal relatives. My parents allowed me to leave them and pursue my own goals, and my maternal relatives gave me the financial support that allowed me to concentrate on study and research.

Contents

List of Figures

List of Tables

Abbreviations

Glosses for the Japanese examples cited in this thesis contain the following abbreviations:

ABL	Ablative case ("*kara*")
ACC	Accusative case ("*o*")
ACT	Active
ADJ	Adjective
ADV	Adverb
CAUS	Causative
COM	Comitative case ("*to*")
COP	Copula
DAT	Dative case ("*ni*")
GEN	Genitive case ("*no*")
HON	Honorific
IMP	Implement case ("*de*")
LCS	Lexical Conceptual Structure
LOC	Locative case ("*de, ni*")
LVC	Light-verb construction
N	Noun
NEG	Negation
NLP	Natural Language Processing
NOM	Nominative case ("*ga*")
PASS	Passive
PAST	Past tense
PERF	Perfective
PRES	Present
PROG	Progressive
QUAN	Quantifier
TOP	Topic
V	Verb

CHAPTER 1 *Introduction*

1.1 Paraphrases

A common characteristic of human languages is the capability of expressing the same information in multiple ways. Paraphrases, which in the literature have also been referred to as variants, reformulations, or inference rules, span a wide range of phenomena.

In this thesis, we focus on a certain group of paraphrases, **lexical and structural paraphrases**, which can be carried out without referring to the communicative situation, but with linguistic knowledge. Let us introduce some examples. Example (1) exhibits[1] a lexical replacement. It can be carried out using the knowledge about synonymous words. Example (2) illustrates a more structural transformation, thus a more general pattern of paraphrases. Such syntactic transformations have been discussed at length in the linguistics literature, such as the theory of transformational grammar (Inoue, 1976; Harris, 1981). Our target also includes phenomena which require both of the above two types of transformation. Example (3) demonstrates an alternation, in which a word is replaced according to a syntactic transformation, which changes the focus of the statement.

(1) s. Tom **purchased** a Honda from John.
    t. Tom **bought** a Honda from John.

(2) s. **It was** a Honda **that** John **sold to** Tom.
    t. John **sold** a Honda **to** Tom.

(3) s. Tom **bought** a Honda **from** John.
    t. John **sold** a Honda **to** Tom.

The following examples do not fall into our focus. Example (4) exhibits a **referential paraphrase**. The expression "last year" in (4s) refers to 2004, because the sentence is written now in 2005. However, the expression must refer to 2005 if the sentence is uttered in 2006, and thus (4t) is inappropriate. In other words, the meaning (referent) of the expression varies according to the situation of utterance.

(4) s. They got married **last year**.
    t. They got married **in 2004**.

[1]For each example, "s," "t," and "r" denote an original, its paraphrase, and manually revised versions of paraphrase, respectively. The mark "$*$" indicates that the expression is either syntactically ill-formed or semantically inappropriate (See also Section 1.3.1). The mark "$\neq$" denotes that the paraphrased expression does not convey the same meaning, particularly denotation, as those of a given expression (See also Section 1.3.2).

Example (5) exemplifies a more special case, i.e., a **pragmatic paraphrase**. Both sentences in (5) can convey "the speaker would like the hearer to open the window." They are the same in terms of *perlocutionary act* in pragmatics. Since the perlocutionary act of a sentence also varies according to the communicative situation, we do not handle them in this thesis.

(5) s. **I want some fresh air.**

t. **Could you open the window?**

In example (6), the source sentence (6s) could entail the statement in (6t). In some applications, such an **entailment** has been treated as a paraphrase (Lin and Pantel, 2001; Ravichandran and Hovy, 2002). However, sentences having this relation do not necessarily convey the same meaning; e.g., (6t) is no longer true, since Himawari 5 has stopped in May 2003.

(6) s. Himawari 5, a meteorological satellite, **was launched in 1995**.

t. Himawari 5, a meteorological satellite, **is in operation**.

To have an intuition about what sorts of paraphrases are included in our target, we show a classification of lexical and structural paraphrases in Chapter 2.

1.2 Automatic paraphrasing

The diversity of linguistic expressions presents a major challenge for application of natural language processing (NLP). Thus, research on automatic paraphrase generation, recognition, and acquisition has been recently attracting increasing attention, and has benefited a broad range of NLP tasks (Association for Natural Language Processing, 2001; Sato and Nakagawa, 2001; Inui and Hermjekob, 2003). For example, in cross-lingual machine translation (henceforth, machine translation), paraphrasing has been applied to pre-editing of the target sentence, post-editing of the generated sentence, similar sentence retrieval for example-based methods, and automatic evaluation for statistical methods (Shirai *et al*., 1993; Kim and Ehara, 1994; Nyberg and Mitamura, 2000; Yamamoto, 2002a; Kanayama, 2003; Shimohata, 2004; Finch *et al*., 2004). Besides that, paraphrasing has been introduced into term expansion and flexible query-content matching in information retrieval (Jacquemin *et al*., 1997; Anick and Tipirneni, 1999; Shiraki and Kurohashi, 2000; Lin and Pantel, 2001), query expansion and answer verification in question answering (Ravichandran and Hovy, 2002; Hermjakob *et al*., 2002; Duclaye *et al*., 2003; Takahashi *et al*., 2003), and single- and multi-document summarization (Robin and McKeown, 1996; McKeown *et al*., 1999; Barzilay *et al*., 1999; Mani *et al*., 1999; Nanba and Okumura, 2000; Barzilay, 2003).

Potential application is not limited to existing NLP tasks alone but also for humans. Automatic paraphrasing technology could bridge gaps between authors and readers. Given a text, for example, a system that is capable of simplifying it, or showing users several alternative expressions conveying the same content, would be useful for assisting readers (Carroll *et al*., 1999; Canning and Tait, 1999; Inui and Yamamoto, 2001; Inui *et al*., 2003; Siddharthan, 2003).

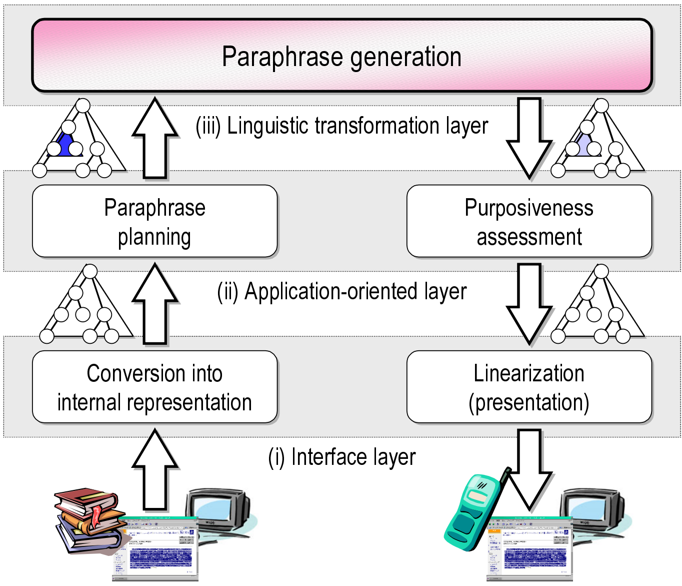

Figure 1.1. A generic architecture of paraphrasing.

Likewise, technology of paraphrasing can be used for supporting document authoring by controlling vocabularies and styles (Takahashi and Ushijima, 1991; Takeishi and Hayashi, 1992; Hayashi, 1992; Inkpen *et al.*, 2004; Kaji *et al.*, 2004).

Paraphrasing can be viewed as a special case of translation in the sense that both transform a given expression into a different expression, while preserving meaning as much as possible. Due to these similarities, one may think that automatic paraphrasing can be carried out simply by employing state-of-the-art technologies of machine translation such as those described in (Brown *et al.*, 1993; Dorr, 1993; Wahlster, 2000; Lavoie *et al.*, 2000; Melamed, 2001; Carl and Way, 2003). However, the application-oriented nature of paraphrasing is more salient than that of machine translation: we need to evaluate purposiveness of the resultant paraphrases according to application-specific criteria. In reading assistance, for example, it is critical that a paraphrase of an input sentence or text actually improves its comprehensibility (Inui and Yamamoto, 2001).

The issue is how to control paraphrasing to achieve a given purpose for a given input. To cope with this demand, we propose a paraphrasing architecture consisting of the following three layers (c.f., Figure 1.1):

(i) Interface layer provides an interface between input and output strings and their internal representations. In our present architecture, all intermediate structures used by the components are represented uniformly as morpho-syntactic dependency structures which can be annotated with semantic and textual information on demand.

(ii) Application-oriented layer covers subtasks related to application-oriented purposiveness.

On the source side, it produces a paraphrasing plan that specifies what class of linguistic transformation should be applied to which part of the input. On the target side, it assesses the purposiveness of each of given paraphrase candidates and selects the optimal one.

(iii) Linguistic transformation layer performs linguistic transformations, the core application-independent subtask of paraphrasing which generates morpho-syntactically well-formed and meaning-preserving paraphrases from a given input coupled with a paraphrase plan.

If this layered architecture works properly, one should be able to build a paraphrase generation system that is reasonably application-independent and can thus be used for different applications. Although it is unclear if one can fully eliminate application-dependent processes in linguistic transformation, we believe that it is important, at least in the preliminary stages of the research, to make a maximum effort to avoid confounding linguistic and purposive appropriateness of paraphrases. On the basis of this architecture, we address the practical issues to realize a linguistic transformation mechanism throughout this thesis. Henceforth, we call the linguistic transformation in this layer **paraphrase generation**.

1.3 What qualifies as a paraphrase and what doesn't

What criteria need to be satisfied for the generated sentence to be considered as paraphrases? To determine the quality of a given paraphrase, we introduce in this section some concepts for evaluating satisfactory paraphrases.

1.3.1 Syntactic well-formedness and semantic appropriateness

Paraphrase generation can be viewed as a special case of natural language generation in the sense that the outputs are natural language expressions. Thus, paraphrases must be **syntactically well-formed** and **semantically appropriate**[2].

First of all, let us consider the following examples:

(7) s. My son **broke the window**

t. **The window broke**.

(8) s. My son **destroyed the window**

t. ***The window destroyed**.

Example (7) illustrates a transitivity alternation for the verb "break:" the verb is used as a transitive verb in sentence (7s), while as an intransitive verb in sentence (7t). In contrast, for the transitive verb "destroy" in sentence (8s), the same transformation produces an unacceptable

[2]In fact, paraphrases are merely regarded as the intermediate forms in a certain application of paraphrase generation, such as machine translation and information retrieval. In such cases, syntactic well-formedness and semantic appropriateness of paraphrases can be negligible.

sentence (8t). Because "destroy" does not behave as an intransitive verb. As a consequence, (7t) is judged to be syntactically well-formed, while (8t) to be syntactically ill-formed.

Even when paraphrased sentences are syntactically well-formed, they are sometimes unacceptable. Let us consider the following examples:

(9) s. The farmer **raised** three children.

t. The farmer **brought up** three children.

(10) s. The farmer **raised** vegetables.

t.*The farmer **brought up** vegetables.

Both sentences (9t) and (10t) are syntactically well-formed because "bring up" behaves as a transitive verb. However, "bring up vegetables" is semantically inappropriate because "bring up" does not take an inanimate object. When the object is animate like "children" or "calves," both "raise" and "bring up" can dominate them, thus such a sentence is semantically appropriate. These exemplify the difference of **selectional restrictions** of verbs "raise" and "bring up." Similarly, "a handsome woman" and "a pretty man" are semantically inappropriate due to violation of selectional restrictions of nouns, while "a pretty woman" and "a handsome man" are semantically appropriate.

1.3.2 Sameness of meaning

The primary property required for paraphrases is to convey the same meaning as that of the given expression. Taking the following examples, we firstly introduce two different grades of meaning: **denotation** and **connotation**.

(11) s. Your cat looks **slim**.

t. Your cat looks **skinny**.

(12) s. We can control growth of cancer by inhibiting **genes**.

t.≠We can control growth of cancer by inhibiting **factors**.

Both sentences (11t) and (12t) are syntactically well-formed and semantically appropriate. However, they should be distinguished because of differences in the meanings they convey relative to their source sentences. In example (11), the sentences give the same description of the same object, but the nuances of their respective descriptions are slightly different; (11t) seems to be less-favorable. The sentences in example (12), on the other hand, convey a different meaning altogether; (12s) notes what affect to "growth of cancer" are "genes," while (12t) does not. The term "factor" denotes a vague (non-specific) concept whereas "gene" is a specific term.

The description of the concept in the world is called **denotation**, while the implicit meaning, such as nuance and style, is called **connotation** (Edmonds, 1999; Inkpen, 2003). Throughout this thesis, we allow the difference of connotation involved when generating paraphrases[3]. Thus, (11t) is judged to be correct, while (12t) to be incorrect because it conveys different denotation from that of (12s).

Dras (1999) argued that structural paraphrasing generally causes some change of emphasis (or constructional meaning). For example, both sentences in example (3) convey that "the owner of a Honda changed from John to Tom." However, their viewpoint is different because the subject is changed. Example (13) demonstrates a more delicate difference. The sentences in example (13) convey the same fact "a man painted the wall," but with a different connotation. In the sentence (13s), the object "paint" is accentuated, but it remains unclear as to whether only part or whole of the wall has been painted. Sentence (13t), on the other hand, strongly suggests that the entire wall has been painted because "wall" is the focal object of the sentence.

(13) s. He sprayed **paint on the wall**.

t. He sprayed **the wall with paint**.

We regard such changes in meaning, which are caused by syntactic transformation, as a difference in connotation, and therefore allow them in paraphrase generation.

1.4 The aims of automatic paraphrase generation

This section describes several issues in paraphrase generation and our position on them.

1.4.1 Implementation

The architecture of automatic paraphrasing generation has been implemented in various ways. Transfer-based approaches include those based on the dependency structure (Kurohashi and Sakai, 1999; Kondo *et al*., 2001; Kaji *et al*., 2001; Takahashi *et al*., 2001; Pang *et al*., 2003), argument structure (Mel'čuk and Polguère, 1987; Meteer and Shaked, 1988; Huang and Fiedler, 1996; Lavoie *et al*., 2000; Brun and Hagège, 2003), and a theory of lexical semantics, Lexical Conceptual Structure (Jackendoff, 1990; Kageyama, 1996; Dorr, 1997; Takeuchi *et al*., 2002). Although a handful of example-based, and statistics-based approaches have been proposed (Barzilay, 2003; Dolan *et al*., 2004), they do not seem feasible because paraphrase examples are generally not available. As several attempts have been made (Shirai *et al*., 2001; Kinjo *et al*., 2003; Shimohata, 2004), it is, in fact, possible to accumulate a reasonable size of paraphrase examples if the domain of text collection is limited or if a particular purpose is assumed. However, from an analytical point of view, this approach still seems unsuitable for determining

[3] Bond and Fujita (2003) divided semantic similarity into three classes. In `same or close` the paraphrased sentence has almost the same meaning as the original; in `different nuance` the meaning is significantly broader or narrower, or only the same in certain contexts; in `different` the meaning changes in the paraphrased examples. In our definition, the first two classes of paraphrases are allowable since the denotation is preserved.

what knowledge is required for paraphrase generation, and exploring paraphrase phenomena themselves.

For the last half decade, researchers have enthusiastically tackled the issue of automatic acquisition of paraphrases from parallel, comparable, and non-parallel corpora (Barzilay and McKeown, 2001; Lin and Pantel, 2001; Shimohata and Sumita, 2002a; Shimohata and Sumita, 2002b; Torisawa, 2002; Pang *et al*., 2003; Shinyama and Sekine, 2003; Ohtake and Yamamoto, 2003; Quirk *et al*., 2004). One of the findings obtained in this previous work is that automatic paraphrase acquisition is feasible for various types of data sources, and that the extracted paraphrases are not only pairs of lexical and phrasal expressions, such as ⟨"burst into tears," "cried"⟩ (Barzilay and McKeown, 2001; Shimohata and Sumita, 2002b; Torisawa, 2002), but also those of syntactic patterns, such as ⟨"*N1* is solved by *N2*," "*N2* finds a solution to *N1*"⟩ (Lin and Pantel, 2001; Shinyama and Sekine, 2003; Ohtake and Yamamoto, 2003). The current fashion implies that state-of-the-art paraphrase generations uses a transfer-based approach. We therefore assume this type of implementation for discussion in this thesis.

1.4.2 Generation of well-formed and appropriate paraphrases

One of the goals of paraphrase generation is to develop a mechanism which is capable of producing syntactically well-formed and semantically appropriate paraphrases for given expressions. In achieving this goal, one of the major difficulties is to determine the applicability conditions of knowledge used for paraphrase generation. Regardless of the implementation level of transformation, it is practically impossible to incorporate all sorts of semantic and discourse-related factors which affect the well-formedness and appropriateness of paraphrases. This is because we cannot examine *every possible context* where an expression may appear.

Let us suppose that a paraphrase (9t) is generated by applying a **transfer rule**[4], such as "raise" ⇒ "bring up", to (9s). Then to avoid generating a semantically inappropriate sentence (10t) from (10s), one may be able to specify applicability conditions for the transfer rule, such as allowable classes of nouns for the direct object of a given verb. For example, in the case of the rule above, animacy (animate or inanimate) of the object noun may seem to be useful information. In general, however, it is not obvious as to under which conditions a given expression can be paraphrased. Moreover, determining all such conditions would not only be time-consuming and costly, but invariably makes transfer rules very complicated.

Hence transfer rules tend to be underspecified in their applicability conditions, and thus they are likely to produce morpho-syntactically ill-formed or semantically inappropriate expressions for the source expression. Moreover, even if the paraphrased expressions do not involve errors, the meaning is occasionally changed from those of the source expressions. Henceforth, we refer to this kind of ill-formedness, inappropriateness, and changes of meaning as **transfer errors**.

To avoid transfer errors, we follow the state-of-the-art technology of machine translation and natural language generation. In natural language generation, for example, candidates can be generated with loose constraints and then ranked by employing a language model (Knight

[4] In this thesis, we use the term "transfer rule" to refer to a pair of expressions used for paraphrase generation such as "raise" ⇒ "bring up," which indicates that the rule replaces the word "raise" with the phrase "bring up."

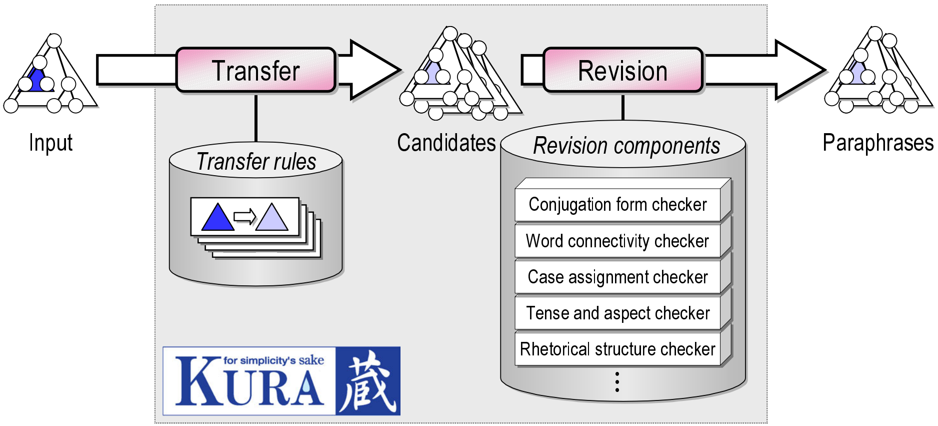

Figure 1.2. KURA: a transfer-and-revise framework for paraphrase generation.

and Hatzivassiloglou, 1995; Beale *et al*., 1998; Langkilde and Knight, 1998; Bangalore and Rambow, 2000). This approach is also state-of-the-art in machine translation (Brown *et al*., 1993; Habash and Dorr, 2002; Germann *et al*., 2004) and other fields. Similarly, we propose a transfer-and-revise model which decomposes the process of paraphrase generation into the following two subprocesses (Takahashi *et al*., 2001) (c.f., Figure 1.2):

Transfer: Apply transfer rules to an input and generate a set of **paraphrase candidates**.

Revision: Detect transfer errors in each paraphrase candidate, and resolve them if possible; otherwise, reject the candidate.

In the transfer process, we leave the description of each transfer rule **underspecified**. In the revision process, on the other hand, we implement a revision component which utilizes linguistic constraints that are independent of particular transfer rules. There should be a wide range of transfer-independent linguistic constraints. Typical examples are constraints on morpheme connectivity, verb conjugation, word collocation, and tense and aspect forms in relative clauses.

Since there is no clearcut criterion for transfer-independence, one may want to employ linguistic constraints on the transfer stage. In statistical machine translation, for example, the models allow the use of constraints (e.g., translation dictionaries for technical terms) to reduce the calculation cost. Yet, we believe that the division of transfer and revision processes makes knowledge representation simpler by eliminating excessive constraints, and thus contributes to pick up potential candidates.

In the framework of transfer-and-revise, one of the issues is how to develop a robust method to detect and correct transfer errors in the post-transfer process by way of a safety net.

1.4.3 Generation of a wide variety of paraphrases

Another critical issue that we face in paraphrase generation is how to develop and maintain knowledge resources that cover a sufficiently wide range of paraphrases: such as those indicating that "*N1* be *V*-PERF by *N2*" ⇒ "*N2 V N1*,"[5], "to make an attempt" ⇒ "to attempt," and "potential" ⇒ "possibility." There are complementary approaches to knowledge development: manual construction and automatic acquisition from corpora.

Several attempts have been made to manually develop resources, such as transfer rules and dictionaries containing lexical information, by exploring particular classes of lexical and structural paraphrases. They cover a wide range of paraphrases, such as lexical and phrasal replacement (Fujita and Inui, 2001; Lapata, 2001; Pearce, 2001; Kaji *et al.*, 2001), verb alternations (Levin, 1993; Kageyama, 1996; Kondo *et al.*, 2001; Baldwin *et al.*, 2001; Kageyama, 2002), topicalization (Sunagawa, 1995; Dras, 1999), and decomposition of compounds (Sato, 1999; Takeuchi *et al.*, 2002).

Previous studies provided varieties of transfer rules and lexical knowledge useful for paraphrase generation. However, the knowledge tended to be specialized to a particular class of paraphrases. Therefore, the issues of covering wide varieties of paraphrases are: (i) to explore what sorts of knowledge can be widely used for paraphrase generation across different paraphrase classes, and (ii) to establish a knowledge construction method which enables complementary use of both linguistic theories and automatically acquired knowledge.

1.5 Research directions

Some issues are crystallized by the decomposition of the paraphrase generation process into transfer and revision processes. In this thesis, we address the following two issues:

- Handling transfer errors in paraphrase generation
- A lexical-semantics-based approach to paraphrase generation

Here we note that our target language is Japanese. English examples are used for explanatory purposes only.

Handling transfer errors in paraphrase generation

We predicted that various types of transfer errors occurring in paraphrase generation can be solved by introducing post-transfer revision processes. The revision process utilizes a wide range of transfer-independent linguistic constraints. On that account, we make a comprehensive investigation into transfer errors across several classes of lexical and structural paraphrases. In the investigation, we address the following issues:

- To explore what sorts of transfer errors tend to occur in paraphrase generation and which of them should be prioritized.
- To develop the technology to detect and correct errors in the post-transfer revision process.

As a test bed of error exploration, we choose an implementation of morpho-syntactic transfer provided by KURA[6] (Takahashi *et al.*, 2001), because it offers an environment which en-

[5]Here, *Nx* and *V* denote nominal and verbal elements, respectively (c.f., **Abbreviations**).
[6]http://cl.naist.jp/kura/ParaCorpus/.

ables quick development of various transfer rules.

A lexical-semantics-based approach to paraphrase generation

There are a variety of lexical and structural paraphrases. Among them, several classes of paraphrases require expressions to be paired as in “kick the bucket” ⇒ “die,” and “potential” ⇒ “possibility.” On the other hand, other classes exhibit a considerable degree of productivity, such as verb alternations and compound noun decomposition. Examples (14) and (15) indicates that verbs “break” and “increase” can be used as both transitive and intransitive, and thus the alternation between these forms are general transformation.

(14) a. She **broke the cup** into pieces.

b. **The cup broke** into pieces.

(15) a. The government certainly **increased the tax on alcohol**.

b. **The tax on alcohol** certainly **increased**.

Productivity allows them to be systematically explained based on a limited number of transfer rules and lexico-semantic properties of constituent lexical items.

To capture the regularity underlying such paraphrases, we propose a lexical-semantics-based account, which is based on the Lexical Conceptual Structure (LCS) (Jackendoff, 1990; Kageyama, 1996), and represents verbs as semantic structures providing the relationships between their arguments and syntactic cases. We explore the sorts of lexico-semantic properties that can be explained by LCS.

1.6 Thesis outline

This thesis is organized as follows. Chapter 2 discusses the target of this thesis, namely, a variety of lexical and structural paraphrases in Japanese. We collect paraphrases discussed on either analyses from the linguistic point of view or previous research from the viewpoint of natural language processing. In the classification, each class of paraphrases is characterized by their syntactic patterns and lexico-semantic information required for carrying them out. The next two chapters are dedicated to an investigation into transfer errors. Chapter 3 details our investigation into transfer errors which explores what types of errors occur and the tendencies and characteristics of each type of error. This work aims to justify our transfer-and-revise approach. Chapter 4 focuses on one of the most crucial transfer errors observed in Chapter 3: incorrect case assignments. Problems in language modeling for error detection are described and our empirical solution for them is presented. Chapter 5 proposes a lexical-semantics-based account of a subset of paraphrases. After displaying the basics of an existing typology of LCS and our refinement, we describe a semantic transformation procedure performed over the LCS. The shortcomings of the typology and the limitations of the theory are also pointed out. Finally, Chapter 6 summarizes the contributions in this thesis and suggests future directions.

CHAPTER 2 Classification of lexical and structural paraphrases

2.1 Introduction

By the term **lexical and structural paraphrases**, we mean meaning-preserving linguistic transformations that can be carried out without referring to the communicative situation, such as where the sentence is uttered. To have an intuition about what sorts of paraphrases we humans manipulate, we collected a variety of linguistic phenomena in Japanese. In the following discussion, each paraphrase example was extracted from either an analysis from the linguistic point of view or previous research in NLP. Then, we classified them according to their similarities and differences in syntactic characteristics[1].

2.2 Classification

2.2.1 Paraphrases of single content words

The most primitive and atomic classes of paraphrases are those performed by replacing words with their synonymous expressions including synonyms and phrases conveying the same denotation. Knowledge for synonymous expressions can be directly extracted from existing language resources such as thesauri and ordinary dictionaries. Several attempts have also been made to extract such knowledge from corpora (Barzilay and McKeown, 2001; Lin and Pantel, 2001; Shimohata and Sumita, 2002a; Shimohata and Sumita, 2002b; Torisawa, 2002; Pang *et al.*, 2003; Shinyama and Sekine, 2003; Ohtake and Yamamoto, 2003; Quirk *et al.*, 2004).

The following shows examples of this level of paraphrases.

(16) **Paraphrasing of common nouns to their synonyms** (Fujita and Inui, 2001; Yamamoto, 2002b; Okamoto *et al.*, 2003)

s. ***kyuryo**-ni* *kinenkan-ga* *kansei-shita.*
hill-LOC a memorial hall-NOM to build up-PAST
A memorial hall was completed on the hill.

t. ***takadai**-ni* *kinenkan-ga* *kansei-shita.*
hill-LOC a memorial hall-NOM to build up-PAST

(17) **Paraphrasing of common nouns to their definition statements** (Fujita *et al.*, 2000)

[1] This chapter refers only to analyses in Japanese. The latest version, which also includes English and other examples, is available at http://cl.naist.jp/kura/ParaCorpus/.

s. *gareki-ya* ***haizai**-no* *kari-okiba.*
rubble-and scrap woods-GEN temporary space
Temporary space for rubble and scrap woods.

t. *gareki-ya* [***iranaku*** ***nat-ta***] ***mokuzai**-no* *kari-okiba.*
rubble-and be unnecessary become-PAST woods-GEN temporary space
Temporary space for rubble and woods that became unnecessary.

(18) **Paraphrasing of verbs to their synonyms** (Kondo and Okumura, 1997; Kondo *et al.*, 1999; Torisawa, 2002)

s. *kare-wa* *team play-**ni*** ***tessuru**.*
he-TOP team play-DAT to devote-PRES
He devotes himself to team play.

t. *kare-wa* *team play-**o*** ***tsuranuku**.*
he-TOP team play-ACC to devote-PRES

(19) **Paraphrasing of verbs to their definition statements** (Kaji *et al.*, 2001; Kaji *et al.*, 2002)

s. *taxi-ni* ***ainori-suru**.*
taxi-DAT share-PRES
We share a taxi.

t. *taxi-ni* ***issho-ni*** ***noru**.*
taxi-DAT together-DAT ride-PRES
We ride a taxi together.

2.2.2 Function-expressional paraphrases

Functional expressions include, for example, function word sequences and modality expressions. Since some such expressions have the same meaning, they can be paraphrased into each other. The issue is to discover the groups of paraphrasable expressions, and to build a dictionary which describes their similarities and differences in meanings and usages.

(20) **Paraphrasing of function word sequences** (Morita and Matsuki, 1989; Iida *et al.*, 2001; Kurokawa, 2003; Matsuyoshi *et al.*, 2004; Tsuchiya *et al.*, 2004)

s. *jisui-wa* [*keizaiteki-dearu*] ***no-wa*** ***motoyori***
home-cooking-TOP economical-COP nominalizer-TOP of course
kenkou-ni-mo *yoi.*
health-DAT-too be good-ADJ
Home cooking is good for our health, and economical too.

t. *jisui-wa* [*keizaiteki-dearu*] ***nominarazu***
home-cooking-TOP economical-COP not only . . . but also
kenkou-ni-mo *yoi.*
health-DAT-too be good-ADJ
Home cooking is not only economical but also good for our health.

(21) **Moving emphatic particles** (Kinsui *et al*., 2000; Tokunaga, 2002)

s.	*kanojo-wa*	*karai*	***okazu-o***	***tabete-bakari-i**-ta.*
	she-TOP	be hot-ADJ	dishes-ACC	to eat-only-PAST
	She used to only eat hot dishes.			
t.	*kanojo-wa*	*karai*	***okazu-bakari-o***	***tabe-tei**-ta.*
	she-TOP	be hot-ADJ	dishes-only-ACC	to eat-PROG-PAST
	She used to eat only hot dishes.			

(22) **Paraphrasing of polite expressions** (Kabaya *et al*., 1998; Ohtake and Yamamoto, 2001; Shirado *et al*., 2003)

s.	*junbi-ga*	*dekiru-made*	***shosho***	***o-machi**-kudasai.*
	preparation-NOM	to finish-until	a moment-ADV	HON-to wait-please
	Could you wait for a moment until it is ready.			
t.	*junbi-ga*	*dekiru-made*	***chotto***	***matte**-kudasai.*
	preparation-NOM	to finish-until	a bit-ADV	to wait-please
	Please wait for a bit until it is ready.			

2.2.3 Paraphrases of compound expressions

When a number of content words form a compound word, the relationship between them is implicitly conveyed, such as conjunctions and case markers. An algorithm which is capable of paraphrasing compound words into expressions with such relationships, would contribute to, for example, reading assistance, information retrieval, and answer verification in question answering. On the other hand, an algorithm which generates compound words from a couple of words contributes to summarization, disambiguation on parsing, and so on.

(23) **Composition or decomposition of compound nouns** (Sato, 1999; Takeuchi *et al*., 2002; Ohashi and Yamamoto, 2004)

s.	*kare-wa*	*kikai-**sousa-ga***	***jouzu-da**.*	
	he-TOP	machine-operation-NOM	be good-COP	
	He is good at operating the machine.			
t.	*kare-wa*	*kikai-**o***	***jouzu-ni***	***sousa-suru**.*
	he-TOP	machine-DAT	well-ADV	to operate-PRES
	He operates the machine well.			

(24) **Composition or decomposition of compound verbs** (Himeno, 1999; Uchiyama and Ishizaki, 2003)

s.	*kare-wa*	*yuhan-o*	***tabe-sugi**-ta.*	
	he-TOP	dinner-ACC	to overeat-PAST	
	He ate too much at dinner.			
t.	*kare-wa*	*yuhan-o*	***hitsuyou-ijo-ni***	***tabe**-ta.*
	he-TOP	dinner-ACC	than necessary-DAT	to eat-PAST
	He ate meal than necessary at dinner.			

(25) ***"N1 no* (of) *N2"* ⇔ noun phrase with relative clause** (Kurohashi and Sakai, 1999; Kataoka *et al*., 2000; Torisawa, 2001)

s.

kare-wa	*senkyo-e-no*	***shutsuba-no***	***aisatsu**-o*	*okonat-ta.*
he-TOP	election-to-GEN	candidacy-GEN	greeting-ACC	to do-PAST

He did the greeting of candidacy for the election.

t.

kare-wa	**[** *senkyo-e-no*	***shutsuba-o***	***hyomei-suru*]**	***aisatsu**-o*	*okonat-ta.*
he-TOP	election-to-GEN	candidacy-ACC	to declare-PRES	greeting-ACC	to do-PAST

He did the greeting that declares candidacy for the election.

2.2.4 Clause-structural paraphrases

These classes of paraphrases involve syntactic transformations within clauses. Since the variation of transformation patterns are not so large, it is possible to manually describe an exhaustive list of patterns. The applicability conditions of each pattern, however, depend on the lexical items involved in the transformation. To specify such conditions, we need to explore and develop a lexical knowledge resource. In addition, these paraphrases affect discourse elements: e.g., example (27) switches the topic, while example (30) changes the focus of the action by switching case markers. Depending on the purpose of paraphrasing, their effects on the context need to be taken into account. In this thesis, we regard paraphrases realized by these classes of transformations to be correct if denotational meanings of them are equivalent.

(26) **Inner-clausal negative-affirmative paraphrasing** (Hayashi, 1992; Kondo *et al*., 2001; Iida *et al*., 2001; Tokunaga, 2002)

s.

*kare-**wa***	*ringo-**shika***	*tabe-**taku-nakat**-ta.*
he-TOP	apple-except	to eat-to want-NEG-PAST

He wanted to eat nothing but apples.

t.

*kare-**ga***	*tabe-**takat-ta-no-wa***	*ringo-**dake-dat**-ta.*
he-NOM	to eat-to want-PAST-thing-TOP	apple-only-COP-PAST

All he wanted to eat was an apple.

(27) **Paraphrasing of comparative expressions** (Kondo *et al*., 2001; Tokunaga, 2002)

s.

*tonari-machi-**wa***	*waga-machi-**yori***	*sanrin-shigen-ga*	***toboshii**.*
neighboring town-TOP	our town-than	forest resources-NOM	be poor-ADJ

The neighboring town is poorer in forest resources than our town.

t.

*waga-machi-**wa***	*tonari-machi-**yori***	*sanrin-shigen-ga*	***yutaka-da**.*
our town-TOP	neighboring town-than	forest resources-NOM	be rich-COP

Our town is richer in forest resources than the neighboring town.

(28) **Voice alternation** (Inoue, 1976; Yamaguchi *et al*., 1998; Kondo *et al*., 2001)

s.

kotoshi-wa	*toshi-kiban-no*	*seibi-**ga***	***okonaw-are**-ta.*
this year-TOP	city infrastructure-GEN	maintenance-NOM	to do-PASS-PAST

City infrastructure was improved this year.

t. *kotoshi-wa* *toshi-kiban-no* *seibi-**o*** ***okonat**-ta.*
this year-TOP city infrastructure-GEN maintenance-ACC to do-PAST
We improved city infrastructure this year.

(29) **Transitivity alternation** (Kageyama, 1996; Kageyama, 2001; Kondo *et al*., 2001; Ito, 2002)

s. *sentakumono-**ga*** *soyokaze-**ni*** ***yureru**.*
laundry-NOM breeze-DAT to sway (intransitive)-PRES
The laundry sways in the breeze.

t. *soyokaze-**ga*** *sentakumono-**o*** ***yurasu**.*
breeze-NOM laundry-ACC to sway (transitive)-PRES
The breeze makes the laundry sways.

(30) **Locative alternation** (Kishimoto, 2001; Ogawa and Ishizaki, 2004)

s. *kacho-ga* *sakazukin-**ni*** *nihonshu-**o*** *mitashi-ta.*
section chief-NOM cup-DAT Japanese sake-ACC to fill-PAST
The section chief filled Japanese sake into the cup.

t. *kacho-ga* *sakazukin-**o*** *nihonshu-**de*** *mitashi-ta.*
section chief-NOM cup-ACC Japanese sake-IMP to fill-PAST
The section chief filled the cup with Japanese sake.

(31) **Donative alternation** (Kageyama, 2002)

s. *sensei-**ga*** *seito-**ni*** *mondai-no* *tokikata-o* ***oshieru**.*
teacher-NOM student-DAT problem-GEN solution-ACC to teach-PRES
The teacher teaches the student how to solve the problem.

t. *seito-**ga*** *sensei-**ni*** *mondai-no* *tokikata-o* ***osowaru**.*
student-NOM teacher-DAT problem-GEN solution-ACC to learn-PRES
The student learns how to solve the problem from the teacher.

(32) **Paraphrasing of light-verb constructions** (Oku, 1990; Muraki, 1991; Kaji and Kurohashi, 2004)

s. *jumin-**no*** *nesshin-**na*** ***yousei-o*** ***uke**,* **(adverbial clause)**
residents-GEN be eager-ADJ request-ACC to receive (PAST)
kouji-o *chushi-shi-ta.* **(matrix clause)**
construction-ACC to stop-PAST
Construction was stopped in response to residents' eager request.

t. *jumin-**ni*** *nesshin-**ni*** ***yousei-s-are**,* **(adverbial clause)**
residents-DAT eagerly-ADV to request-PASS (PAST)
kouji-o *chushi-shi-ta.* **(matrix clause)**
construction-ACC to stop-PAST
Construction was stopped because residents eagerly requested it.

2.2.5 Multi-clausal paraphrases

The following classes of paraphrases straddle two or more clauses. In examples (33) and (34), the themes of the sentences are changed by switching syntactic heads or splitting complex sentences. These motivate some accessory operations such as pronoun generation. In example (35), on the other hand, a conjunction which connects two clauses (e.g., "*dakara* (thus)") is required to preserve the rhetorical structure of the original compound sentence. As the following examples demonstrate, these classes of paraphrases affect discourse-related correctness, so-called cohesiveness.

(33) **Separating relative clause from a given sentence / sentence aggregation** (Inui and Nogami, 2001; Ori and Sato, 2002)

s. **[** *Stockholm-no* *nansei-ni* *ichi-suru* **]** **(relative clause)**
Stockholm-GEN south-west-DAT to be located
***Småland-chiho**-wa* *"garasu-no oukoku"-to* *yob-are-teiru.* **(matrix clause)**
Småland region-TOP "Kingdom of Glass"-as to call-PASS-PROG
Småland, which is located on the south-west of Stockholm, is called "The Kingdom of Glass."

t. *Småland-chiho-wa* *Stockholm-no* *nansei-ni* *ichi-suru.* **(satellite)**
Småland region-TOP Stockholm-GEN south-west-DAT to be located
***kono-chiho**-wa* *"garasu-no oukoku"-to* *yob-are-teiru.* **(nucleus)**
this region-TOP "Kingdom of Glass"-as to be called-PROG
Småland is located on the south-west of Stockholm. The region is called "The Kingdom of Glass."

(34) **Removal of cleft constructions** (Sunagawa, 1995)

s. [*konshu* *tosen-shi-ta*] ***no-wa*** *Nara-ken-no* *dansei-**dat-ta**.*
this week-ADV to win-PAST thing-TOP Nara prefecture-GEN man-COP-PAST
The person who won the game this week was a man in Nara.

t. *konshu-**wa*** *Nara-ken-no* *dansei-**ga*** *tosen-shi-ta.*
this week-TOP Nara prefecture-GEN man-NOM to win-PAST
A man in Nara won the game this week.

(35) **Separating adverbial clause from a given sentence** (Takeishi and Hayashi, 1992; Kouda *et al*., 2001; Mitamura and Nyberg, 2001)

s. *kinou-wa* *nagai-aida* *hare-tei-ta-**tame**,* **(adverbial clause)**
yesterday-TOP for long time-ADV be sunny-PROG-PAST-because
sentakumono-ga *yoku* *kawai-ta.* **(matrix clause)**
laundry-NOM well-ADV to dry-PAST
Since it was sunny yesterday, the laundry dried well.

t. *kinou-wa* *nagai-aida* *hare-tei-ta.* **(satellite)**
yesterday-TOP for long time-ADV be sunny-PROG-PAST
dakara *sentakumono-ga* *yoku* *kawai-ta.* **(nucleus)**
thus laundry-NOM well-ADV to dry-PAST
It was sunny yesterday. Thus, the laundry dried well.

(36) **Inter-clausal negative-affirmative paraphrasing** (Hayashi, 1992; Tokunaga, 2002)

s.

*kakunin-**sur-eba***	*moushikomi-wa*	*torikes-**are-nai**.*
to confirm-if	application-TOP	to cancel-PASS-NEG

The application will not be canceled if you confirm it.

t.

*kakunin-**shi-nai-to***	*moushikomi-wa*	*torikes-**areru**.*
to confirm-NEG-if	application-TOP	to cancel-PASS

The application will be canceled if you do not confirm it.

2.2.6 Paraphrases of idiosyncratic expressions

To carry out the following paraphrases, we need to collect the exact pairs of expressions, since they cannot be generalized at all.

(37) **Paraphrasing of idiomatic expressions** (Oku, 1990; Mitamura and Nyberg, 2001)

s.

biotechnology-ga	***kyakkou-o***	***abi**-teiru.*
biotechnology-NOM	footlight-ACC	to be flooded-PROG

Biotechnology is in the limelight.

t.

biotechnology-ga	***chumoku-s-are**-teiru.*
biotechnology-NOM	to attract ones attention-PASS-PROG

(38) **Altering notational variants, abbreviations, and acronyms** (Terada and Tokunaga, 2001; Sakai and Masuyama, 2003)

s.

ooku-no	*shimin-ga*	***genpatsu**-no*	*kensetsu-ni*	*hantai-shi-teiru.*
many-QUAN	citizen-NOM	nuclear plant-GEN	construction-DAT	to oppose-PROG

Many people oppose building a nuclear plant.

t.

ooku-no	*shimin-ga*	***genshiryoku-hatsudensho**-no*
many-QUAN	citizen-NOM	nuclear plant-GEN

kensetsu-ni	*hantai-shi-teiru.*
construction-DAT	to oppose-PROG

(39) **Analysis of metonymy and synecdoche**

s.

Shakespeare-o	*yomu.*
Shakespeare-ACC	to read-PRES

I read a Shakespeare.

t.

[*Shakespeare-**ga***	***kai-ta***]	***hon**-o*	*yomu.*
Shakespeare-NOM	to write-PAST	book-ACC	to read-PRES

I read a book written by Shakespeare.

2.3 Related work

Levin (1993) has examined verbs on several types of possible alternations aiming to classify them. Although her classification provides an extensive list of verb classes, it does not contain other paraphrase phenomena. In addition, it does not define the underlying meaning components of each class. Nevertheless, approaches of recent lexical semantics in English, such as those in (Dorr, 1997), could utilize Levin (1993)'s classification for various NLP tasks, by introducing special definitions of various types of meaning components. Dras (1999), on the other hand, created several classes of paraphrases from the grammatical viewpoint. Since his aim was to represent paraphrases via the formalism of a synchronous tree adjoining grammar (STAG), structural transformation is involved in most examples for labeling each type of paraphrases. Some recent reports (Kozlowski *et al*., 2003; Rinaldi *et al*., 2003; Dorr *et al*., 2004) enumerated paraphrases to illustrate their concerns. For example, (Dorr *et al*., 2004) listed types of relationships underlying paraphrases in order to argue for the applicability of their interlingual representation which was designed based on the Lexical Conceptual Structure. A comparison between such studies and our classification suggests that the variations in Japanese and English have much phenomena in common, although there exist some idiosyncratic phenomena in each language. For example, *dative alternation* shown in example (41) is an English phenomenon, which cannot be expressed in Japanese. Example (40), on the other hand, shows a *de-causativization* (Kageyama, 1996) which appears in Japanese but not in English:

(40) s. ***chichi-ga*** *niwa-ni* *atarashii* ***ki-o*** ***ue-ta***.
father-NOM garden-DAT new-ADJ tree-ACC to plant-PAST.
My father planted a new tree in the garden.

t. *niwa-ni* *atarashii* ***ki-ga*** ***uwat-ta***.
garden-DAT new-ADJ tree-NOM to plant-PAST.
*A new tree planted in the garden.

(41) s. John sold **a Honda** to **Tom**.

t. John sold **Tom a Honda**.

Our classification looks similar to those in (Ohtake and Yamamoto, 2003; Shimohata, 2004). However, ours does not limit the domain of texts or sentences as in (Ohtake and Yamamoto, 2003), and ours provides more detailed classification than in (Shimohata, 2004). Yamamoto (2001) enumerated several factors which motivates paraphrasing, such as style, manner, and circumstantiality. However, at the moment, we do not take such a factor into account in the classification of paraphrases.

2.4 Summary

In this chapter, we proposed a classification of lexical and structural paraphrases in Japanese. By building such a classification, we expect the following benefits:

- It gives a chance to discover knowledge which can be commonly used for explaining similar classes of paraphrases.

- It enables us to construct a paraphrase corpus which is specialized to a particular class of paraphrases. We are currently addressing the issue of paraphrase corpus construction relying on the classification proposed.

- If we can establish a method to carry out a particular class of paraphrases, this may be used for decomposing automatically acquired paraphrases. Taking (42) as an example, one can naively obtain the pattern: "*N1* is purchased by *N2*" ⇒ "*N2* buys *N1*." By using a well-defined syntactic transformation pattern, such as "*N1* be *V*-PERF by *N2*" ⇒ "*N2 V N1*" (voice activization), we would be able to decompose the above complex paraphrase, and obtain a lexical paraphrase ("purchase" ⇒ "buy"). Such a decomposition contributes to reduce the cost of manual correction.

(42) s. This car **was purchased by** him.

t. He **bought** this car.

The remaining issues are as follows:

Definition of each paraphrase class: Each paraphrase class is merely characterized by a prototypical example, but not concretely defined. Although our bottom-up classification is useful to overview the variety and similarities of paraphrases, we need to give a definition to each class for computational treatment.

Completeness over a broad range of classes: There are undiscovered paraphrase phenomena, namely, a newly collected paraphrase example may fall into the other class we labeled. To build an exhaustive collection, we need to continue collecting paraphrase examples. In particular, we need to develop a methodology which exhumes new or non-stereotypical paraphrase examples from automatically acquired and manually generated ones.

In the following chapters, we will use the names the of paraphrase classes in this chapter to refer back to the typical examples.

CHAPTER 3 *Transfer-and-revise approach to paraphrase generation*

3.1 Introduction

Let us consider an example:

(43) s.

ringo	***shika***	*tabe*	***nai****.*
apple	except	to eat	NEG

I eat nothing but apples.

t.

taberu	***no***	***wa***	*ringo*	***dake***	***da****.*
to eat	thing	TOP	apple	only	COP

All I eat is apples.

Generalizing from this example, one can come up with a syntactic transfer rule such as the following:

(44)

N	*shika*	*V*	*nai*	$\Rightarrow$	*V*	*no*	*wa*	*N*	*dake*	*da*
NOUN	except	VERB	NEG		VERB	thing	TOP	NOUN	only	COP

This rule, however, does not completely specify how to paraphrase such sentences as (43). For example, it does not take into account the conjugation form of the verbs. The form of an input verb has to be changed depending on the context; in example (43), "*tabe*" needs to be transformed into "*taberu*."

The next example illustrates the situation can become even more complicated:

(45) s.

kare-wa	***ringo-shika***	*tabe-**taku-nakat**-ta.*
he-TOP	apple-except	to eat-to want-NEG-PAST

He wanted to eat nothing but apples.

t.

**kare-wa*	*tabe-**taku-no-wa***	***ringo-dake-da****-ta.*
he-TOP	to eat-to want-thingTOP	apple-only-COP-PAST

r.

kare-ga	*tabe-takat-ta-no-wa*	*ringo-dake-dat-ta.*
he-NOM	to eat-to want-PAST-thing-TOP	apple-only-COP-PAST

All he wanted to eat was an apple.

A naive application of rule (44) to sentence (45s) produces (45t), which has a topicalization error ("*wa*"), a tense-related error ("*taku*"), and two conjugation errors ("*taku*" and "*da*") — the desirable output would be (45r). We henceforth call such errors **transfer errors**.

How do we avoid transfer errors? One simple approach is to annotate additional conditions into the transfer rules: for example, "if the input has a topic, its case must be changed."

However, incorporating all such factors into a transfer rule would invariably make it very complicated. Moreover, it is impossible to condition a large number of paraphrasing rules taking into account all sorts of contextual factors. Hence, transfer rules tend to have insufficient applicability conditions, and thus they are likely to produce morpho-syntactically ill-formed or semantically inappropriate expressions.

Our approach to this problem is to (i) leave the description of each transfer rule **underspecified** as in example (44) and (ii) introduce a revision component that utilizes the knowledge about linguistic constraints to detect, revise, and reject faulty results occurring in transfer, such as those above. We call the description of linguistic constraints a **language model**.

As described in Chapter 2, various case studies have so far been done on lexical and structural paraphrases in Japanese. And some of them described several typical errors and linguistic constraints employed for avoiding errors. With respect to these studies, there should be a wide range of linguistic constraints that are independent of a particular transfer rule and thus should be implemented separately from the transfer knowledge. Typical examples are constraints on morpheme connectivity, verb conjugation, word collocation, and tense and aspect forms in relative clauses. There has not, however, been a comprehensive investigation into transfer errors and required linguistic constraints across different paraphrase classes, because the work has tended to focus only on particular single paraphrase class.

In this chapter, we empirically justify our transfer-and-revise approach to paraphrase generation by making an extensive investigation into transfer errors. The following sections support this argument. Section 3.2 illustrates that the knowledge proposed in previous case studies contains several sorts of linguistic constraints which can be considered to be transfer-independent. Section 3.3 describes a transfer error typology which is built referring to knowledge provided in previous studies on lexical paraphrasing. Then, Section 3.4 explores how transfer errors occur across different paraphrase classes. To show the tendencies of transfer errors, we newly develop and acquire paraphrasing rules without limiting our target to lexical paraphrasing, and use this together with those built in Section 3.3. Finally, Section 3.5 summarizes the findings from the error distributions and designates the errors which should be given priority to be solved.

3.2 Knowledge in previous case studies

What types of transfer errors occur in lexical and structural paraphrasing? Analyses of previously proposed knowledge may give us an insight into the transfer errors and linguistic constraints. In this section, we therefore review transfer knowledge described in some case studies and draw our blueprint for separating transfer knowledge and linguistic constraints.

Recall that we distinguish transfer and revision processes as follows:

Transfer: Apply transfer rules to an input even if they have underspecified applicability conditions, and generate a set of **paraphrase candidates**.

Revision: Detect transfer errors in each paraphrase candidate, and if possible resolve them utilizing the knowledge base of linguistic constraints; otherwise, reject the candidate.

Note that we will never say it is wrong to employ linguistic knowledge as a constraint in the transfer stage. Although we decompose knowledge as much as possible, our goal is a comprehensive exploration of how errors occur in order to see the revision processes that are required.

Sato (1999)

Sato (1999) addressed the task of compound noun decomposition. His algorithm handles technical papers' titles, which tend to be compound nouns, with a regular expression and produces their paraphrases by transforming their structures as example (46) illustrates:

(46) s. *kasetsu-sentei-kikou*
hypotheses-selecting-mechanism

t. *kasetsu-**o*** ***sentei-suru*** *kikou*
hypotheses-ACC to select mechanism
Mechanisms which selects hypotheses

The captured expression is then transformed by using a sequence of primitive transformation procedures which involves miscellaneous changes, such as those on dependencies, conjugation forms, and case assignments. Since these changes seem to be independent of transformation (i.e., capturing elements and reordering them), they can be regarded as candidates for revision processes.

Kondo *et al.* (1999)

The distinction between transfer and revision processes is more salient in (Kondo *et al*., 1999). To paraphrase Sino-Japanese verbal nouns[1] into ordinary Japanese verbs such as example (47), they proposed a number of transformation procedures such as shown in example (48):

(47) s. *hannin-wa* *keisatsu-ni* ***taiho-s-are**-ta.*
criminal-TOP police-DAT to arrest-PASS-PAST
The criminal was arrested by police.

t. *hannin-wa* *keisatsu-ni* ***tsukamae-rare**-ta.*
criminal-TOP police-DAT to capture-PASS-PAST

(48) s. *SV* *s* *areru* *
SINO-JAPANESE VERBAL NOUN / to do / PASS / VERBAL SUFFIXES

Step 1. Replace the expression above with an ordinary Japanese verb phrase which corresponds to the Sino-Japanese verbal noun ("*taiho-s-are-ta*" ⇒ "*tsukamaeru*").

Step 2. Modify the conjugation form of the head of verb phrase according to those of the given "*s* (to do)" ("*tsukamaeru*" ⇒ "*tsukamae*").

[1] These nouns are called "*sahen*-nouns" as they behave like "*sahen*-verbs" when followed by "*suru*."

Step 3. Append a verbal suffix "*reru* (PASS)" if the head of verb phrase is either a kind of Japanese *sahen*-verb (e.g. "*suru* (to do)" and "*tassuru* (to achieve)") or a consonant verb; otherwise, "*rareru* (PASS)" ("*tsukamae*" ⇒ "*tsukamae-rareru*").

Step 4. Modify the conjugation form of the appended verbal suffix according to those of the given "*reru*" or "*rareru*" ("*tsukamae-rareru*" ⇒ "*tsukamae-rare*").

Step 5. Append the given verbal suffixes to the newly placed verb phrase ("*tsukamae-rare*" ⇒ "*tsukamae-rare-ta*").

This type of paraphrases corresponds to a subclass of paraphrases of single content words described in Chapter 2. Step 1 is a transfer process and is followed by the post-transfer processes in steps 2 to 5. We can regard the latter steps as the revision processes, because these processes are created not only for producing paraphrases but for avoiding transfer errors.

Kondo *et al.* (2001)

Kondo *et al*. (2001) presented 43 transfer rules which handle three paraphrase classes: (a) voice alternation, (b) transitivity alternation, and (c) paraphrase of comparative expressions (see examples (27) to (29) in Chapter 2 for the corresponding examples). The following transfer rule is one of those developed for voice alternation.

(49)	*N1-ga*	*N2-o*	*V-suru*	⇒	*N2-ga*	*N1-ni*	*V-sa-reru*
	NOUN-NOM	NOUN-ACC	VERB-ACT		NOUN-NOM	NOUN-DAT	VERB-PASS

(where noun *N2* must be animate, verb *V* must allow the passive form, and *V* requires an noun phrase with the accusative case "*o*," while it does not require the an noun phrase with dative case "*ni*.")

Some constraints are involved in the rules, but they seem to be independent of the transfer. However, we regard them as linguistic constraints: for example, they can be detached from the transfer pattern and implemented in the form of a set of revision patterns such as those described in (50), where a symbol "→" is introduced to represent revision patterns, which differs from "⇒" for transfer patterns.

(50)

a. *hito-**kara*** *tsukaw-areru* → *hito-**ni*** *tsukaw-areru*
human-ABL to be used → human-DAT to be used

b. *N-ga* *V-sa-reru* → *<reject>*
INANIMATE NOUN-NOM VERB-PASS

c. *a-wareru* → *<reject>*
to be meeten

3.3 Building a transfer error typology

In this section, we reexamine the knowledge provided in several previous studies on lexical paraphrasing, and enumerate transfer errors and their revision patterns occurring in paraphrase generation.

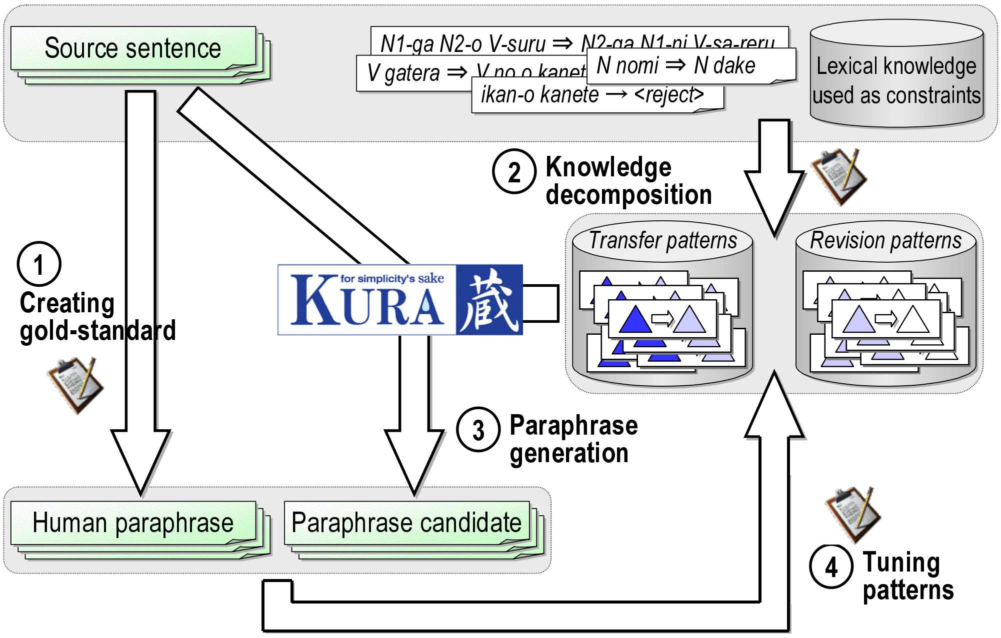

Figure 3.1. Knowledge decomposition scheme for collecting revision examples.

3.3.1 Method

Our method for building a transfer error typology consists of two processes, namely, collecting revision examples and distilling them into error patterns.

Collecting revision examples

Paraphrased sentences can involve a plural number of primitive transfer errors, such as that we at first tentatively called tense-related errors and conjugation errors during the explanation of (45t). In this way, transfer errors exhibit a wide range of types from morphological to semantic and discourse-related ones. However, the sorts of error that occur and how they are defined have not been investigated yet, as there was no generally available resource of transfer errors. We therefore collected erroneous paraphrase examples and their revision examples as follows (c.f., Figure 3.1):

Step 1. For an initial set of sentences, manually develop their paraphrases, assuming particular paraphrase classes. The resultant pairs of source and paraphrased sentences can be regarded as a gold-standard.

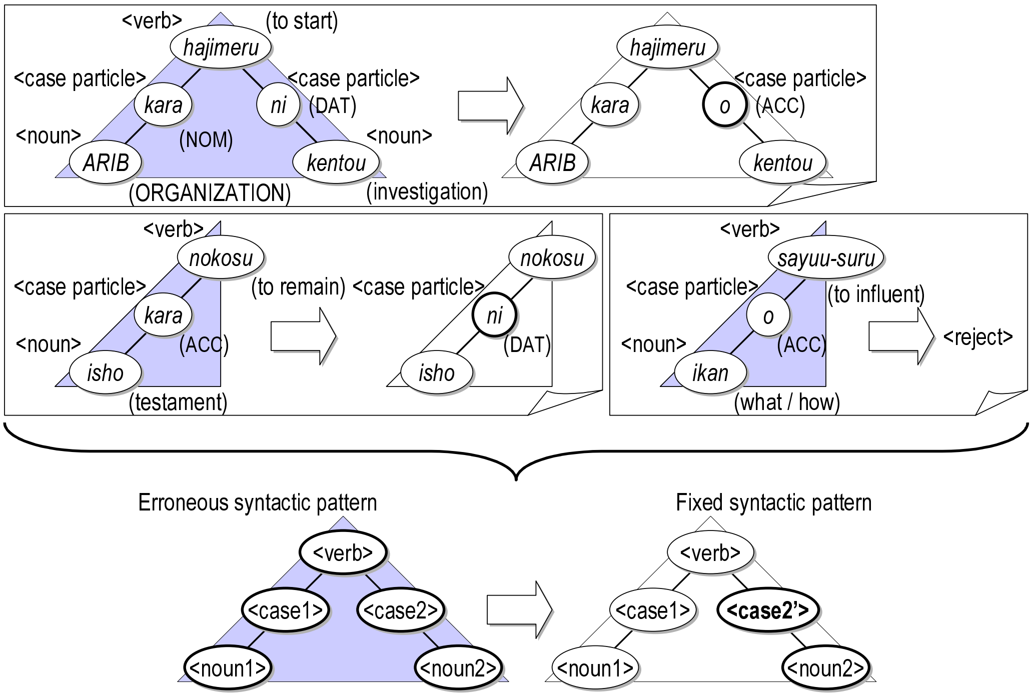

Figure 3.2. Distillation of revision examples into patterns.

Step 2. Decompose the transfer knowledge available from previous case studies into transfer knowledge and transfer-independent linguistic constraints[2], and then implement them in the paraphrase generation system KURA as transfer and revision rules, respectively.

Step 3. Feed KURA with the source sentences obtained in step 1 and the newly tailored patterns in step 2, and generate paraphrase candidates. Some of them are erroneous, and are sometimes revised by KURA using already implemented revision rules.

Step 4. Refine the transfer and revision rules comparing the paraphrase candidates with the gold-standards.

Step 5. Repeat steps 3 and 4 until the rules can generate the entire set of the gold-standard without any error. The final set of revision rules are the examples of error correction.

Distillation of revision examples into patterns

While collecting revision examples, we allow the use of lexical information for describing applicability conditions of revision rules, since we need to generate paraphrases accurately. To

[2]Note that we will never say it is wrong to employ linguistic knowledge as constraint in the transfer stage. In this section, we decompose knowledge as much as possible only for exploring errors and revision processes for them.

generalize revision patterns, we then discard such information from collected revision examples. As Figure 3.2 illustrates, generalization is conducted at the morpho-syntactic level.

In particular, we employ a morpheme-based dependency structure (MDS), which is the internal representation of KURA. Each tree in Figure 3.2 represents an MDS. MDS is a dependency tree each of whose nodes corresponds to a morpheme (lexical entry) with various types of morpho-syntactic information such as part-of-speech tags and conjugation types and forms annotated by the morphological analyzer ChaSen (version 2.2.9) and the morphological dictionary IPADIC (version 2.4.4)[3].

With this decomposition, the following benefits can be expected:

- Each piece of revision component can be utilized for compensating the broad range of transfer rules for errors, even if the transfer rules have been acquired automatically and may cause errors.
- Linguistic knowledge required for developing each piece of revision component should be simple. Thus, development of such a knowledge base is feasible.

3.3.2 Examination

Although a number of studies have shown examples of linguistically motivated post-processes, we empirically reexamine only a few of the studies for the following reasons. First, we prefer studies which freely provide either traceable knowledge acquisition procedures or resultant knowledge bases. Second, in order to build a typology of transfer errors, we need to implement knowledge in our paraphrase generation system, KURA. Therefore, we prefer knowledge bases which suit the environment, in particular, we prefer morpho-syntactic paraphrase rules and morpheme-related lexical knowledge. Among a number of case studies that satisfy the above conditions, we take into consideration a couple of studies tailored for the following paraphrase classes:

SVs: Paraphrasing “Sino-Japanese verbal noun + verbal suffix” expressions into paraphrastic verb phrases such as example (47). Kondo *et al.* (1999) presented 5 transformation procedures containing revision processes such as those we have seen in example (48). To perform Step 1 in (48), we extracted pairs of a Sino-Japanese verbal noun and its corresponding paraphrastic verb phrase from definition statements in dictionaries (Ôno and Hamanishi, 1981; RWC, 1998), the given expression is then paraphrased into verb phrases within its definition statements. We employed 6,642 pairs, a part of which was revised using the 23 paraphrase examples examined in (Kondo *et al.*, 1999).

Cas: Based on (Kondo *et al.*, 2001), we implemented 33 transfer rules and revised using the 25 single clause sentences (and their 38 paraphrases) examined in their article. Since the original rules contained constraints independent of the transfer, we detached these constraints and implemented them as revision rules such as those shown in example (50). We used more than 200 pairs of transitive and intransitive verbs provided by them.

[3]ChaSen and IPADIC are available at http://chasen.naist.jp/.

Neg: Negative-affirmative transformations were discussed in (Tokunaga, 2002). He provided 291 transfer rules and 30 revision rules that were revised using 500 negative sentences extracted from newspaper articles. Example (44) shows a transfer rule which is included in this paraphrase class.

Aux: Iida *et al*. (2001) studied alterations of auxiliary verbs and functional expressions. They provided 261 transfer and 42 revision rules that were revised using the 185 sentences (and their 234 paraphrases) collected from three teaching sources for Japanese as a second language. Since the transfer rules for this paraphrase class tended to contain one kind of linguistic constraint, we discarded the constraints and rebuilt 248 transfer rules. Samples of the transfer and revision rules are shown in examples (51) and (52), respectively.

(51) a.

V	*gatera*	⇒	*V*	*no*	*o*	*kanete*
VERB	while		VERB	thing	ACC	be partly for

b.

N	*nomi*	⇒	*N*	*dake*
NOUN	only		NOUN	only

c.

N-ni	*kakawaru*	⇒	*N-o*	*sayuu-suru*
NOUN-DAT	to affect		NOUN-ACC	to influence

(52) a.

dake	*narazu*	→	*<reject>*
only	not but		

b.

ikan-o	*sayuu-suru*	→	*<reject>*
what / how-ACC	to influence		

As the result, the revision examples were distilled into 13 types of patterns. The transfer errors observed in the above examination exhibited a wide range of variety from morphological errors to semantic and discourse-related ones, namely, it covers morphological, syntactic, semantic, and discourse-related levels. We will exemplify each error in Section 3.4 with discussion on its tendency correlation with the paraphrase classes of transfer rules.

3.4 Error distributions

In the preceding section, we revealed several types of errors occurring in the range of lexical paraphrase classes. In this section, in order to justify our transfer-and-revise approach to paraphrase generation, we explore how transfer errors occur across different paraphrase classes, employing a sort of structural paraphrasing and automatically acquired transfer rules.

3.4.1 Paraphrase generation and manual evaluation

To determine the distributions of errors for a broader range of paraphrase classes, we newly introduced the implementation of the following four classes of transfer rules on KURA[4]:

[4] All information has been refined manually for three years after this investigation.

Table 3.1. Numbers of rules and paraphrase candidates.

Transfer rule class	**Cas**	**Neg**	**Aux**	**SVs**	**Unc**	**VPs**	**CWs**	**Idi**	Total
Corresponding examples in Chapter 2	(27), (28), (29)	(26), (36)	(20)	(18)	(34)	(18)	(16), (18)	(37)	-
# of transfer rules	33	291	248	6,642	18	3,630	13,348	3,942	28,152

Unc: To explore the errors broadly, we newly consider a class of structural paraphrasing. Cleft construction is the structure of clause which emphasizes one of the nominal elements within a clause as shown in sentence (34s) in Chapter 2. Such a structure can be paraphrased into unmarked form as shown in example (34). On the basis of examples examined in (Sunagawa, 1995), we handcrafted 18 transfer rules.

VPs: From a case frame dictionary (Ikehara *et al.*, 1997), we extracted 3,630 pairs of verb phrases. Verb phrases in a pair have the same English translations, thus they seem to be paraphrased into each other. Although paraphrasing of a verb phrase into another in the pair belongs to the same class as the single content word replacement **CWs** below, we consider them separately because the selectional restrictions for their syntactic cases can offer advantages in preciseness.

CWs: Replacing single content words with their synonyms. It is well-known that even synonyms in thesauri are often not interchangeable (Edmonds, 1999; Fujita and Inui, 2001; Lapata, 2001; Pearce, 2001). We therefore extracted highly reliable pairs of synonyms among those derived from the EDR Japanese dictionary (version 1.5) (EDR, 1995). Here, reliability is defined by comparing their definition statements in an ordinary dictionary: a pair of synonyms is reliable if their definition statements share a large portion of the same expressions. We first extracted two sets of synonym pairs utilizing the Kadokawa Synonym New Dictionary (Ôno and Hamanishi, 1981) and Iwanami Kokugo Jiten (RWC, 1998), respectively, and then distilled them into a set of transfer rules[5]. As a result we obtained 29,643 transfer rules for 13,348 words consisting of pairs of nouns, verbs, adjectives, adverbs, and so on. For each source word, we selected the target word at random, and used for error exploration.

Idi: Paraphrases of idiomatic expressions to simpler (literal) expressions. We extracted 3,942 pairs from the Kadokawa Synonym New Dictionary (Ôno and Hamanishi, 1981) using heuristically determined sequential patterns. Since such information includes noise, it will show us the capability of integrating paraphrase systems with automatically acquired transfer rules.

We implemented a total of 28,152 transfer rules[6] on KURA as shown in Table 3.1. We then fed KURA with a set of 1,220 sentences that had been newly excerpted from the Kyoto Univer-

[5]Aiming at the purpose of simplification, which was our present concern, from the all possible source-target pairs, we filtered those consisting of ⟨unfamiliar word, more familiar word⟩ according to the word familiarity scores determined in the Lexical properties of Japanese (Kondo and Amano, 2000).

[6]http://cl.naist.jp/kura/KuraData/.

Table 3.2. Summary of transfer rules and error distributions.

Transfer rule class	**Cas**	**Neg**	**Aux**	**SVs**	**Unc**	**VPs**	**CWs**	**Idi**	Total
# of paraphrase candidates	138	75	19	39	20	60	221	58	630
# of erroneous paraphrase candidates	137	57	9	35	17	53	172	36	516
(a) Incorrect conjugation form	125	41	3	31	7	43	47	6	303
(b) Incorrect functional word connection	42	14	2	3	5		8	4	78
(c) Missing case marker					6	2			8
(d) Collision of two arguments				7				4	11
(e) Incorrect case assignment	66			8		28	57	3	162
(f) Incorrect content words collocation except (e)						3	28	5	36
(g) Change of denotational meaning							30	1	31
(h) Change of meaning of functional expressions	1	5		3			13		22
(i) Discrepancy of tense and aspect	2	1			3				6
(j) Inappropriate style			1						1
(k) Uncoordinated word order	23				2	7		2	34
(l) Inappropriate theme-rheme structure	10	1			10	1			22
(m) Inappropriate rhetorical structure	2	4	2						8
Other errors	38	16	2	7	8	3	19	22	115
(A) Misidentification of idioms and named entity recognition	9			1			26	4	40
(B) Noises of dictionary-specific meta-expressions							18	20	38
(C) Errors on morphological analysis and dependency parsing	7	5	5	1			22	1	41
(D) Errors on transfer	8	1	1		1	1	1	2	15

sity text corpus (Kurohashi and Nagao, 1998), and obtained 630 transferred output sentences. We then manually revised or rejected erroneous sentences, classifying errors within them according to the error classification we created in Section 3.3. The frequency distributions for the observed errors are shown in Table 3.2.

3.4.2 Observations

As shown in Table 3.2, the types of error occurred irrespective of paraphrase classes. This fact suggests that if one creates a revision component specialized for a particular error type, it can be used across different paraphrase classes.

The rest of this section elaborates the characteristics of each type of error in turn.

Morphological errors

The morphological errors, i.e., error types (a) and (b), occurred most frequently. In particular, the most frequent error type involved inappropriate conjugation forms of verbs, i.e., error type (a) (also remember that example (45) contains this error). However, they can be easily resolved, since the conjugation form of a verb can be determined according to the conjugation type of the verb and the part-of-speech of the following word. IPADIC, for example, provides the information of the conjugation type of a large number of verbs, and a connection table. Since the errors are also well-known as a trivial problem of morphological generation, we expect that the conventional rule-based approaches can solve the errors.

The morphological errors also included the third most frequent type of error, namely, incorrect functional word connections, i.e., error type (b) such as the one that example (53t) exhibits. Verbs with non-allowable verbal suffixes are also included in this type of error. These errors can also be resolved by replacing nominal and verbal suffixes into allowable ones whose meaning are the same as the original ones, such as “no” and “na” in example (53).

(53)	s.	***yosougai**-no*	*makekata-ni*	*akke-ni*	*to-rare-ta.*
		unexpected-GEN	defeat-DAT	amazement-DAT	to take-PASS-PAST
		I was taken aback by the unexpected defeat.			
	t.	*****igai**-no*	*makekata-ni*	*akke-ni*	*to-rare-ta.*
		unexpected-GEN	defeat-DAT	amazement-DAT	to take-PASS-PAST
	r.	*igaina*	*makekata-ni*	*akke-ni*	*to-rare-ta.*
		unexpected-ADJ	defeat-DAT	amazement-DAT	to take-PASS-PAST

Syntactic (collocative) errors

Errors in case assignments of verbs, i.e., type (e), formed the second most common error type in Table 3.2. Previous case studies (Kaji *et al*., 2001; Fujita and Inui, 2001; Kondo *et al*., 2001) have also pointed out that this type of error occurred frequently irrespective of the paraphrase classes. For example, “*tessuru*” and “*tsuranuku*” in example (54) have the same translation “to devote;” thus they seem to be interchangeable. However, since the case markers they assign are different, the paraphrased sentence (54t) is incorrect.

(54)	s.	*kare-wa*	*team play-ni*	***tessuru**.*
		he-TOP	team play-DAT	to devote-PRES
		He devotes himself to team play.		
	t.	**kare-wa*	*team play-ni*	***tsuranuku**.*
		he-TOP	team play-DAT	to devote-PRES
	r.	*kare-wa*	*team play-o*	*tsuranuku.*
		he-TOP	team play-ACC	to devote-PRES

One may think that (54t) can be revised by replacing the case marker as in (54r). 22 candidates out of 162 could be revised in a similar manner, while the others cannot be salvaged. Since this ratio of frequency is not necessarily large, we conclude that the revision component for this type of error should divide its process into error detection and error correction and address the first task, namely, error detection.

Error types (c) and (d) can be also associated with case structures of verbs, even though our distillation step does not merge revision examples having different syntactic structures. However, their frequency is significantly smaller than those of (e). This means that addressing the problems of handling case assignments is one of the pressing issues. On the other hand, error type (f) sums up various types of collocation-related errors apart from (e). Typical collocation examples are those of “noun-GEN-noun,” “verb-verb,” “adverb-verb,” and “adjective-noun.”

Semantic errors

Ambiguous words can also be problematic, i.e., error type (g). For example, the words “*shiageru*” and “*kukuru*” have the same semantic concept “`3cf6fc:`to bring things or jobs to an end" in EDR (1995), and thus seem to be interchangeable. However, in the context of sentence (55t), it is more natural to think that "*kukuru*" takes the sense "**`3be172:`**to bind into one," which is inappropriate in this case. 12 erroneous paraphrase candidates out of 31 involved such ambiguous words.

(55) s. *okusan-gata-wa* [*hari-to* *ito-de* ***shiageru***]
mesdames-TOP needle-COM thread-IMP to finish-PRES
yukata-ni *ooyorokobi.*
informal kimono-DAT to be pleased-PRES
The mesdames are pleased with the informal kimono finalized with needles and thread.

t. ≠*okusan-gata-wa* [*hari-to* *ito-de* ***kukuru***]
mesdames-TOP needle-COM thread-IMP to bind-PRES
yukata-ni *ooyorokobi.*
informal kimono-DAT to be pleased-PRES
*The mesdames are pleased with the informal kimono bound by needles and thread.

On the other hand, sentence (56t) does not convey the same meaning as those of (56s) due to the vagueness of the word "*inshi* (factor)." In other words, it is more general than the source word "*idenshi* (gene)," although these words have the same concept "**`3bf4d0:`**a factor in a chromosome that attributes to an individual's genetic traits" in EDR (1995).

(56) s. [***idenshi**-o* *osaeru*] *koto-de* *gan-no* *seichou-o* *yokusei-dekiru.*
gene-ACC to inhibit-PRES thing-by cancer-DAT growth-ACC to control-be able to
We can control growth of cancer by inhibiting genes.

t. ≠[***inshi**-o* *osaeru*] *koto-de* *gan-no* *seichou-o* *yokusei-dekiru.*
factor-ACC to inhibit-PRES thing-by cancer-DAT growth-ACC to control-be able to
We can control growth of cancer by inhibiting factors.

These problems are considered to be matters of lexical choice between near-synonyms that require deep semantic processing. Edmonds (1999) have proposed an ontology for representing the sameness and idiosyncrasy between near-synonyms and thereby improved lexical choice in machine translation. In terms of knowledge acquisition, recent advances, such as (Okamoto *et al.*, 2003; Inkpen, 2003), have shown that utilizing existing synonym dictionaries such as WordNet (Miller *et al.*, 1990) and Kadokawa Synonym New Dictionary (Ôno and Hamanishi, 1981) is a feasible way of acquiring semantic differences between near-synonyms, although several issues still remain.

Discourse-related errors

When a system paraphrases a discourse element of the input texts, it can affect their cohesiveness.

Error type (i) indicates the discrepancies of tense and aspect forms in relative clauses and those in matrix clauses, as shown in example (45). On the other hand, error type (j) indicates the misuses of stylistic expressions including polite expressions. Although we observed only one example involving this type error, we speculate that this is because we examined only newspaper articles. Since there is a variety of Japanese polite expressions, even native speakers often misuse them (Kabaya *et al*., 1998; Shirado *et al*., 2003). However, by clarifying usages of each polite expression as being handled in (Inkpen, 2003), this type of error can be automatically detected and corrected.

Error type (k) is also one of the discourse-related errors. Word order in Japanese is relatively unrestricted compared with English. For example, sentence (57t) is unacceptable, while (57r) would be the more natural option.

(57) s. ***jumin-ga*** ***hantai-undou-o*** ***okoshi***, **(adverbial clause)**
residents-NOM opposition movement-ACC to raise (PAST)
kouji-ga *omouyoi-ni* *susum-anakat-ta.* **(matrix clause)**
construction-NOM wish-DAT to advance-NEG-PAST
Since residents caused an opposition movement, the construction did not progress as we planned.

t. ?***hantai-undou-ga*** ***jumin-kara*** ***okori***, **(adverbial clause)**
opposition movement-NOM residents-ABL to arise (PAST)
kouji-ga *omouyoi-ni* *susum-anakat-ta.* **(matrix clause)**
construction-NOM wish-DAT to advance-NEG-PAST

r. *jumin-kara* *hantai-undou-ga* *okori*, **(adverbial clause)**
residents-ABL opposition movement-NOM to arise (PAST)
kouji-ga *omouyoi-ni* *susum-anakat-ta.* **(matrix clause)**
construction-NOM wish-DAT to advance-NEG-PAST
Since an opposition movement arose from residents, the construction did not progress as we planned.

Finally, error types (l) and (m) in Table 3.2 indicate two sorts of discourse-related inappropriateness: theme-rheme structure and rhetorical structure. The distribution of these errors revealed that not only inter-clausal paraphrases, but also inner-clausal paraphrases can involve these errors when they switch topics or the subject. Paraphrase-based methods of cohesiveness criteria, such as those proposed in (Inui and Nogami, 2001; Siddharthan, 2003), should be incorporated into our transfer-and-revise paraphrase generation framework.

3.5 Summary

To justify our transfer-and-revise approach to paraphrase generation, we empirically confirmed that the transfer errors occurred irrespective of paraphrase classes. This suggests that if one creates a revision component specialized for a particular error type, it can be used for detecting, revising, and rejecting errors across different paraphrase classes.

The tasks of detecting and revising errors have been considered to be in the area of post-editing in the literature of transfer-based machine translation. For example, Knight and Chander (1994) addressed the issue of selecting particles in Japanese-English machine translation. Allen and Hogan (2000) focused on correcting numerous trivial errors in order to reduce the cost of manual post-editing. Recently, such an approach has been generalized into a framework of "candidate generation plus ranking," and applied to several tasks, such as decoding on statistical machine translation, similar sentence retrieval in example-based machine translation, and ranking of text generation. In contrast to the work above, our transfer-and-revise approach can be seen as an effort to maximally enhance the role of generation and post-editing in order to maximally reduce the load of transfer. The expected advantages are as follows:

- The incorporation of the revision component will reduce the redundancy and thus the complexity of the transfer knowledge.

- Our approach can be integrated with emerging corpus-based approaches to the paraphrase acquisition (Barzilay and McKeown, 2001; Lin and Pantel, 2001; Shimohata and Sumita, 2002a; Shimohata and Sumita, 2002b; Torisawa, 2002; Pang *et al.*, 2003; Shinyama and Sekine, 2003; Ohtake and Yamamoto, 2003; Quirk *et al.*, 2004), because the ability to revise transfer results will make the system tolerant of the deficiencies of automatically acquired knowledge.

By conducting an investigation into transfer errors, we empirically revealed what kinds of transfer errors tend to occur in lexical and structural paraphrasing by determining error distributions using over 28,000 transfer rules ranging from lexical paraphrases to structural paraphrases. Even though the present error classification does not necessarily cover every possible error occurring in paraphrase generation, the observations above draw a fine perspective for further research. We conclude that the detection of incorrect case assignments is one of the most serious problems that should have a priority to be solved. In the next chapter, we elaborate the difficulties of handling this type of error, and propose an automatic error detection model.

CHAPTER 4 Automatic detection of transfer errors in paraphrased sentences

4.1 Introduction

In Chapter 3, we investigated transfer errors in Japanese from two points of view: (i) what types of errors occur in generating lexical and structural paraphrases of Japanese sentences, and (ii) which of them tend to cause serious problems. The investigation revealed that case assignment is a major error source in paraphrasing of Japanese sentences. Examples (58) and (59) exemplify a typical example of incorrect case assignment in English: applying the paraphrasing rule "divide" ⇒ "decompose" generates correct sentence (58t) and incorrect one (59t).

(58) s. He **divided** the problem into three smaller subproblems.

t. He **decomposed** the problem into three smaller subproblems.

(59) s. He **divided** the cake into three pieces.

t. *He **decomposed** the cake into three pieces.

This disparity emerges because the word "decompose" requires words such a composite element as "sunlight," "milk," and "(complex) problem," for its direct object. Therefore, one may suspect that incorrect case assignments can be detected simply by referring to a handcrafted case frame dictionary which describes allowable cases and their **selectional restrictions** for each verb. However, in existing case frame dictionaries of Japanese, selectional restrictions are generally specified using coarse-grained semantic classes of noun. They are therefore not appropriate for the purpose of detecting incorrect case assignments.

To capture the difference between the usages of near-synonyms, we deal with words directly, instead of their semantic classes. Since a considerable number of positive examples, namely, correct examples of case assignments, can be collected from existing corpora, one can construct a **statistical language model** and apply it to the error detection task. In this chapter, we demonstrate that the state-of-the-art statistical language model can be improved by incorporating negative examples as well. The use of negative examples raises the following issues:

- Unlike positive examples, negative examples are generally not available. A challenging issue is therefore how to effectively use a limited number of manually collected negative examples combined with a large number of positive examples.

- Manual collection of negative examples is costly and time-consuming. Moreover, any such collection is sparse in the combinatorial space of words. Hence, we need an effective way to collect negative examples that truly contribute to error detection.

This chapter presents our empirical approach to this type of error. Section 4.2 elaborates the erroneous examples and difficulties of handling case assignments. Section 4.3 describes our error detection model. Section 4.4 shows the performance of our error detection model. Finally, Section 4.5 summarizes our achievements and further ideas to improve the model.

4.2 Incorrect case assignment

4.2.1 Three levels of incorrectness

Through the investigation in Chapter 3, we observed that case assignment tends to be incorrect, irrespective of the types of paraphrasing. A quarter of the paraphrased sentences (162 / 630) involved this type of error.

Case assignment can be incorrect at three different levels, namely, (i) violation of syntactic constraints, (ii) violation of selectional restrictions, and (iii) semantic inconsistency between sibling cases. We list some of the illustrative examples below.

(i) Violation of syntactic constraints

Though both of the verbs "*tessuru*" and "*tsuranuku*" have the same meaning "to devote," the paraphrased sentence (60t) is incorrect because "*tsuranuku*" cannot take the "*ni* (DAT)" case.

(60)	s.	*kare-wa*	*team play-ni*	***tessuru**.*
		he-TOP	team play-DAT	to devote-PRES
		He devotes himself to team play.		
	t.	**kare-wa*	*team play-ni*	***tsuranuku**.*
		he-TOP	team play-DAT	to devote-PRES

(ii) Violation of selectional restrictions

The verb "*katameru* (to strengthen)" requires a concrete object for its "*o* (ACC)" case. Hence, the noun "*kontei* (basis)" in the paraphrased sentence (61t) does not satisfy this constraint, thus making the sentence incorrect.

(61)	s.	*building-no*	***kiban-o***	*katameru.*
		building-GEN	foundation-ACC	to strengthen-PRES
		He strengthens the foundation of the building.		
	t.	**building-no*	***kontei-o***	*katameru.*
		building-GEN	basis-ACC	to strengthen-PRES
		*He strengthens the basis of the building.		

(iii) Semantic inconsistency between sibling cases

The nouns "*kotoba* (expressions)" and "*kakuchi* (every land)" in the paraphrased sentence (62t) satisfy the semantic constraint for "*ga* (NOM)" and "*ni* (LOC)" cases of the verb "*aru* (to exist)," respectively. Nevertheless, (62t) is incorrect, because of a semantic discrepancy between the nouns of the nominative and locative cases.

(62) s. *nankai-na* *kotoba-ga* ***zuisho**-ni* *aru.*
be crabbed-ADJ expressions-NOM many places-LOC to exist-PRES
There are crabbed expressions in many places (of the document).

t. **nankai-na* *kotoba-ga* ***kakuchi**-ni* *aru.*
be crabbed-ADJ expressions-NOM every land-LOC to exist-PRES
*There are crabbed expressions in every land.

4.2.2 Difficulties

As mentioned in Section 4.1, one may think that incorrect case assignments can be detected simply by referring to an existing case frame dictionary. This approach may be able to treat syntactic constraints, namely, subcategorization structures, because it is possible to enumerate all allowable cases for each verb. However, it has the following shortcomings on handling semantic constraints, namely, selectional restrictions:

- As we pointed out, even the difference between near-synonyms causes errors. For example, the difference between "*kiban* (foundation)" and "*kontei* (basis)" is crucial in the context of example (61). However, existing case frame dictionaries do not distinguish them, because they tend to specify selectional restrictions using a coarse-grained semantic typology[1]. This is because they are tailored by and large for disambiguation of verb phrases, in which case assignments are assumed to be correct.

- Manual construction and maintenance of a case frame dictionary is time-consuming and costly. Several attempts have been made to construct them with semi-automatic procedures (Kawahara and Kurohashi, 2001; Fujita and Bond, 2002a; Kawahara and Kurohashi, 2002; Fujita and Bond, 2002b; Bond and Fujita, 2003). However, these procedures tend to require tuning skills, otherwise nonnegligible human labor for correcting the resultant case frames.

- Case frame dictionaries rarely contain the information of idiomatic verb phrases and correspondences of cases between active forms and derivative forms such as passive and causative. We need such information to build a robust error detection model.

[1] For example, EDR (1995) classifies them together into "`3cf93c:`a basis of a thing or an action." The NTT Japanese Lexicon (Ikehara *et al.*, 1997) does "`2446:`basis and source."

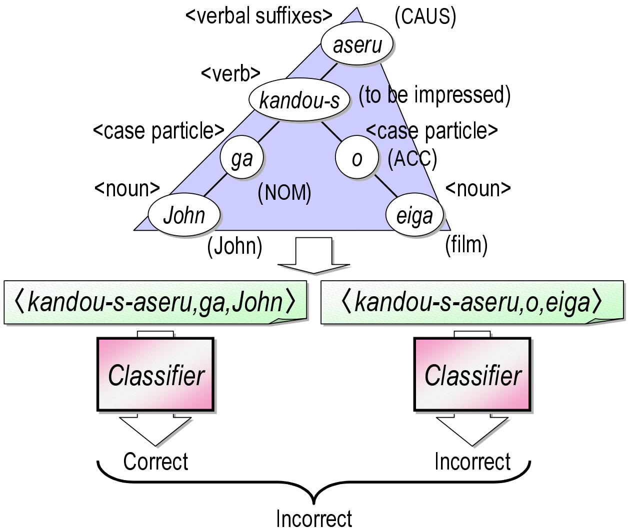

Figure 4.1. Classification of a paraphrase candidate.

4.3 Error detection models

Supposing that the case assignments in input sentences into paraphrase generation systems are all correct, the task of error detection of incorrect case assignments is defined as follows:

Input: A paraphrased sentence containing paraphrased case structures.

Output: "Incorrect" if the input is an anomaly; otherwise "correct."

In the rest of this section, we first describe our formulation of the task of error detection in Section 4.3.1, and motivation for the use of negative examples in Section 4.3.2. We then elaborate on our error detection model in Section 4.3.3 and selective sampling scheme for negative examples in Section 4.3.4.

4.3.1 Error detection as classification

On the basis of the discussion in Section 4.2.2, we develop language models that deal with lexical items directly to distinguish semantically similar words (**near-synonyms**). Let v, n and c be a verb, a noun and the case particle which is assigned to relate n to v, respectively. We decompose the detection of errors in a sentence into the classification of each triplet $\langle v, c, n\rangle$ (e.g., ⟨*katameru*, *o*, *kontei*⟩ in (61t)) contained in the given sentence into *correct* or *incorrect*. A given sentence is judged to be incorrect if and only if any of the triplets included in the sentence is classified as incorrect (c.f., Figure 4.1).

If we deal with $\langle v, c_1, n_1, c_2, n_2\rangle$ (e.g., ⟨*aru*, *ga*, *kotoba*, *ni*, *kakuchi*⟩ in (62t)) to take into account the association between two sibling cases, as in (Torisawa, 2002), we would be able to detect semantic inconsistency between sibling cases such as shown in example (62t). However, we firstly examine an error detection model taking only $\langle v, c, n\rangle$ into account because of the following reasons. First, building a language model of $\langle v, c_1, n_1, c_2, n_2\rangle$ is more likely to cause a data sparseness problem than taking only $\langle v, c, n\rangle$, because v and n have a great number of varieties. Second, although sibling cases can often be semantically inconsistent, in most cases it is wrong because of simpler reasons, namely, violation of either syntactic constraints or selectional restrictions. According to the analysis in Chapter 3, only 8 cases (4.9%) of the 162 incorrect case assignments were regarded as semantically inconsistent sibling cases.

To handle case assignments, we determine case structures based on dependency structures due to the following reasons. First, word ordering in Japanese is less restricted than that in English. Therefore, there is no guarantee that linear structure-based language models, e.g., n-grams, perform well in Japanese, although they can adequately predict human plausibility judgements in English, i.e., correctness of adjacent collocation (Lapata *et al*., 2001; Keller *et al*., 2003). Second, most of the paraphrase generation systems for Japanese rely on dependency structures (Kondo *et al*., 2001; Kaji *et al*., 2001; Takahashi *et al*., 2001). That is, such a system generates paraphrases individually annotated with a dependency structure, regardless of the transfer error that occurs.

4.3.2 Utilizing negative examples

In generative approaches to parsing and statistical machine translation, systems use statistics to estimate *relative likelihood* of output candidates. For the error detection in paraphrasing, however, we need a model for estimating the *absolute correctness* of output candidates in the sense that it must be capable of not only comparing candidates but also giving up producing output when none of the candidates is correct. Since paraphrase generation systems are typically developed for a particular purpose, such as simplifying text and controlling wording, they all feature a limited variety of transfer rules and thus can generate only inappropriate paraphrases. Hence a capability to reject all candidates is indispensable.

Error detection can be illustrated as a discriminative task, i.e., classifying the candidates into *correct* or *incorrect*, one may want to use both positive and negative examples to train a classifier. However, there is no available resource of negative examples. Moreover, even if we manually collected negative examples, any collection of negative examples is likely to be too small to represent the distribution of the negative class. Hence it is probably not a good choice to input them together with a vast amount of positive examples into a single classifier induction algorithm such as Support Vector Machines. Instead, we separately train two models, the positive model (*Pos*) and the negative model (*Neg*), then combine them to create another model (*Com*) as shown in Figure 4.2. Since negative examples have to be collected by hand, we also investigate the effectiveness of a selective sampling scheme to reduce human labor.

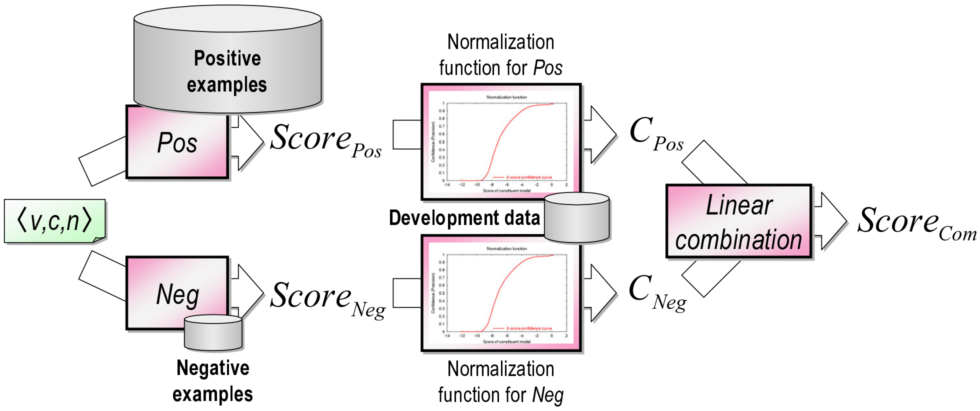

Figure 4.2. Calculating correctness of a case assignment.

4.3.3 Combining separately trained models

Positive model

Since a considerably large number of positive examples can be collected from existing corpora, one can estimate the probability $P(\langle v,c,n\rangle)$ with reasonable accuracy. On that account, we first construct a baseline model *Pos*, a statistical language model trained only on positive examples.

To calculate $P(\langle v,c,n\rangle)$ avoiding the data sparseness problem, one can use Probabilistic Latent Semantic Indexing (*PLSI*) (Hofmann, 1999), a variant of maximum likelihood estimation method. Since the formulation bases itself on **distributional clustering** (Pereira *et al.*, 1993), it offers an architecture to handle similarity between words according to their distributions. By dividing[2] a $\langle v,c,n\rangle$ into $\langle v,c\rangle$ and n, one can estimate $P(\langle v,c,n\rangle)$ by:

$$\begin{aligned} P(\langle v,c,n\rangle) &= P(\langle v,c\rangle,n) \\ &= \sum_{z\in Z} P(\langle v,c\rangle|z)P(n|z)P(z), \end{aligned} \tag{4.1}$$

where Z denotes a set of latent classes of co-occurrence. Probabilistic parameters $P(\langle v,c\rangle|z)$, $P(n|z)$, and $P(z)$ can be estimated from sufficiently large number of triplets $\langle v,c,n\rangle$ by applying the following EM algorithm (Hofmann, 1999; Rooth *et al.*, 1999).

E-step:

$$P(z|\langle v,c\rangle,n) = \frac{P(\langle v,c\rangle|z)P(n|z)P(z)}{\sum_{z'\in Z} P(\langle v,c\rangle|z')P(n|z')P(z')},$$

[2] $P(\langle v,c,n\rangle)$ can be represented by the product of $P(\langle v,c\rangle)$ and $P(n|\langle v,c\rangle)$. Each of the marginal distributions corresponds to an existing linguistic concept: the former indicates the likelihood of a case structure, while the latter does the satisfaction degree of semantic constraint on a case slot.

M-step:

$$
\begin{aligned}
P(\langle v,c\rangle|z) &= \frac{\sum_n f(\langle v,c\rangle,n)P(z|\langle v,c\rangle,n)}{\sum_{\langle v,c\rangle,n'} f(\langle v,c\rangle,n')P(z|\langle v,c\rangle,n')},\\
P(n|z) &= \frac{\sum_{\langle v,c\rangle} f(\langle v,c\rangle,n)P(z|\langle v,c\rangle,n)}{\sum_{\langle v,c\rangle',n} f(\langle v,c\rangle',n)P(z|\langle v,c\rangle',n)},\\
P(z) &= \frac{\sum_{\langle v,c\rangle,n} f(\langle v,c\rangle,n)P(z|\langle v,c\rangle,n)}{\sum_{\langle v,c\rangle',n'} f(\langle v,c\rangle',n')},
\end{aligned}
$$

where $f(\langle v,c\rangle,n)$ indicates the frequency of $\langle v,c,n\rangle$ in the given corpus.

The output of *Pos* is the estimate $Score_{Pos}(\langle v,c,n\rangle)$ of the likelihood of a given triplet $\langle v,c,n\rangle$. Given $P(\langle v,c,n\rangle)$, we can use various co-occurrence measures to compute $Score_{Pos}(\langle v,c,n\rangle)$. Well-known options are mutual information (*MI*), and the Dice coefficient (*Dice*) given by the following equations:

$$
\begin{aligned}
MI(\langle v,c,n\rangle) &= \log_2 \frac{P(\langle v,c,n\rangle)}{P(\langle v,c\rangle)P(n)},\\
Dice(\langle v,c,n\rangle) &= \frac{2\times P(\langle v,c,n\rangle)}{P(\langle v,c\rangle)+P(n)}.
\end{aligned}
$$

$P(\langle v,c,n\rangle)$ (*Prob*) itself is also a possible measure.

Negative model

Pos might not be able to properly judge the correctness of $\langle v,c,n\rangle$ by setting a simple threshold, particularly in cases where $P(\langle v,c\rangle)$ or $P(n)$ is low. This defect can be compensated for by means of negative examples. However, we cannot incorporate negative examples into the statistical language model directly, because the model is trained only on positive examples to determine the likelihood of given input. Hence we construct another error detection model *Neg* separately from *Pos*.

How do we calculate the correctness of the given triplet $\langle v,c,n\rangle$ using negative examples? One simple way is the k-nearest neighbor (k-NN) averaging method. Assuming that the distance between an input triplet $\langle v,c,n\rangle$ and a labeled negative example $\langle v',c',n'\rangle$ depends on both the distance between $\langle v,c\rangle$ and $\langle v',c'\rangle$ and that between n and n', we formulate the following distance function:

$$
Dist(\langle v,c,n\rangle,\langle v',c',n'\rangle) = DS\Big(P(Z|\langle v,c\rangle),P(Z|\langle v',c'\rangle)\Big)+DS\Big(P(Z|n),P(Z|n')\Big).
$$

Here, the probability distributions $P(Z|\langle v,c\rangle)$ and $P(Z|n)$ are regarded as the feature vectors for $\langle v,c\rangle$ and n, which are obtained through the EM algorithm for *Pos*. The function DS denotes **distributional similarity**[3] (Pereira *et al.*, 1993) between two probability distributions.

[3] Although it is called "similarity," actual value measures the distance (dissimilarity) between given two objects.

We employ one of the popular measures of distributional similarity, Jensen-Shannon divergence (DS_{JS}), which is proposed by Lin (1991) and well-examined in (Lee, 1999; Lapata *et al.*, 2001; Lee, 2001). Given a pair of probability distributions q and r, $DS_{JS}(q,r)$ is given by:

$$DS_{JS}(q,r) \quad = \quad \frac{1}{2}\left[D\left(q \;\Big\|\; \frac{q+r}{2}\right) + D\left(r \;\Big\|\; \frac{q+r}{2}\right)\right],$$

where the function D is the Kullback-Leibler divergence. DS_{JS} is always non-negative, and $DS_{JS} = 0$ iff $q = r$.

$$D\left(P_1(X) \,\|\, P_2(X)\right) \quad = \quad \sum_{x \in X} P_1(x) \log \frac{P_1(x)}{P_2(x)}.$$

Given an input $\langle v,c,n \rangle$, *Neg* outputs the weighted average distance $Score_{Neg}$ between the input and its k nearest neighbors. Formally,

$$Score_{Neg}(\langle v,c,n \rangle) \quad = \quad \frac{1}{k}\sum_{i=1}^{k} \lambda_i \, Dist\left(\langle v,c,n \rangle, \langle v',c',n' \rangle_i\right),$$

where λ_i is the weight for $\langle v',c',n' \rangle_i$, the i-th nearest neighbor of $\langle v,c,n \rangle$. As the value of $Score_{Neg}$ decreases, the input is more likely to be incorrect.

Combined model

Given a pair of scores output by *Pos* and *Neg*, our error detection model *Com* converts them into normalized confidence values C_{Pos} and C_{Neg} ($0 \leq C_{Pos}, C_{Neg} \leq 1$). Each normalization function can be derived using development data by the following steps:

Step 1. Train a model on training data, and estimate the score of each triplet $\langle v,c,n \rangle$ within development data[4].

Step 2. For each subinterval of score, compose a $\langle s,C \rangle$, where s indicates the center of the subinterval, while C does the ratio of correct triplets.

Step 3. Derive a normalization function by interpolating points between $\langle s,C \rangle$.

Com then outputs the weighted average of C_{Pos} and C_{Neg} as the overall score:

$$Score_{Com}(\langle v,c,n \rangle) \quad = \quad \beta\, C_{Pos}(\langle v,c,n \rangle) + (1-\beta)\, C_{Neg}(\langle v,c,n \rangle),$$

where β ($0 \leq \beta \leq 1$) determines the weights for the models, $Score_{Com}$ indicates the degree of correctness.

[4] In the experiment, we conduct 4-fold cross-validation over the training data in each step of 5-fold cross-validation.

4.3.4 Selective sampling of negative examples

We need negative examples that are useful in improving *Neg* and *Com*. For the current purpose, an example is not useful either if it is positive, or if it is similar to any of the known negative examples. In other words, we prefer negative examples that are not similar to any existing negative example.

Our strategy for selecting unlabeled examples is straightforward. We use *Pos* to estimate how likely an unlabeled example is negative. To compute the similarity between an unlabeled example and labeled examples, we use *Neg*. Let p_x be the estimated likelihood of an unlabeled example x, and s_x (> 0) be the similarity between x and its nearest negative example. Given an example x, its preference $Pref(x)$ is computed from the values of p_x and s_x, where a larger value of $Pref(x)$ indicates that x is more preferable for labeling.

Our selective sampling scheme is summarized as follows:

Step 1. Generate a set of paraphrases by applying paraphrasing rules to sentences sampled from documents in a given target domain.

Step 2. Extract a set of triplets from the set of paraphrases. We call it a *sample pool*.

Step 3. Sample a small number of triplets at random from the sample pool, and label them manually. Use only negative examples as the seed of the negative example set for *Neg*.

Step 4. For each unlabeled example x in the sample pool, calculate its preference by $Pref(x)$.

Step 5. Select the most preferred example, and label it manually. If it is negative, add it into the negative example set.

Step 6. Repeat steps 4 and 5 until a certain stopping condition is satisfied (e.g., the performance of error detection for development data does not improve).

4.4 Experiments in error detection

In this section, we describe experiments in error detection using manually labeled correct and incorrect paraphrase candidates.

4.4.1 Data and evaluation measures

We constructed data for training *Pos* and *Neg* in the following way (c.f., Figure 4.3). During this process, paraphrase candidates were constructed for evaluation as well.

Step 1. 53 million tokens (8.0 million types) of triplets $\langle v, c, n\rangle$ were collected from the parsed[5] sentences of newspaper articles[6]. To handle case alteration precisely, we dealt with active, passive, and causative forms of verbs separately.

[5] We used the statistical Japanese dependency parser CaboCha for parsing. http://chasen.org/~taku/software/cabocha/.

[6] Extracts from 9 years of the Mainichi Shinbun and 10 years of the Nihon Keizai Shinbun consisting of 25,061,504 sentences were used.

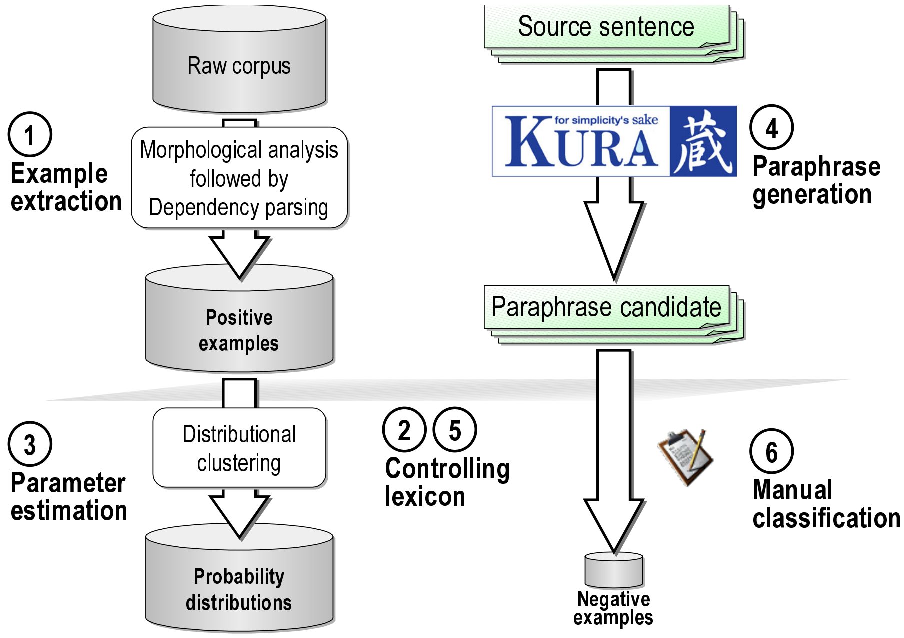

Figure 4.3. Model construction scheme.

Step 2. Triplets occurring only once were filtered out. We also restricted c to be the most frequent seven case particles: "*ga* (NOM)," "*o* (ACC)," "*ni* (DAT / LOC)," "*de* (LOC / IMP)," "*e* (to)," "*kara* (ABL)," and "*yori* (ABL / than)." This procedure resulted in 3.1 million types of triplets consisting of 38,512 types of n and 66,484 of $\langle v, c\rangle$.

Step 3. We estimated the probabilistic parameters of *PLSI* by applying the EM algorithm[7] to the data, changing the number of latent classes $|Z|$ from 2 through 1,500.

Step 4. To develop a negative example set, we excerpted 90,000 sentences from the newspaper articles used in Step 1, input them into our paraphrase generation system KURA, and obtained 7,167 paraphrase candidates by applying the same set of transfer rules as used for our previous investigation into transfer errors in Chapter 3.

Step 5. We filtered out the generated candidates that contained no changed case structure and those that included either v or n with a frequency of less than 2,000 in the collection given in Step 1. As a result, 3,166 candidates remained.

Step 6. Finally, we manually labeled the 3,166 candidates and the triplets included in the can-

[7] http://chasen.org/~taku/software/plsi/.
The EM algorithm requires the $(|Z||\langle v, c\rangle| + |Z||n| + |Z|) * 8$bytes (size of double) $*$ 2states$(old and new)$ of memories. For the lexicon above, $|Z| = 1{,}500$ is almost the limit for the number of latent classes, which occupies 2.34GB.

didates. We obtained (i) 2,358 positive and 808 (25.5%) negative paraphrase candidates[8], and (ii) 3,704 types of triplets consisting of 2,853 positive and 851 negative. The set of paraphrase candidates was used for evaluation, while the set of negative triplets was used for training *Neg* and *Com*.

For evaluation, we compared the performance of *Pos*, *Neg*, and *Com*. For each model, we set a threshold th and used it so that a given input was classified as erroneous if and only if it received a lower score than the threshold. Given such a threshold, recall R and precision P of the model are defined as follows:

$$R = \frac{\text{\# of correctly detected erroneous candidates}}{\text{\# of erroneous candidates}},$$
$$P = \frac{\text{\# of correctly detected erroneous candidates}}{\text{\# of candidates the model classified as erroneous}}.$$

Recall that a given paraphrase candidate is judged to be incorrect if and only if any of the triplets included in the candidate is classified as incorrect.

While we can estimate the optimal value of th for each model, in the experiments, we plot recall-precision (R-P) curves by varying the threshold. To summarize an R-P curve, we use the 11-point average precision where R is varied from 0.0 to 1.0 step 0.1. To compare two arbitrary R-P curves, we conduct the Wilcoxon rank-sum test using precision at eleven point above, assuming $p <$ 0.05 as the significance level.

4.4.2 Results

Baseline

First, to illustrate the complexity of the task, we show the performance of the baseline models: a dictionary-based model, a naive smoothing model, and variants of our statistical language model *Pos*. We regard *Pos* as a baseline because our concern is to what extent *Pos* can be enhanced by introducing negative examples. For the case frame dictionary, we used the largest Japanese case frame dictionary, the NTT Japanese Lexicon (Ikehara *et al.*, 1997) (*Dic*), while the Good-Turing estimation (*GT*) was employed as the naive smoothing model.

Dic classified a given $\langle v, c, n\rangle$ into correct or not according to the semantic typology of nouns assigned for each case frame of a verb if and only if both v and n were described in the dictionary. In our experiment, 338 paraphrase candidates (10.7%) were not judged; thus recall and precision were calculated for judged 2,828 candidates. At this, *Dic* achieved 41.9% precision and 61.6% recall.

GT estimates $P(\langle v, c, n\rangle)$ with significantly lower cost than *PLSI*. However, the recall for the model cannot be varied to lower value, because it does not distinguish the triplets that have the same frequency[9]. In our experiment, 530 negative examples out of 808 for evaluation were

[8] 41 (5.1%) paraphrase candidates out of 808 were incorrect due to semantic inconsistency between sibling cases.

[9] The Good-Turing estimation employs the value of $r^* = (r+1)\frac{N_{r+1}}{N_r}$ as the revised value of frequency r, where N_r denotes the sum of the frequencies of $\langle v, c, n\rangle$ that appeared just r times in the given corpus. Provided that, $P(\langle v, c, n\rangle)$ is given by $P(\langle v, c, n\rangle) = \frac{r^*}{N}$, where N denotes the total number of all $\langle v, c, n\rangle$ in the given corpus.

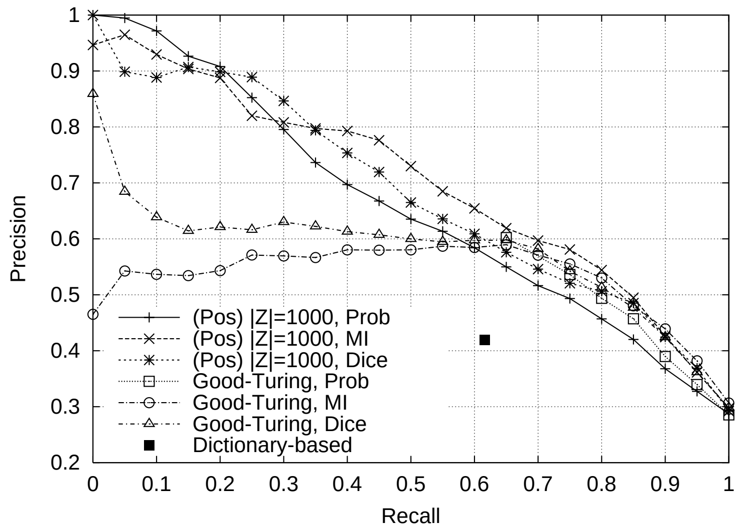

Figure 4.4. R-P curves of baseline models.

triplets that did not appear in 3.1 million types of collected examples. Therefore, the recall could not be varied to less than $R \leq 0.656$ (530 / 808). Even for higher recall, however, it did not perform well. *MI* and *Dice* based on *GT* allow varying recall to 0; however, they achieved only 51.9% and 58.0% 11-point average precision, respectively.

Figure 4.4 shows that *Pos* significantly outperformed both *Dic* and *GT*. *Prob*, *MI* and *Dice* with $|Z| = 1{,}000$ achieved 65.6%, 69.2% and 67.5% 11-point average precision, respectively; however, there was no significant difference among the measures. Their performance indicated that the *Pos* sufficed the conditions as a baseline to examine the use of negative examples. In general, *PLSI*, one of the smoothing methods, estimates high value of $P(\langle v, c, n\rangle)$ when probabilistic parameters $P(\langle v, c\rangle|z)$, $P(n|z)$ and $P(z)$ are high, even if the real $P(\langle v, c, n\rangle)$ is low. Co-occurrence measures, on the other hand, avoid this defect by referring to $P(\langle v, c\rangle)$ and $P(n)$, sacrificing the reliability of estimated score for those when either $P(\langle v, c\rangle)$ or $P(n)$ is low. These characteristics of measures were represented in the figure, namely, *Prob* outperformed *MI* and *Dice* for lower recall, while *MI* and *Dice* outperformed *Prob* for higher recall.

The performance of *Pos* is shown over the number of latent classes $|Z|$ in Figure 4.5. The larger $|Z|$ achieves higher 11-point average precision. However, overly enlarging $|Z|$ presumably does not work well because the performance of *Pos* almost hits a ceiling. The optimal $|Z|$ relies on the size of lexicon and sparsity of example distribution. Fortunately, the performance distribution over $|Z|$ looks moderate, particularly, there is no significant difference among the range of $|Z| \leq 200$. We therefore expect it can be estimated using development data at a reasonable cost.

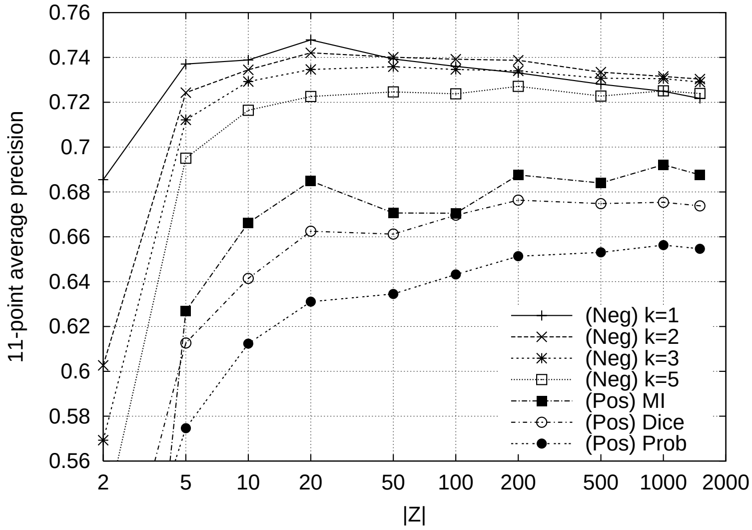

Figure 4.5. 11-point average precision of *Pos* and *Neg* over $|Z|$.

Properties of the negative model

Neg was evaluated by conducting 5-fold cross-validation over the labeled negative examples to keep training and test data exclusive. To give a higher reliability to a closer neighbor, the weighting function λ_i for i-th nearest neighbor was set to $1/i$, the reciprocal of the similarity rank. The 11-point average precision for combinations of parameters are shown in Figure 4.5. In contrast to *Pos*, the performance of *Neg* peaked at small $|Z|$. This is good news because a larger number of $|Z|$ obliges a higher computational cost for calculating distance, and thus determining the nearest neighbor. On the number of referring neighbors k, the 11-point average precision peaked at $k = 1$. We speculate that the negative examples are so sparse against the combinatorial space that a larger k causes more noise. We thus conclude that $k = 1$ is sufficiently reasonable for this task.

The performance of *Neg* may seem too high given the number of negative examples we used. Moreover, *Neg* outperformed *Pos* in 11-point average precision. It is, however, a plausible result due to the characteristics of the data. We speculate that the variety of triplets involved in paraphrase candidates is relatively small, because the set of transfer rules we used was built for the purpose of simplifying text (see also Chapter 3). Since it is common property in applications of paraphrase generation systems as mentioned in Section 4.3, we can expect that a limited number of negative examples can sufficiently cover the negative classes.

Combining models with selectively sampled examples

Using the 3,704 types of labeled triplets, we conducted simulations to evaluate the effectiveness of (a) combining *Pos* with *Neg* and (b) selective sampling of negative examples. We first sam-

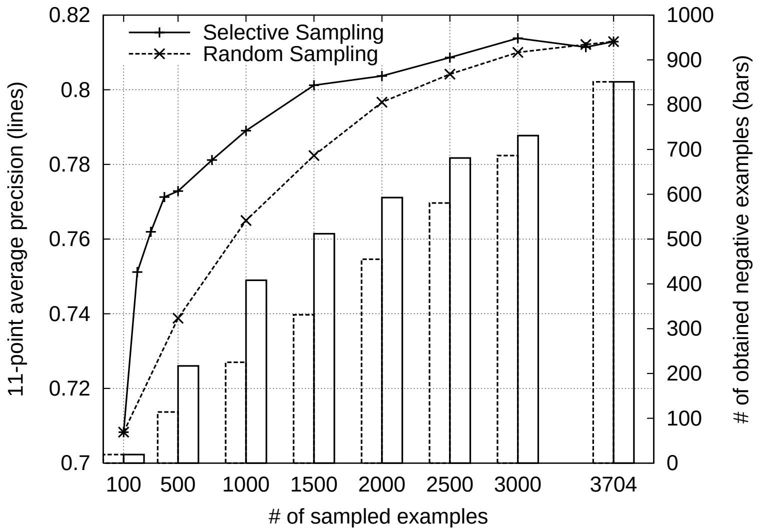

Figure 4.6. Learning curves of *Com*. Lines: 11-point average precision. Bars: # of obtained negative examples.

pled at random two sets of 100 samples from 3,704 labeled triplets. One involved 16 negative examples and the other 22. Using each negative example set, we then simulated the selective sampling scheme, regarding the remaining 3,604 triplets as the sample pool. Parameters and metrics employed were *Prob* and $|Z| = 1{,}000$ for *Pos*, $|Z| = 20$ and $k = 1$ for *Neg*. The preference function $Pref(x)$ was set to $-s_x \log_2(p_x)$, in which we preferred a sample x which satisfied both preference measure represented by p_x and s_x. Logarithm for p_x was introduced to give s_x more preference than p_x, since we would like to cover the quite sparse sample space quickly.

In each stage of selective sampling (learning), we formed a combined model *Com*, employing the parameters and metrics on which each component model performed best, i.e., *MI* and $|Z| = 1{,}000$ for *Pos*, and $|Z| = 20$ and $k = 1$ for *Neg*. Combining ratio β was set to 0.5. We then evaluated *Com* by conducting 5-fold cross-validations as well as for *Neg*. Figure 4.6 compares the performance of selective and random sampling, showing the averaged results for two seeds. In the figure, the horizontal axis denotes the number of sampled examples. The bars in the figure, which denote the number of obtained negative examples, designate that our preference function efficiently selects negative examples. The curves in the figure, which denote the performance curve with reference to 11-point average precision, designate a remarkable advantage of selective sampling, particularly in the early stage of learning.

Figure 4.7 illustrates the R-P curves of *Pos*, *Neg*, and *Com*. *Com* surpasses *Pos* and *Neg* over all ranges of recall. One can see that the models trained on selectively sampled negative examples exhibit R-P curves as nicely as the model with the largest negative example set. It is therefore confirmed that even if the collection of negative examples is not sufficient to represent

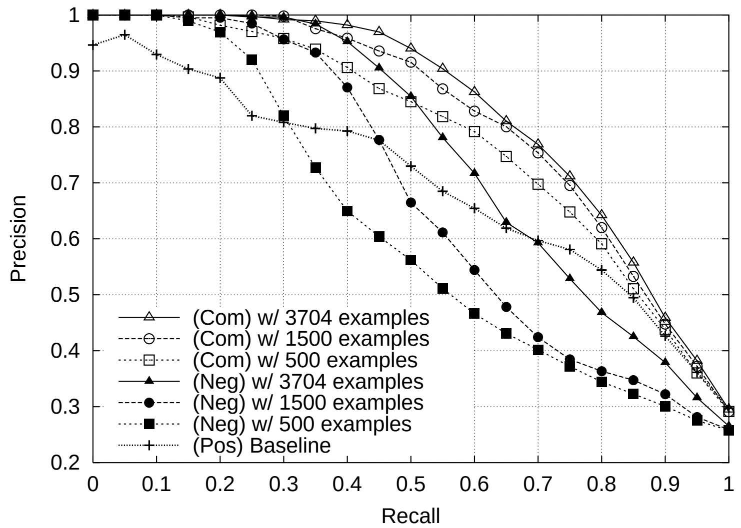

Figure 4.7. R-P curves of the proposed models.

the distribution of the negative classes, we can enhance the baseline model *Pos* by combining it with *Neg*. With the largest negative example set, *Com* achieved 81.3% 11-point average precision, a 12.1 point improvement upon *Pos*. The advantage gained over *Pos* was statistically significant. For the present settings, the performance peaked when a slightly greater weight was given to *Neg*, i.e., $\beta = 0.4$. However, it is reasonable to regard $\beta = 0.5$ as default, because there is no significant difference in performance between $\beta = 0.4$ and $\beta = 0.5$.

4.4.3 Discussion on parameter setting

The proposed model has six configurable parameters: (i) the number of latent classes $|Z|$, (ii) the co-occurrence measure of *Pos*, (iii) the number of referring neighbors k, (iv) the weighting function of nearest neighbors λ_i, (v) the combining ratio β, and (vi) the discrimination threshold th. There is no guarantee that the combination of the parameters which best performed in the above experiment always work well, because the data used in Section 4.4.1 was biased to contain frequency words. For actual application of the proposed model, we discuss the feasibility of parameter estimation. According to the experimental results shown in Section 4.4.2, we determine each parameter over development data independently.

Number of latent classes $|Z|$

Both component models of *Com*, *Pos* and *Neg*, depend on the probability distributions estimated via the distributional clustering, *PLSI*. One problem when conducting the clustering algorithm is how to determine the number of the clusters (latent classes $|Z|$ in the notion of distributional clustering).

To evaluate how a language model reduces the uncertainty about the data, several measures have been proposed. An information theoretic criteria, the Akaike information criterion (AIC), calculates how the model fits the data in consideration with the maximum likelihood of the data and the complexity of the model. In the minimum description length (MDL) principle, on the other hand, the sum of the data description length and the model description length are taken into account. Introducing the calculation of the complexity of models (or model description length) avoids selecting a model with larger numbers of parameters which tends to lead to over-fitting. AIC and MDL are expressed as follows:

$$\begin{aligned} AIC &= -2L + 2M, \\ MDL &= -L + \frac{M}{2}\log_2 N, \end{aligned}$$

where L denotes the maximum likelihood of the data, M the complexity of the model, and N the size of data. The values of L and M for a model built by our distributional clustering algorithm, are given by:

$$\begin{aligned} L &= -\sum_{\langle v,c,n\rangle \in C} f(\langle v,c,n\rangle)\log_2 P(\langle v,c,n\rangle) \\ M &= |Z|(|\langle v,c\rangle| - 1) + |Z|(|n| - 1) + (|Z| - 1) \\ &= |Z|(|\langle v,c\rangle| + |n| - 1) - 1, \end{aligned}$$

where $f(\langle v,c,n\rangle)$ indicates the frequency of $\langle v,c,n\rangle$ in the corpus C used for constructing the model, while $P(\langle v,c,n\rangle)$ is the estimated probability given by Eq. (4.1). The model that minimizes these values should be selected.

Another simple method for evaluating language models is the test-set perplexity. Given the test data S, the test-set perplexity of our model is given by 2^{H_T}, where H_T, which indicates the cross-entropy of S over the given model, is calculated as follows:

$$H_T = -\sum_{\langle v,c,n\rangle \in S} P_T(\langle v,c,n\rangle)\log_2 P(\langle v,c,n\rangle),$$

where $P_T(\langle v,c,n\rangle)$ denotes the empirical distribution of the sample $\langle v,c,n\rangle$ in the given test data S, while $P(\langle v,c,n\rangle)$ is the estimated probability given by Eq. (4.1).

Figure 4.8 shows the values of above measures over the number of latent classes $|Z|$. According to Figure 4.5, which shows that the performance of the model *Pos* stabilizes when $200 \leq |Z| \leq$ 1,500, we regard that only MDL is capable of estimating the optimal number of $|Z|$: the value of the measure peaked between $500 \leq |Z| \leq$ 1,000. As shown in Figure 4.5, the performance of *Neg* peaked at a smaller number of $|Z|$ than that of *Pos*. Hence we conclude that the (sub-) optimal number of $|Z|$ can be estimated using a sufficient number of held-out examples, namely, development data.

Co-occurrence measure of *Pos*

The co-occurrence measure of *Pos* must be selected according to the following two properties of metrics: (i) performance of error detection, and (ii) stability with respect to $|Z|$. Primarily,

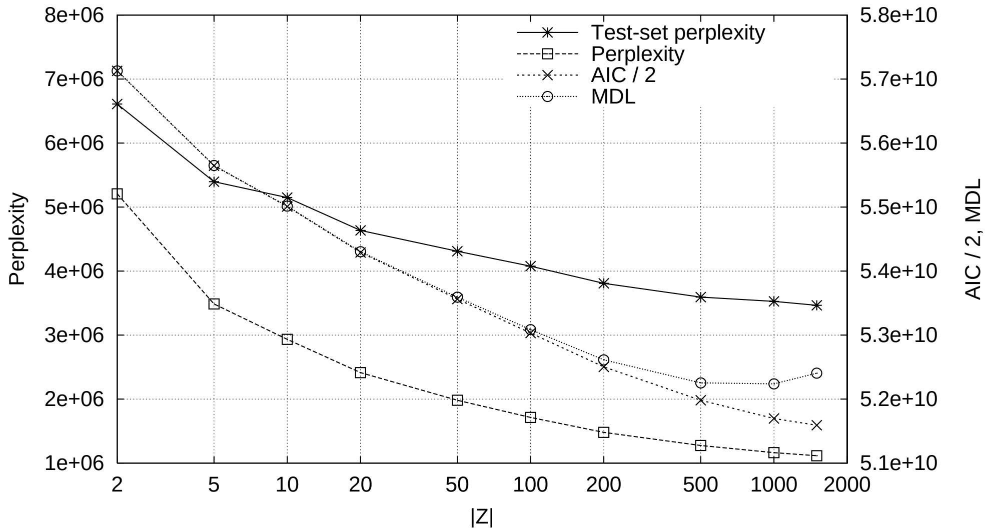

Figure 4.8. Estimating the optimal number of latent classes $|Z|$.

we should employ high performance metric. However, as we discussed in Section 4.4.2, it is often the case that no single metric beats the others. In such a case, their characteristics represented in R-P curves should be taken into account in order to select the metric. As we chose different metrics for error detection (*MI*) and selective sampling (*Prob*), utilizing the fortes of each metric is important. Second, we prefer a metric which is stable with respect to $|Z|$. Figure 4.5 provides the insight that all metrics of *Pos* performed moderately offering stability over $|Z|$.

Number of referring neighbors k and weighting function of nearest neighbors λ_i

As far as we concluded from Figure 4.5, $k = 1$ is sufficiently reasonable for this task. This is because any collection of negative examples is sparse in the combinatorial space for $\langle v, c, n \rangle$. Distance calculation according to large number of negative examples leads to noises (Figure 4.5). Accordingly, the problem of how to define the weighting function λ_i can be neglected.

Combining ratio β

Figure 4.9 depicts the 11-point average precision of *Com* over β in each stage of selective sampling of negative examples. The figure shows that the peak of 11-point average precision varies according to the set of negative examples, but in most cases the peak can be found when β is set to the range of 0.4 $\leq \beta \leq$ 0.5. Figure 4.9 indicates that the performance is relatively stable around the peak and thus we may be able to regard certain values in the range of 0.4 $\leq \beta \leq$ 0.5 as default. However, strictly speaking, we need to estimate optimal β

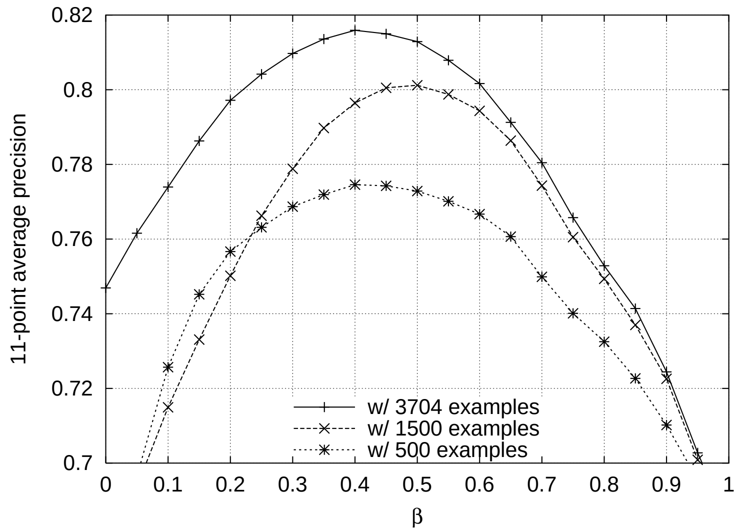

Figure 4.9. 11-point average precision of *Com* over β.

empirically. We speculate that given the set of negative examples, the optimal β can also be estimated using development data, because β is the sole parameter for combining *Pos* and *Neg*.

Discrimination threshold th

In this thesis, we evaluated each model by 11-point average precision. For the real error detection application, however, we need to determine the discrimination threshold th: a given input is judged to be an error if and only if the model scores lower than th. We think it is natural to determine the threshold according to what the application requires. For example, if the supposed application strictly requires correct paraphrases, the threshold ought to be set high, whereas the threshold can be set to low when paraphrases are allowed to be syntactically ill-formed, such as in query expansion for information retrieval. Cross-validation in our experiments showed the small variance of error detection performance. We thus conclude that the threshold can be determined using development data.

What criterion can be used for determining th? Given an error detection model, recall R and precision P are calculated against the number of erroneous examples when we discuss the performance of error detection (c.f., Section 4.4.1). On the other hand, to measure how well correct ones are preserved through the task of error detection, we take the following measures:

$$R_{cor} = \frac{\text{\# of correctly preserved correct candidates}}{\text{\# of correct candidates}},$$
$$P_{cor} = \frac{\text{\# of correctly preserved correct candidates}}{\text{\# of candidates the model classified as correct}}.$$

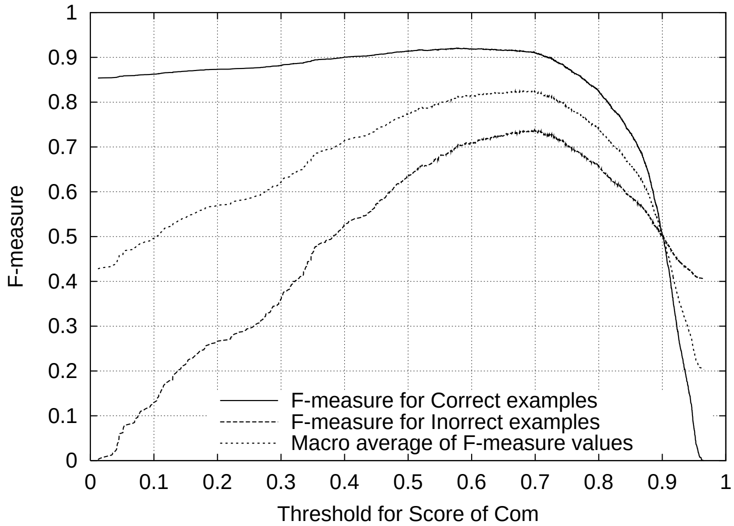

Figure 4.10. F-measure over the threshold for $Score_{Com}$.

To take into consideration both the performance of preserving correct examples and detecting erroneous ones, we first introduce two sorts of F-measures for identifying correct and incorrect examples as follows:

$$F_{cor} = \frac{2R_{cor}P_{cor}}{R_{cor}+P_{cor}},$$
$$F_{inc} = \frac{2R_{inc}P_{inc}}{R_{inc}+P_{inc}},$$

where $R_{inc} = R$, $P_{inc} = P$. Finally, taking the macro average of F_{cor} and F_{inc}, the arithmetic mean of the two values, we can evaluate how both correct and incorrect examples are identified by the given model.

Figure 4.10 shows several variations of F-measures over the threshold value of $Score_{Com}$. While F_{cor} kept high values when the errors were conservatively detected (when the th had low value), F_{inc} peaked when th was set to relatively high value ($th = 0.698$), where the model *Com* aggressively identifies examples as incorrect. However, this leads to some correct examples to be misidentified as incorrect ($R = 0.730$, and $P = 0.747$ in Figure 4.7). Such tendencies occurred because the number of correct examples was considerably larger than those of incorrect ones (2,358 vs. 808). An argument here is that the macro average of F_{cor} and F_{inc} can be a reasonable option to determine the discrimination threshold for the real application of error detection, because macro average takes both F_{cor} and F_{inc} into account.

4.5 Summary

We presented the task of detecting incorrect case assignment, a major error source in paraphrase generation of Japanese sentences. Our proposals are: (i) an empirical method to detect incorrect case assignments, where we enhanced a statistical language model by combining it with another model trained only on a small collection of negative examples, and (ii) a selective sampling scheme for effective collection of negative examples. Through the empirical experiments, we justified the feasibility of model construction and error detection.

There are several avenues for further enhancements for handling incorrect case assignments:

- The current design of *Neg* employs only negative examples; thus, the model simply outputs the distance to the nearest neighbor of given $\langle v, c, n\rangle$, even if the triplet is enclosed by the positive examples. To measure the correctness of such an idiosyncratic example, we need to seek an effective way of utilizing both positive and negative examples. Since the number of the two groups of examples are considerably imbalanced, we plan to examine an undersampling method such as described in (Akbani *et al.*, 2004) to choose positive examples which effectively work together with a limited number of negative examples.

- Our statistical language model *Pos* dramatically outperformed several conventional models in the task of error detection. However, it was not guaranteed that our devision of $\langle v, c, n\rangle$ into $\langle v, c\rangle$ and n was the best approach. We plan to examine various modeling method for likelihood estimation, such as those proposed in (Gildea, 2002). Furthermore, our models are not capable of detecting the semantic inconsistencies between sibling cases such as (62t). Analogously to the classification models proposed in (Gildea and Jurafsky, 2002), we plan to complementally use our model and those proposed in (Torisawa, 2002) to tackle the more complex problems.

- Correcting the detected errors is also an important issue. Our error analysis in Chapter 3 revealed that only a part of incorrect case assignments (22 / 162) could be corrected by replacing the case markers. Nevertheless, we should to look for a way to rescue them, since the aim of this study is to generate paraphrases accurately.

Through addressing the technical issues on detecting incorrect case assignments, we confirmed that statistical approaches to this level of transfer errors are feasible. Hence, our future plan also includes the application of the proposed model to the other collocative errors.

CHAPTER 5 *A lexical-semantics-based approach to paraphrase generation*

5.1 Introduction

This chapter addresses the issue of knowledge exploration for covering a wide range of paraphrases. In order to see potential problems in resource development, we firstly conducted a survey of conventional rule-based approaches to lexical paraphrases (Muraki, 1991; Sato, 1999; Kondo *et al.*, 1999; Kondo *et al.*, 2001; Takeuchi *et al.*, 2002). Through this analysis (part of this is described in Chapter 3), we observed that most semantic knowledge taken into account were associated with verbs. On the basis of this observation, in this chapter, we propose a novel lexical-semantics-based account of paraphrase generation based on the framework of the **Lexical Conceptual Structure** (LCS) (Jackendoff, 1990). The aim of this study is to explore what sorts of lexico-semantic properties of verbs can be explained by LCS.

The following sections describe the basic framework of LCS and our refinements on an existing LCS typology of Japanese verbs (Section 5.2), target paraphrase phenomena and related work (Section 5.3), our LCS-based paraphrase generation model (Section 5.4), and experiments (Section 5.5). Finally, Section 5.6 summarizes our findings with a brief description of work to be done in the future.

5.2 The Lexical Conceptual Structure

5.2.1 Basic framework

The Lexical Conceptual Structure (LCS) (Jackendoff, 1990) is a linguistic theory which associates a verb with a semantic structure as exemplified in Table 5.1.

An LCS consists of semantic predicates (“CONTROL,” “BE AT,” etc.) and their argument slots (x, y, z). There are two types of arguments, i.e., external arguments such as “Agent” and internal arguments such as “Theme” and “Goal.” Argument slots x, y, and z roughly correspond to the semantic roles “Agent,” “Theme,” and “Goal,” respectively. In the LCS of the verb “transmit,” for example, “[y MOVE TO z]” denotes the state of affairs that the state of the “Theme” changes to the “Goal,” and “[x CONTROL ...]” denotes that the “Agent” causes the state change. The predicate “BECOME” represents a change of state or change of physical location. The difference between “BECOME BE AT” and “MOVE TO” is underlying their telicity: the former indicates telic, while the latter atelic. The predicate “FILLED” means that the argument is implicitly filled with something. In the LCS of the verb “sign,” for example,

Table 5.1. Inventory of the T-LCS dictionary.

LCS for verb	Example verb / Example phrase
[x ACT]	*shigoto-suru* (to work), *kokyu-suru* (to breathe) I (Agent) work hard today.
[x ACT ON y]	*unten-suru* (to drive), *sousa-suru* (to operate) He (Agent) operates the machine (Theme).
[x CONTROL [y BE AT z]]	*hoshu-suru* (to maintain), *iji-suru* (to retain) He (Agent) maintains a machine (Theme) in good condition (Goal).
[x CONTROL [BECOME [y BE AT z]]]	*hon'yaku-suru* (to translate), *shoukai-suru* (to introduce) He (Agent) translates the book (Theme) into Japanese (Goal).
[x CONTROL [BECOME [x BE WITH y]]]	*ninshiki-suru* (to recognize), *yosoku-suru* (to predict) We (Agent) finally recognizes alternative medicine (Theme).
[x CONTROL [BECOME [y NOT BE AT z]]]	*yokushi-suru* (to deter), *shahei-suru* (to shield) He (Agent) shielded me (Goal) from hostile criticism (Theme).
[x CONTROL [BECOME [[FILLED]y BE AT z]]]	*shomei-suru* (to sign) Subjects (Agent) have to sign the contract (Goal) before the exams.
[x CONTROL [y MOVE TO z]]	*denpan-suru* (to transmit), *enki-suru* (to postpone) The enzyme (Agent) transmits messages (Theme) to the muscles (Goal).
[y MOVE TO z]	*ido-suru* (to move), *sen'i-suru* (to propagate) My sister (Theme) moves to a neighboring town (Goal).
[y BE AT z]	*ichi-suru* (to locate), *sonzai-suru* (to exist) The school (Theme) locates near the river (Goal).
[BECOMNE [y BE AT z]]	*houwa-suru* (to become saturate), *bunpu-suru* (to be distributed) This flower (Theme) is distributed all over the world (Goal).
[x=y CONTROL [BECOME [y BE AT z]]]	*kaifuku-suru* (to recover), *teigen-suru* (to decrease), *kanryou-suru* (to complete) Our method (Agent) decreased the cost (Theme) to approximately $1,000 (Goal). The cost (Theme) decreased to approximately $1,000 (Goal).

"FILLED" denotes the name of the "Agent." Provided that, when the verb which has this type of argument governs an element for this argument, the element can override the argument in any case.

Several studies in linguistics such as (Jackendoff, 1990; Kageyama, 1996) have shown that the LCS framework provides a systematic explanation of semantic decomposition as well as the syntax structure. There exists a large-scale LCS dictionary for English (Dorr, 1997) which was tailored based on a verb classification introduced in (Levin, 1993) with an expansion for the thematic role delivered to arguments. The dictionary contains rich information that works well in conjunction with other resources such as WordNet (Miller *et al.*, 1990), FrameNet (Baker *et al.*, 1998), and Proposition Bank (Gildea and Palmer, 2002). Kageyama (1996), on the other hand, has shown that some kinds of simple LCS can explain the word formation of compound nouns, transitivity alternation, and derivations of words as well as the grammatical structure of sentences in typical cases. Although LCS can be expected to be useful for explaining various linguistic phenomena, existing LCS frameworks can not be straightforwardly applied to the explanation of paraphrases in Japanese. Because they do not provide an exhaustive list of LCS nor is there any objective criterion for assigning LCS to lexical elements.

Recently, an LCS dictionary for Japanese verbs has been developed (Takeuchi *et al.*, 2002). In this thesis, we make use of this dictionary, the T-LCS dictionary, because it offers the following advantages:

- It is based on a linguistic work, as in (Kageyama, 1996).
- Its scale is considerably larger than any other existing collections LCS entries.

- It provides a set of concrete rules for LCS assignment, which ensures the reliability of the dictionary.

5.2.2 Refinements

In spite of some of the above advantages, our preliminary examination of the T-LCS dictionary version 0.9 (Takeuchi *et al*., 2002) revealed that further refinements were needed. We newly established the following four types of new LCS according to the pieces of linguistic analyses in (Kageyama, 2001; Kageyama, 2002). The first two have been incorporated into the current version of the T-LCS dictionary (version 0.95) (Takeuchi, 2004). For convenience, we refer to the extended dictionary as the LCSdic.

Verbs of obtaining

The nominative cases "*ga*" of "*ukeru* (to receive)" and "*eru* (to acquire)" indicate "Goal" of a "Theme." Likewise, the ablative cases "*kara*" indicate "Source." Verbs of obtaining (Levin, 1993, p.142) have such a case structure. Although the T-LCS typology in (Takeuchi *et al*., 2002) does not take into account the ablative cases, they need to be represented in LCS, because they can be altered with either nominative "*ga*," accusative "*o*," or dative "*ni*."

(63)	*ten'in-ga*	*kyaku-kara*	*kujo-o*	***ukeru***.
	salesclerk-NOM	customer-ABL	complaint-ACC	to receive-PRES
	The salesclerk receives a complaint from a customer.			

There are lexical active-passive relationships which can be seen between "*oshieru* (to teach)" and "*osowaru* (to be taught)," "*sazukeru* (to grant)" and "*sazukaru* (to be granted)." Kageyama (2002) gave the following explanation for these derivative relationships:

(64) Derivation of "*osowaru* (to be taught)" from "*oshieru* (to teach)"

oshieru [x CONTROL [y MOVE FROM x TO z]]

$\Downarrow$ transformation into possession form

oshieru [x CONTROL [$\underline{\text{BECOME } [z \text{ BE WITH}}$ [y MOVE FROM x TO z]]]]

$\Downarrow$ decausativization

osowaru [$\underline{\phi}$ CONTROL [BECOME [z BE WITH [y MOVE FROM x TO z]]]]

$\Downarrow$ simplification

osowaru [BECOME [z BE WITH [y MOVE FROM x TO z]]]

According to this derivation, we defined the LCS for these verbs as follows:

(65) [BECOME [z BE WITH [y MOVE FROM x TO z]]]
(For (63), x:customer, y:complaint, z:salesclerk)

Require verb

"*motomeru* (to ask)" and "*yokyu-suru* (to require)" denote the existence of the external "Agent" who controls the action of the other "Agent" or "Theme."

(66) s. *Ken-ga* *George-ni* *shazai-o* ***motomeru***.
Ken-NOM George-DAT apology-ACC to ask-PRES
Ken asks George for an apology.

(67) [x CONTROL [y MOVE FROM z TO [FILLED]]]
(For (66), x:Ken, y:apology, z:George)

Treatment of "Partner"

Verbs such as "*hankou-suru* (to oppose)" and "*sansei-suru* (to agree with)" subcategorize the dative case "*ni*." However, as sentence (68) illustrates, it does not denote the "Goal" of the action, but the "Partner."

(68) *Ken-ga* *oya-ni* ***hankou-suru***.
Ken-NOM parents-DAT to oppose-PRES
Ken opposes his parents.

Nonetheless, it has to be represented in LCS because the element of "*ni* (DAT)" can be reassigned to the nominative case "*ga*" when the verb is passivized as sentence (69).

(69) *oya-ga* *Ken-ni* ***hankou-s-are-ru***.
parents-NOM Ken-DAT to oppose-PASS-PRES
Ken's parents are opposed by him.

In addition, an aspectual test using an adverb "*takusan* (a lot)" indicates that those verbs do not mean the change of state but transient action. We therefore newly defined the following LCS:

(70) [x ACT TO z]
(For (68), x:Ken, z:parents)

Verbs of psychological state

"*kandou-suru* (to be impressed)" and "*osoreru* (to fear)" indicate the change of psychological state of the "Agent." The ascriptive factors of such changes have to be represented.

(71) *Ken-ga* *ongaku-ni* ***kandou-suru***.
Ken-NOM music-DAT to be impressed-PRES
The music impressed Ken. / Ken is impressed by the music. (more Japanese-like)

For a long time, semantic structures for psych-verbs (Levin, 1993, p.188) have been undecided in lexical semantics (Kageyama, 2001; Takeuchi, 2004; Hatakeyama *et al*., 2005). We tentatively defined the LCS for these verbs as follows:

(72) [BECOME [z BE WITH [[FILLED]y MOVE FROM x TO z]]]
(For (71), x:music, z:Ken)

5.3 Paraphrasing of light-verb constructions

5.3.1 Target structure and required operations

In Japanese, like other languages, there are several paraphrase classes that exhibit a degree of regularity that allows them to be systematically explained by a limited number of general rules and lexico-semantic knowledge. For example, paraphrases associated with voice alteration, verb/case alteration, compounds, and lexical derivations all fall into such classes. In this chapter, we focus our discussion on another useful paraphrase class, namely, paraphrasing of light-verb constructions (LVCs)[1], and propose a computational model for generating paraphrases of this class.

Sentence (73s) is an example of an LVC. An LVC is a verb phrase ("*kandou-o atae-ta* (made an impression)") that consists of a light-verb ("*atae-ta* (to give-PAST)") that syntactically governs a nominalized verb ("*kandou* (an impression)") (c.f., Figure 5.1). A paraphrase of (73s) is sentence (73t), in which the nominalized verb functions as the main verb with its verbal form ("*kandou-s-**ase**-ta* (to be impressed-CAUS-PAST)").

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

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

Figure 5.1 demonstrates tree representations of source and target expressions involved in LVC paraphrasing, taking (73) as an example. The oval objects in the figure denote Japanese base-chunk so-called *bunsetsu*. To generate this type of paraphrase, we need a computational model that is capable of the following operations for involved modifiers:

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

Re-conjugation: Change the conjugation form of the elements (d) and occasionally (c)[2], according to the category change of their modifiee: the given nominalized verb becomes a verb. As we discussed in Section 3.4.2, this operation can be carried out easily, independently of the LVC paraphrasing.

Selection of the voice: The model must choose the voice of the target sentence from active, passive, causative, etc. In example (73), the causative voice is chosen, which is indicated

[1] Miyamoto (1999) has considered only "*suru* (to do)" as a light-verb. However, various verbs can actually function as a light-verb, namely, the meaning of the given light-verb construction is represented by the nominalized verb (Muraki, 1991).

[2] The most delicate element is (c) because it acts either as an adverb or as a case depending on its context. In the former case, it requires a change of conjugation, while it needs the reassignment of the syntactic case as well as the element (b) in the latter case.

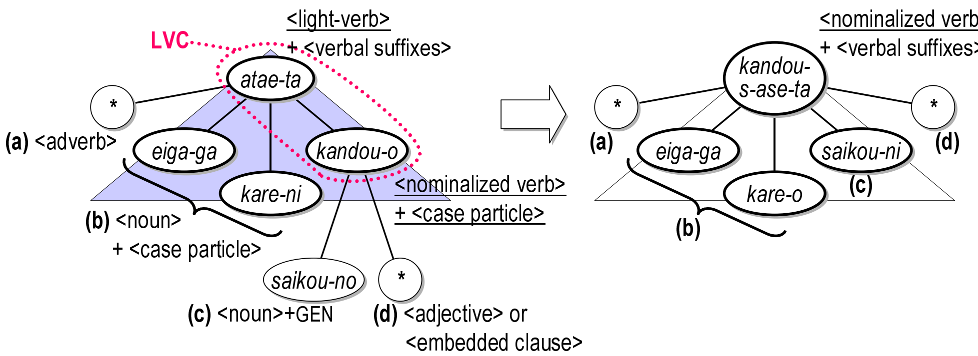

Figure 5.1. Dependency structure showing the range which the LVC paraphrasing affects.

by the auxiliary verb "*ase* (CAUS)." The task is not as simple as it may seem. Muraki (1991) pointed out that the decision depends not only on the syntactic and semantic attributes of the given light-verb, but also on those of the nominalized verbs.

Reassignment of the cases: The model must reassign case markers of the elements (b) and (c), the arguments of the main verb. In (73), the syntactic case of "*kare* (him)," which was originally assigned the dative case "*ni*" is changed to accusative "*o*."

In this chapter, we realize a paraphrasing model which carries the last two operations relying on the framework of LCS. At the moment, we focus on handling the element (b), namely, the sibling cases of the nominalized verb. Triangles in both trees in Figure 5.1 indicate the range which we handle. Elements outside of the triangles, i.e., (a), (c), and (d), are used only for explanatory purposes.

5.3.2 Related work

The paraphrases associated with LVCs are not idiosyncratic to Japanese but also appear commonly in other languages such as English and French (Mel'čuk and Polguère, 1987; Iordanskaja *et al.*, 1991; Dras, 1999). Dras (1999) exemplified the following:

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

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

Our approach raises the interesting issue of whether the paraphrasing of LVCs can be modeled in an analogous way across languages.

Iordanskaja *et al.* (1991) proposed a set of paraphrasing rules including one for LVC paraphrasing based on the Meaning-Text Theory introduced by (Mel'čuk and Polguère, 1987). The model seemed to handle the paraphrasing of LVC correctly, because their rules were described according to the deep semantic analysis and heavily relied on what were called lexical functions, such as lexical derivation (e.g., $S_0(affect) = effect$), alteration between antonym (e.g., $Anti(victory) = low$), and possible light-verb (e.g., $Oper_1(attempt) = make$). To take this approach, however, we need a vast amount of lexical knowledge to form each lexical function, because they only virtually specified all the choices relevant to LVC paraphrasing for every combination of nominalized verb and light-verb individually. In contrast, our approach is to employ lexical semantics to provide a general account of those classes of choices.

On the other hand, (Kaji and Kurohashi, 2004) proposed paraphrase generation models which utilized an ordinary dictionary. Given an input LVC, their model paraphrases it referring to the glosses of both the nominalized verb and light-verb, and a manually assigned semantic feature of the light-verb. Their model looks robust because of the availability of the ordinary dictionary. However, their model fails to explain the difference in the voice selection between examples (76) and (77) since it selects the voice based only on the light-verb — in their approach, the light-verb "*ukeru* (to receive)" is always mapped to the passive voice irrespective of the nominalized verb.

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

t. *Ken-ga* *eiga-ni* ***shigeki-s-are****-ta.*
Ken-NOM film-DAT to inspire-PASS-PAST
Ken was inspired by the film.

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

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

In their model, the target expression is restricted only to the LVC itself (c.f., Figure 5.1). Hence, their model is unable to reassign the case markers as we saw in example (73).

5.4 LCS-based paraphrase generation model

This section describes how we generate paraphrases of LVCs. Figure 5.2 illustrates how our model paraphrases the LVC of example (76). The idea is to exploit the LCS representation as a semantic representation and to model the LVC paraphrasing by the transformation of the LCS representation. The process consists of the following three steps:

Step 1. Semantic analysis: The model first analyzes a given input sentence including an LVC to obtain its LCS representation. In Figure 5.2, this step produces LCS_{V1} by filling arguments of LCS_{V0} with nominal elements.

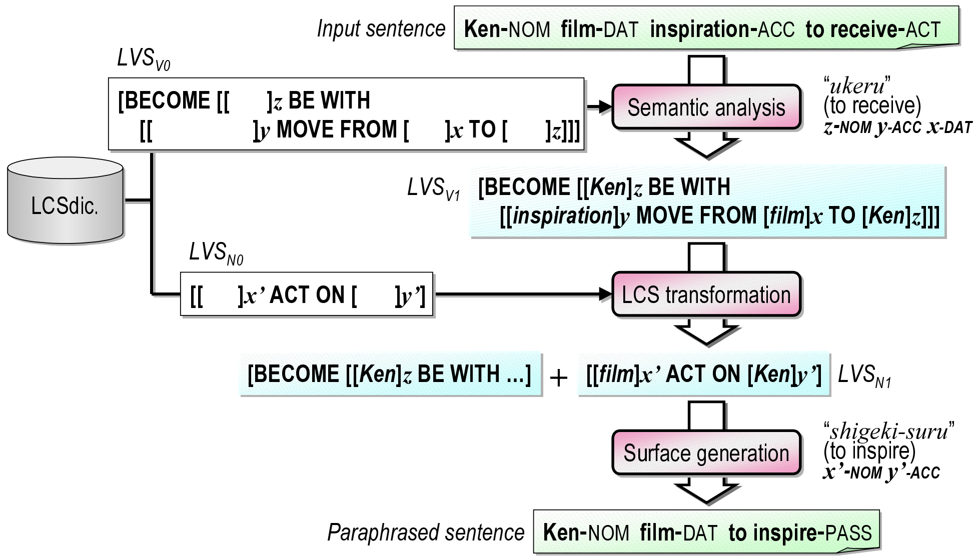

Figure 5.2. LCS-based paraphrase generation model.

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

Step 3. Surface generation: Having obtained the target LCS representation, the model finally lexicalizes it to generate the output sentence. LCS_S is interpreted as a signal of syntactic alternation such as passivization and causativization.

So, the key issue is how to control the second step, namely, the transformation of the LCS representation.

The rest of this section elaborates on each step, differentiating symbols to denote arguments; x, y, and z for LCS_V, and x', y', and z' for LCS_N.

5.4.1 Semantic analysis

Given an input sentence, which we assume to be a simple clause with an LVC, we first look up the LCS template LCS_{V0} for the given light-verb in the LCSdic, and then apply the **case assignment rule** below (Takeuchi *et al.*, 2002) to obtain its LCS representation LCS_{V1}:

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

- Otherwise, fill arguments y and z of the LCS_{V0} with the nominative and the dative, respectively.

In the example shown in Figure 5.2, the nominative "*Ken*" fills the leftmost argument z. Accordingly, the accusative "*shigeki* (inspiration)" and the dative "*eiga* (film)" fill y and x, respectively.

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

LCS_{V0} [BECOME [z BE WITH [y MOVE FROM x TO z]]]

LCS_{V1} [BECOME [[*Ken*]z BE WITH [[*inspiration*]y MOVE FROM [*film*]x TO [*Ken*]z]]]

5.4.2 LCS transformation

The second step of our paraphrase generation model matches the resultant LCS representation (LCS_{V1} in Figure 5.2) with the LCS of the nominalized verb (LCS_{N0}) to generate the target LCS representation (LCS_{N1}). Figure 5.3 shows a more detailed view of this process for the example shown in Figure 5.2.

Predicate and argument matching

The first step is to determine the predicate in LCS_{V1} that should be matched with the predicate in LCS_{N0}. Assuming that only the agentivity is relevant to the selection of the voice in LVC paraphrasing, which is our primary concern, we classify the semantic predicates into the following classes:

Agentive predicates: "CONTROL," "ACT ON," "ACT TO," "ACT," "BE AGAINST," and "MOVE FROM TO."

State of affair predicates: "MOVE TO," "BE AT," and "BE WITH."

Aspectual predicates: "BECOME."

We also assume that any pair of predicates of the same class is allowed to match, and that the aspectual predicates are ignored. In our example, "MOVE FROM TO" matches "ACT ON" as shown in Figure 5.3.

LCS representations have right-branching (or right-embedding) structures. Since inner-embedded predicates denote the state of affairs, they take priority in the matching. In other words, the matching proceeds from the rightmost inner predicates to the outer predicates.

Having matched the predicates, we then fill each argument slot in LCS_{N0} with its corresponding argument in LCS_{V1}. In Figure 5.3, argument z is matched with y', and x with x'. As a result, "*Ken*" comes to the y' slot and "*eiga* (film)" comes to the x' slot[3].

This process is repeated until the leftmost predicate in LCS_{N0} or that in LCS_{V1} is matched.

[3] When an argument is filled with another LCS, arguments within the inner LCS are also matched. Likewise, with regard to an assumption that the input sentences are periphrastic, we introduced some exceptional rules. That

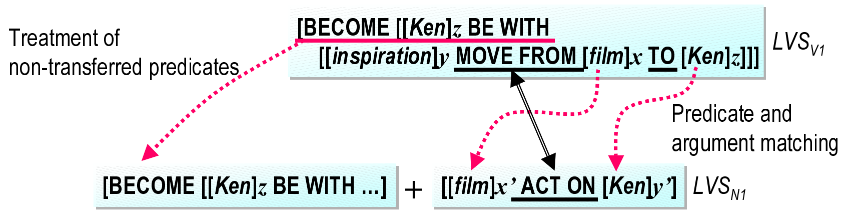

Figure 5.3. An example of LCS transformation.

Treatment of non-transferred predicates

If LCS_{V1} has any non-transferred predicates when the predicate and argument matching has been completed, they represent the semantic content that is not conveyed to LCS_{N1} and which needs to be lexicalized by auxiliary linguistic devices such as voice auxiliaries. In the case of Figure 5.3, "[BECOME [[*Ken*]z BE WITH]]" in LCS_{V1} remains non-transferred. In such a case, we attach the non-transferred predicates LCS_S to LCS_{N0}, which are then lexicalized by auxiliaries in the next step, the surface generation.

5.4.3 Surface generation

We again apply the aforementioned case assignment rule to generate a sentence from the resultant LCS representation. From the LCS_{N1} in Figure 5.2, (79t) is firstly generated.

(79) LCS_S [BECOME [[*Ken*]z BE WITH]]

LCS_{N1} [[*film*]x' ACT ON [*Ken*]y']

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

The model then makes the final decisions on the selection of the voice and the reassignment of the cases, according to the following decision list:

1. If the leftmost argument of LCS_S has the same value as the leftmost argument in LCS_{N1}, their viewpoints is same. Therefore, the active voice is selected and the case structure is left as is.

2. If the leftmost argument of LCS_S has the same value as either z' or y' in LCS_{N1}, lexicalization is performed to make the argument a subject. Therefore, the passive voice is selected and case alternation (passivization) is applied.

3. If LCS_S has "BE WITH" and its argument has the same value as x' in LCS_{N1}, the causative voice is selected and case alternation (causativization) is applied.

is, arguments filled with the implicit filler represented by "FILLED" or the target nominalized verb N are never matched, and z in LCS_{V1} can be matched to y' in LCS_{N0}.

4. If LCS_S has an agentive predicate, and its argument is filled with a value different from those of the other arguments, then the causative voice is selected and case alternation (causativization) is applied.

5. Otherwise, active voice is selected and thus no modification is applied.

Since the example in Figure 5.2 satisfies the second condition, the model chooses "*s-**are**-ru* (PASS)" and passivizes the sentence (79t). As a result, "*Ken*" fills the nominative case "*ga*" as in (80t).

(80) t. *Ken-**ga*** *eiga-**ni*** ***shigeki-s-are**-ta.*
Ken-NOM film-DAT to inspire-PASS-PAST
Ken was inspired by the film.

5.5 Experiment

To empirically evaluate our paraphrase generation model and the current typology of T-LCS, and to clarify the remaining problems, we analyze a set of automatically generated paraphrase candidates.

5.5.1 Resources

To conduct an experiment, we collected the following sets of words as the entries of the LCSdic. Note that more than one LCS was assigned to a verb if it was polysemous.

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

Light-verbs: Although Muraki (1991) listed a number of verbs which can form LVCs, it is possible to be prejudiced by his analysis. We therefore tailor an objective collection of light-verbs. We also take into account that a verb takes different meanings when it is a part of LVCs with different case particles, and a tuple of a light-verb and a case particle does not necessarily function as an LVC. To this end, we collected frequent tuples $\langle v, c \rangle$, in which c and v indicates a case particle and a verb, respectively, from corpus in the following manner.

Step 1. We collected 876,101 types of triplets $\langle v, c, n \rangle$, where v, c, and n denote a base form of verb, a case particle, and an nominalized verb, from the parsed sentences of newspaper articles[4].

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

[4] This is just the same set as those we used in Section 4.4.

Step 3. Each $\langle v, c, n\rangle$ was manually evaluated to determine whether it was an LVC. If any of 10 triplets was determined to be an LVC, the tuple $\langle v, c\rangle$ was merged into the list of light-verbs. As a result, we collected 40 types of $\langle v, c\rangle$ for light-verbs.

The set of pairs of an LVC and its correct paraphrase were manually constructed in the following way:

Step 1. From the 876,101 types of triplet $\langle v, c, n\rangle$ collected during building a vocabulary list, 23,608 types of $\langle v, c, n\rangle$ were extracted, whose components, n and $\langle v, c\rangle$, were listed in the LCSdic.

Step 2. For each of the 245 most frequent $\langle v, c, n\rangle$, the 3 most frequent simple clauses including the $\langle v, c, n\rangle$ were extracted from the same corpus. As a result, we collected 735 simple clauses (sentences).

Step 3. A paraphrase for each sentence was manually described. 24 sentences out of 735 could not be paraphrased because $\langle v, c, n\rangle$ within them did not behave as LVCs. Therefore 711 paraphrases were described as a gold-standard.

Coverage of these 245 $\langle v, c, n\rangle$ with regard to the all LVCs among corpora can be calculated as follows. When we assume all $\langle v, c, n\rangle$ are LVCs, then the coverage of our test data is estimated at 6.47% (492,737 / 7,621,089). Likewise, when we assume our dictionary covers all light-verbs, even though this is a bold assumption, the coverage is estimated at 24.1% (492,737 / 2,044,387). The real coverage fails the range between the two values above, and we think the ratio is reasonably high for the empirical evaluation.

5.5.2 Paraphrase generation and evaluation

For the 735 sentences, our paraphrase generation model automatically generated 822 paraphrase candidates, that is, at least one for each input. This indicated that our model generated all possible paraphrase candidates, when more than one LCS was assigned to a verb in the LCSdic it was due to its polysemy. We manually classified the resultant 822 paraphrase candidates into correct and incorrect according to the gold-standard we built above. As a result, we obtained 624 correct and 198 incorrect paraphrase candidates. Finally, the error detection model built in Chapter 4 was applied to the set of paraphrase candidates to filter out anomalies. The same parameters and measures were employed as those used for building our model *Com*. Additionally, in this experiment, the threshold for discrimination was set to 0.698 according to the discussion on parameters in Section 4.4.3. Our error detection model discarded 66 incorrect paraphrases, sacrificing 15 correct ones. Table 5.2 summarizes the results in several measures, where recall, precision, and F-measure are calculated as follows:

$$\begin{aligned}
Recall &= \frac{\text{\# of correctly generated paraphrase candidates}}{\text{\# of manually described paraphrase candidates}}, \\
Precision &= \frac{\text{\# of correctly generated paraphrase candidates}}{\text{\# of generated paraphrase candidates}}, \\
F-measure &= \frac{1}{\alpha(1/R) + (1-\alpha)(1/P)}.
\end{aligned}$$

Table 5.2. Performance of the proposed model.

Setting	w/o error detection	w/ error detection
# of generated candidates	822	741
# of correct candidates	624	609
Precision	0.759	0.822
Recall	0.878	0.857
F-measure ($\alpha = 0.5$)	0.814	0.839

5.5.3 Error analysis

To clarify the cause of the erroneous paraphrases, we manually analyzed 198 erroneous paraphrase candidates. Table 5.3 lists the error sources. In this section, we denote some typical errors one by one.

LCS transformation algorithm

The experiment came close to confirming that the right-first matching algorithm in our paraphrasing model operates correctly. Unfortunately, the matching rules produced some erroneous paraphrases in LCS transformation.

(i) Errors in Predicate Matching: To paraphrase sentence (81s) below, "CONTROL" in LCS_{V1} must be matched with "CONTROL" in LCS_{N0}, and x to x'. However, our model first matched "CONTROL" in LCS_{V1} with "MOVE FROM TO" in LCS_{N0}. Thus, x was incorrectly matched with z' while x' remained empty. The desired form of LCS_{N1} is shown in example (82).

(81) s. *kacho-ga* *buka-ni* ***shiji-o*** ***dasu***.
section chief-NOM subordinate-DAT order-ACC to issue-PRES
The section chief issues orders to his subordinates.

LCS_{V1} [[*chief*]x CONTROL [BECOME [[*order*]y BE AT [*subordinate*]z]]]

LCS_{N0} [x' CONTROL [y' MOVE FROM z' TO [FILLED]]]

LCS_{N1}*[x' CONTROL [[*subordinate*]y' MOVE FROM [*chief*]z' TO [FILLED]]]

(82) LCS_{N1} [[*chief*]x' CONTROL [y' MOVE FROM [*subordinate*] TO [FILLED]]]

This error was caused because "CONTROL" in LCS_{V1} was inappropriately matched with "MOVE FROM TO" in LCS_{N0}. Although we regard some predicates as being in the same classes as those described in Section 5.4.2, these need to be reconsidered with care. In particular "MOVE FROM TO" needs further investigation because it causes many errors whenever it has the "FILLED" argument.

Table 5.3. Error distribution.

Setting	w/o error detection		w/ error detection	
# of erroneous candidates	198	(100.0%)	132	(100.0%)
Definition of LCS	30	(15.2%)	19	(14.4%)
LCS for light-verb	24		14	
LCS for nominalized verb	6		5	
Paraphrasing model	61	(30.8%)	38	(28.8%)
LCS transformation algorithm	59		36	
Treatment of "*suru* (to do)"	2		2	
Ambiguity	107	(54.0%)	75	(56.8%)
Ambiguous thematic role of dative	78		47	
Recognition of LVC	24		24	
Selection of transitive/intransitive	5		4	

(ii) Errors in Argument Matching: Even if all the predicates are matched accurately, there would still be a chance of errors being caused by incorrect argument matching. With the present algorithm, z can be matched with y' if and only if z' contains "FILLED." In the case of example (83), however, z has to be matched with y', even though z' is empty. The desired form of LCS_{N1} is shown in example (84).

(83) s. *jikan-ni* ***seigen-ga*** ***aru***.
time-DAT limitation-NOM to exist-PRES
There is a time limitation.

LCS_{V1} [[*limitation*]y <u>BE AT</u> [*time*]z]

LCS_{N0} [x' CONTROL [BECOME [y' <u>BE AT</u> z']]]

LCS_{N1}*[x' CONTROL [BECOME [y' BE AT [*time*]z']]]

(84) LCS_{N1} [x' CONTROL [BECOME [[*time*]y' BE AT z']]]

Ambiguous thematic role of dative cases

In contrast to dative cases in English, those in Japanese have ambiguity. That is, they can be either complements to the verbs or adjuncts[5]. However, since LCS is not capable of determining whether a case is a complement or an adjunct, z is occasionally inappropriately filled with an adjunctive element. For example, "*medo-ni*" in sentence (85s) should not fill z, because it acts as an adverb, even though it consists of a noun "*medo* (prospect)" and a case particle "*ni*" for the dative. We found that 78 erroneous candidates constitute this most dominant type of errors.

(85) s. *kin'you-o* *medo-ni* ***sagyo-o*** ***susumeru***.
Friday-NOM by-DAT work-ACC to carry on-PRES
I plan to finish the work by Friday.

[5] Muraki (1991) classifies dative cases into 11 thematic roles that can be regarded as complements. In contrast, there is no reliable typology of dative cases that act as adjuncts.

LCS_{V0} [x CONTORL [BECOME [y BE AT z]]]
LCS_{V1}*[x CONTROL [BECOME [[*work*]y BE AT [*by*]z]]]

The ambiguity of dative cases in Japanese has been discussed in the literature of corpus linguistics and several work on NLP (Muraki, 1991; Kawahara and Kurohashi, 2001; Kawahara and Kurohashi, 2002). To date, however, a practical compliment/adjunct classifier has not been established. We plan to address this topic in our future research. Preliminary investigation revealed that only certain groups of nouns, such as a temporal expression "*gogo-sanji-ni* (at/to 3 p.m.)," can be both compliments and adjuncts according to the governing verb. We therefore expect to be able to determine whether a word in dative cases acts as a complement without combining it with the verb.

Recognition of LVC

In our model, we assume that a triplet $\langle v, c, n\rangle$ consisting of a nominalized verb n and a light-verb tuple $\langle v, c\rangle$ from our vocabulary lists always act as an LVC. However, not only the triplet itself but also its context sometimes affects whether the given triplet can be paraphrased. For example, we regard "*imi-ga aru*" as an LVC, because both the nominalized verb "*imi*" and the tuple $\langle$"*ga*," "*aru*"$\rangle$ appear in the vocabulary lists. However, the $\langle v, c, n\rangle$ in sentence (86s) does not act as an LVC, while the same triplet in sentence (87s) does.

(86) s. *sanka-suru-koto-**ni*** ***imi-ga*** ***aru***.
to participate-DAT meaning-NOM to exist-PRES
There is the meaning in participating.

t. $^{\neq}$*sanka-suru-koto-**o*** ***imi-suru***.
to participate-ACC to mean-PRES
$^{\neq}$It means to participate in it.

(87) s. *"kennel"-**ni-wa*** *inugoya-**toiu*** ***imi-ga*** ***aru***.
"kennel"-DAT-TOP doghouse-of meaning-NOM to exist-PRES
"kennel" has the meaning of doghouse.

t. *"kennel"-**wa*** *inugoya-**o*** ***imi-suru***.
"kennel"-TOP doghouse-ACC to mean-PRES
"kennel" means doghouse.

The above difference is caused by the polysemy of the nominalized verb "*imi*" that denotes "worth" in the context of sentence (86s), but "meaning" in sentence (87s). This error seems to be avoided by determining the word sense of "*imi*" using contextual clues before inputting it to our paraphrasing model. However, introducing such a mechanism complicates our model, since in fact only a limited number of nominalized verbs are polysemous. We therefore plan to consider a reasonable way that we list them as a trigger for making a decision as to whether we need to take the context into account. Namely, given a triplet $\langle v, c, n\rangle$, we would be able to identify whether it is (a) a main verb phrase, (b) a delicate case in terms of the dependence of its context, or (c) an LVC.

5.6 Summary

In this chapter, we explore what sorts of knowledge explain the systemicity underlying several classes of paraphrases, particularly focusing on what sorts of lexico-semantic properties can be represented by the theory of LCS. We first investigated an existing LCS typology of Japanese verbs, and make a suggestion on several refinements of the typology. Through the investigation, we justified that the theory of LCS provides the information to express (i) the relationship between syntactic cases and semantic-level arguments, (ii) the semantic structure which also determines the potential of several types of lexical alternations and derivations, and (iii) the possibilities of syntactic alternations and transformations. On the basis of this recognition, we then proposed an LCS-based paraphrase generation model for LVCs. Through an experiment, we showed that our model generated paraphrases of LVCs accurately in selecting the voice and reassigning the cases, as well as gaining advantages on the treatment of sibling cases over conventional approaches to this class of paraphrasing.

In experiments, our model generated paraphrases for most input sentences. However, the transformation algorithm introduced several types of errors. Although errors can be detected by applying language models, the model still has room for further improvement. To make our model more accurate, in particular, we plan to have more discussion on the semantic analysis and the LCS transformation algorithm.

- On the semantic analysis, the polysemy of nominalized verb and "*ni* (DAT)" case formed a major problem. As the shallow level analyzers, such as morphological analyzer, part-of-speech tagger, and parser, have benefited various NLP tasks, establishing the technology of LCS analysis as indispensable for handling semantics with LCS.

- We followed the case assignment rule proposed by (Takeuchi *et al*., 2002). In the English LCS dictionary (Dorr, 1997), on the other hand, LCS itself contains the information for linking syntactic elements and arguments. To determine the fine-grained description level, we need a discussion with extensive investigation into lexicology. Equally, we need a criterion for assigning LCS to new verbs. Therefore, we must theoretically justify our intuitive refinements of LCS typology, as described in (Takeuchi *et al*., 2002).

- LCS has been discussed as a means of representing verbal semantics and used for explaining the difference between transitive/intransitive verbs, and the construction of compounds. Therefore, one of the next goals is to assess the applicability of LCS on practical use for explaining regularity of corresponding classes of paraphrases: transitivity alterations, (e.g., (29)), and compound noun decomposition (e.g., (23)). On the other hand, paraphrasing of LVCs requires semantic-level transformations. Hence one may suspect that the LCS representations of the source sentence, i.e., LCS_{V1}, occasionally seem to be distorted, while those of the target sentence, i.e., LCS_{N1}, seem to be straightforward. We thus need further intensive discussion on the semantic-level transformation to design paraphrasing models for various paraphrase classes.

CHAPTER 6 Conclusion

In this thesis, we addressed two issues of lexical and structural paraphrases: handling transfer errors and a lexical-semantics-based account for paraphrase generation. This chapter concludes the thesis with the following sections. Section 6.1 outlines the major contributions of the thesis. Section 6.2 lists ideas for possible future work following on the work in this thesis.

6.1 Summary of contributions

Transfer error analysis

Various case studies have so far been done on paraphrasing. However, a comprehensive investigation has not been made into transfer errors across different paraphrase classes. We therefore investigated transfer errors by reexamining previously proposed knowledge and newly developed ones (Chapter 3).

First, we collected examples of transfer errors with their revised forms, and then built a typology by decomposing these examples into primitive revision examples and distilling them into morpho-syntactic patterns. The resultant error typology exhibited a wide range of variety from morphological errors to semantic and discourse-related ones.

Second, our investigation into transfer error distributions revealed that a large part of actual transfer errors fell into either the error type of our typology, irrespective of the paraphrase classes. This suggests that if one creates a revision component specialized for a particular error type, it will compensate for errors resulting from the underspecified transfer rules. Among various types of transfer errors, the shallow levels of transfer errors occurred relatively frequently, such as inappropriate conjugation forms of verbs, and incorrect functional word connections. Since these are well-known problems in the literature of morpheme generation, we expect that making a solution for this type of errors is quite feasible.

Given the observations, we concluded that the detection of incorrect case assignments was one of the most urgent issues that should have preference as a research topic.

Error detection model

Our focus was then turned to the most imperious type of error occurring in paraphrase generation, viz. incorrect case assignments (Chapter 4).

We first assessed a conventional language modeling method which employed a large collection of positive examples obtained from corpus, and then argued how to use negative examples effectively. Since the negative examples were not available in nature, we manually collected a small number of negative examples, and trained another model independent of the language model (positive model). Our error detection model combined these two models, positive and negative models, and significantly outperformed the baseline model which was trained only on positive examples. We also proposed a selective sampling scheme to reduce the cost of collecting negative examples. The scheme also combined the two models above in order to determine example which is most likely to be incorrect. Experimental results confirmed its effectiveness for the error detection task.

Through the empirical experiments, we proved the feasibility of our over-generation plus filtering approach.

Lexical-semantics-based account for paraphrases

Although we were able to show the potential of our transfer-and-revise approach to paraphrase generation, it does not contribute to the sufficiency of paraphrase generation. Our approach to this issue is the lexical-semantics-based account for paraphrase generation (Chapter 5).

We aimed at capturing the systemicity underlying several classes of paraphrases, such as verb alternation and compound noun decomposition, and assessed the Lexical Conceptual Structure (LCS), which represented verbs as semantic structures together with the relationships between their arguments and syntactic cases. Relying on this framework, we developed a paraphrase generation model consisting of a handful of LCS transformation rules, particularly focusing on paraphrasing of Japanese light-verb constructions (LVCs). Through an experiment, we showed that our model generated paraphrases of LVCs accurately, gaining advantages over conventional approaches to this class of paraphrasing.

Our approach is promising for realizing paraphrases, since it relies on a linguistic theory.

Paraphrase collection

We collected paraphrases discussed on either analyses from the linguistic point of view or previous research from the viewpoint of natural language processing (Chapter 2). The collection, the byproduct of the thesis, does not cover a sufficiently wide range of paraphrases and further collection is needed. But it still shows more extensive examples than the existent collections. We therefore believe that it would be useful for further research. Potential work includes the following: case studies focusing on the particular paraphrase class, and knowledge discovery and its assessment across the different paraphrase classes.

Transfer rules, lexical knowledge, and paraphrase examples

Throughout the work in this thesis, a large amount of transfer rules and lexical knowledge was constructed[1]. Furthermore, throughout the experiments, we manually constructed thousands of paraphrase examples for evaluation. These are also contributions of this thesis.

A part of the transfer rules were handcrafted, while the remaining part were automatically acquired and then manually refined. Although they may still contain errors or underspecified applicability conditions, we can compensate for the transfer errors caused by them on our transfer-and-revise model.

[1] We have been releasing the knowledge-base at http://cl.naist.jp/kura/KuraData/.

Several types of lexical knowledge are automatically acquired from existing language resources followed by manual noise filtering. The discovered lexical knowledge is merely the tip of the iceberg, yet they are useful for the current stage of paraphrase generation.

Paraphrase examples were mainly generated automatically using KURA (Takahashi *et al.*, 2001), and then were evaluated manually. Since the evaluation included manual revision and error type annotation, the resultant paraphrases can be used as a gold-standard. In addition, negative examples which cannot be differentiated from positive ones by syntactic patterns are also useful, since such examples are usually not available. They offer a chance for discovering further information required for paraphrase generation.

6.2 Future work

As described in Chapter 1, handling paraphrases is promised to benefit plenty of natural language applications. Likewise, paraphrases are deeply associated with human communication. The work in this thesis explored the methods for addressing underlying problems; however, several issues are still unexplored. We would like to conclude the thesis with our research plan for the future.

Exploration of LCS

We think that research on paraphrases gives us a perspective for semantic-level processing of natural language. Therefore, foremost in our future work is a more extensive investigation into LCS. We are planning to address the following three issues.

First, we observed that our LCS-transformation model is still immature. Although our basic standing is to generate paraphrase candidate on the assumption that incorrect ones can be thrown away automatically in the post-generation stage, we should make more effort to improve the candidate generation technologies[2]. For example, the treatment of attachment consists of four decision rules and a default rule. To refine this, we need to examine linguistically motivated applicability conditions of each decision rule.

Second, the LCS dictionary also needs to be enhanced. As described in Chapter 5, then nonnegligible ratio (15%) of errors occurred by the suspicious definition of LCS, even though we achieved a couple of refinements. Therefore, we plan to reexamine the LCS typology as well as giving more numbers of verbs consideration.

Third, we have to establish an LCS analyzer. The present case assignment rule shown in Section 5.4.1 does not take into account the arbitrariness of cases as we saw in example (85). In order to employ LCS as a standard form of representing sentences in paraphrase generation, an LCS analyzer is required which performs as sufficiently accurately as conventional analyzers such as morphological analyzers, named entity recognizers, and parsers.

[2]Similar claims have also made in (Yamamoto, 2001).

Constructing a large-scale paraphrase corpus for evaluation

Paraphrase generation systems have so far been evaluated by means of manual judgement on automatically generated paraphrases: an assessor classifies each paraphrase into correct or incorrect. Such an evaluation method, however, was time-consuming, and thus leads to lagging implementation-evaluation cycles. Therefore, a pressing issue is to establish an evaluation scheme.

The analogous use of statistical measures might be a reasonable option. With the introduction of BLEU (Papineni *et al*., 2002) and NIST (NIST, 2002) for machine translation evaluation, the advantages of doing automatic evaluation have become increasingly appreciated. For example, in the community of automatic summarization, a metric ROUGE (Lin and Hovy, 2003) has been proposed and is currently widely used. However, we have to be prudent, because these metrics refer to only the local context, i.e., n-gram statistics, which do not necessarily work well in handling Japanese (see the discussion in Section 4.2.2). Furthermore, paraphrase generation is the task of restructuring the given sentences, and thus most lexical items and their ordering are preserved. Hence, while for the most part it is needless to say whether such expressions are correct, the affected parts should be strictly evaluated. Here the claim becomes just the same as the use of language modeling in our transfer-and-revise model.

Again, we should make more efforts to improve the candidate generation technologies. Then what is and isn't carried out by the given system, we need to construct a large-scale paraphrase corpus, which enables measuring recall and precision that are well-known in various research fields. However, an issue arises: how to reduce the human labor required?

On the construction of a paraphrase corpus for this purpose, it seems to be impossible to take into account all sorts of paraphrase phenomena at the same time. The reasons are as follows. First, a paraphrase for a given expression is also an natural language expression, while output of traditional NLP tasks (e.g., part-of-speech tagging, named entity recognition, and dependency parsing) can be represented by closed sets of tags. Second, one can allow the plural paraphrases for a given expression. Consequently, nobody can say whether the collected paraphrases are *exhaustive*; nevertheless we need to exhaustively collect all possible (acceptable) answers for given dataset regardless of task. Moreover, collecting sufficiently *diverse* phenomena for evaluation is also difficult because of the complexity.

On the basis of these recognitions, we are planning to construct a paraphrase corpus for each paraphrase class described in Chapter 2. Once we limit the phenomena to a particular paraphrase class, we will be able to avoid the two problems above, namely, lack of exhaustiveness and those of diversity.

Complemental use of transfer knowledge

For the complemental use of automatically acquired transfer knowledge and those developed manually, we plan to develop a knowledge decomposer. As we mentioned in Section 2.4, by handling well-defined syntactic transformation patterns, we will have the opportunity to decompose automatically acquired knowledge. Such work contributes to discovering further paraphrase classes, as well as to reduce the cost of maintaining knowledge.

Bibliography

Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. 2004. Applying Support Vector Machines to imbalanced datasets. In *Proceedings of the 2004 European Conference on Machine Learning* (*ECML*), pp. 39–50.

Jeffrey Allen and Christopher Hogan. 2000. Toward the development of a postediting module for raw machine translation output: a controlled language perspective. In *Proceedings of the 3rd International Workshop on Controlled Language Applications* (*CLAW*), pp. 62–71.

Peter G. Anick and Suresh Tipirneni. 1999. The paraphrase search assistant: terminological feedback for iterative information seeking. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR*) *Workshop on Customized Information Delivery*, pp. 153–159.

The Association for Natural Language Processing (Ed.). 2001. *Workshop on Automatic Paraphrasing*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics* (*COLING-ACL*), pp. 86–90.

Timothy Baldwin, Francis Bond, and Kentaro Ogura. 2001. Dictionary-driven analysis of Japanese verbal alternations. In *Proceedings of the 7th Annual Meeting of the Association for Natural Language Processing*, pp. 281–284.

Srinivas Bangalore and Owen Rambow. 2000. Corpus-based lexical choice in natural language generation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 464–471.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 550–557.

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 50–57.

Regina Barzilay. 2003. *Information fusion for multidocument summarization: paraphrasing and generation*. Ph.D. thesis, Graduate School of Arts and Sciences, Columbia University.

Stephen Beale, Sergei Nirenburg, Evelyne Viegas, and Leo Wanner. 1998. De-constraining text generation. In *Proceedings of the 9th International Workshop on Natural Language Generation* (*INLG*), pp. 48–57.

Francis Bond and Sanae Fujita. 2003. Evaluation of a method of creating new valency entries. In *Machine Translation Summit IX* (*MT Summit*), pp. 16–23.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of machine translation: parameter estimation. *Computational Linguistics*, 19(2):263–311.

Caroline Brun and Caroline Hagège. 2003. Normalization and paraphrasing using symbolic methods. In *Proceedings of the 2nd International Workshop on Paraphrasing: Paraphrase Acquisition and Applications* (*IWP*), pp. 41–48.

Yvonne Canning and John Tait. 1999. Syntactic simplification of newspaper text for aphasic readers. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR*) *Workshop on Customized Information Delivery*, pp. 6–11.

Michael Carl and Andy Way (Eds.). 2003. *Recent advances in example-based machine translation*. Kluwer Academic Publishers.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics* (*EACL*), pp. 269–270.

William Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics* (*COLING*), pp. 350–356.

Bonnie Jean Dorr. 1993. *Machine translation: a view from the lexicon*. The MIT Press.

Bonnie Jean Dorr. 1997. Large-scale dictionary construction for foreign language tutoring and interlingual machine translation. *Machine Translation*, 12(4):271–322. http://clipdemos.umiacs.umd.edu/englcslex/.

Bonnie Jean Dorr, Rebecca Green, Lori Levin, Owen Rambow, David Farwell, Nizar Habash, Stephen Helmreich, Eduard Hovy, Keith J. Miller, Teruko Mitamura, Florence Reeder, and Advaith Siddharthan. 2004. Semantic annotation and lexico-syntactic paraphrase. In *Proceedings of the 4th International Conference on Language Resources and Evaluation* (*LREC*) *Workshop on Building Lexical Resources from Semantically Annotated Corpora*, pp. 47–52.

Mark Dras. 1999. *Tree adjoining grammar and the reluctant paraphrasing of text*. Ph.D. thesis, Division of Information and Communication Science, Macquarie University.

Florence Duclaye, François Yvon, and Olivier Collin. 2003. Learning paraphrases to improve a question-answering system. In *Proceedings of the 10th Conference of the European*

Chapter of the Association for Computational Linguistics (*EACL*) *Workshop on Natural Language Processing for Question-Answering*, pp. 35–41.

Philip Edmonds. 1999. *Semantic representations of near-synonyms for automatic lexical choice*. Ph.D. thesis, Graduate Department of Computer Science, University of Toronto.

EDR. 1995. *EDR electronic dictionary version 1.5 technical guide*. Japan Electronic Dictionary Research Institute. (in Japanese).

Andrew Finch, Taro Watanabe, Yasuhiro Akiba, and Eiichiro Sumita. 2004. Paraphrasing as machine translation. *Journal of Natural Language Processing*, 11(5):87–111.

Atsushi Fujita, Kentaro Inui, and Hiroko Inui. 2000. An environment for constructing nominal-paraphrase corpora. In *Technical Report of Institute of the Electronics, Information and Communication Engineers, TL-2000-32*, pp. 53–60. (in Japanese).

Atsushi Fujita and Kentaro Inui. 2001. Paraphrasing of common nouns to their synonyms using definition statements. In *Proceedings of the 7th Annual Meeting of the Association for Natural Language Processing*, pp. 331–334. (in Japanese).

Sanae Fujita and Francis Bond. 2002a. A method of adding new entries to a valency dictionary by exploiting existing lexical resources. In *Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation* (*TMI*), pp. 42–52.

Sanae Fujita and Francis Bond. 2002b. Extending the coverage of a valency dictionary. In *Proceedings of the 19th International Conference on Computational Linguistics* (*COLING*) *Workshop on Machine Translation in Asia*, pp. 67–73.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2004. Fast decoding and optimal decoding for machine translation. *Artificial Intelligence*, 154(1-2):127–143.

Daniel Gildea and Martha Palmer. 2002. The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 239–246. http://www.cis.upenn.edu/˜ace/.

Daniel Gildea. 2002. Probabilistic models of verb-argument structure. In *Proceedings of the 19th International Conference on Computational Linguistics* (*COLING*), pp. 308–314.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Nizar Habash and Bonnie Jean Dorr. 2002. Handling translation divergences: combining statistical and symbolic techniques in generation-heavy machine translation. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas*, pp. 84–93.

Zellig Harris. 1981. Co-occurrence and transformation in linguistic structure. In Henry Hiz (Ed.), *Papers on Syntax*, pp. 143–210. D. Reidel Publishing Company.

Shin'ichi Hatakeyama, Hiroshi Sakamoto, Tsuneaki Kato, and Takane Ito. 2005. How to determine Japanese verb classes — a case study on transitive verbs—. In *Information Processing Society of Japan SIG Notes, NL-165-1*, pp. 1–8. (in Japanese).

Yoshihiko Hayashi. 1992. A three-level revision model for improving Japanese bad-styled expressions. In *Proceedings of the 14th International Conference on Computational Linguistics* (*COLING*), pp. 665–671.

Ulf Hermjakob, Abdessamad Echibahi, and Daniel Marcu. 2002. Natural language based reformulation resource and web exploitation for question answering. In *Proceedings of the 11th Text Retrieval Conference* (*TREC 2002*).

Masako Himeno. 1999. *Structures and semantic usages of compound verbs*. Hitsuji Syobo. (in Japanese).

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR*), pp. 50–57.

Xiaorong Huang and Armin Fiedler. 1996. Paraphrasing and aggregating argumentative text using text structure. In *Proceedings of the 8th International Workshop on Natural Language Generation* (*INLG*), pp. 21–30.

Ryu Iida, Yasuhiro Tokunaga, Kentaro Inui, and Junji Eto. 2001. Exploration of clause-structural and function-expressional paraphrasing using KURA. In *Proceedings of the 63th Annual Meeting of Information Processing Society of Japan*, pp. 5–6. (in Japanese).

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi (Eds.). 1997. *Nihongo Goi Taikei – a Japanese lexicon*. Iwanami Shoten. (in Japanese), http://www.kecl.ntt.co.jp/mtg/resources/GoiTaikei/.

Diana Zaiu Inkpen. 2003. *Building a lexical knowledge-base of near-synonym differences*. Ph.D. thesis, Graduate Department of Computer Science, University of Toronto.

Diana Zaiu Inkpen, Ol'ga Feiguina, and Graeme Hirst. 2004. Generating more-positive or more-negative text. In *Proceedings of the Workshop on Attitude and Affect in Text, AAAI 2004 Spring Symposium*, pp. 83–89.

Kazuko Inoue. 1976. *Transformational grammar and Japanese: syntax*, volume 1. Taishukan Shoten.

Kentaro Inui and Masaru Nogami. 2001. A paraphrase-based exploration of cohesiveness criteria. In *Proceedings of the 8th European Workshop on Natural Language Generation* (*EWNLG*), pp. 101–110.

Kentaro Inui and Satomi Yamamoto. 2001. Corpus-based acquisition of sentence readability ranking models for deaf people. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium* (*NLPRS*), pp. 159–166.

Kentaro Inui and Ulf Hermjekob (Eds.). 2003. *The 2nd International Workshop on Paraphrasing: Paraphrase Acquisition and Applications* (*IWP*). ACL-2003 Workshop.

Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: a project note. In *Proceedings of the 2nd Interna-*

tional Workshop on Paraphrasing: Paraphrase Acquisition and Applications (*IWP*), pp. 9–16.

Lidija Iordanskaja, Richard Kittredge, and Alain Polguère. 1991. Lexical selection and paraphrase in a meaning-text generation model. In Cécile L. Paris, William R. Swartout, and William C. Mann (Eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pp. 293–312. Kluwer Academic Publishers.

Takane Ito (Ed.). 2002. *Grammatical theory: lexicon and syntax*. University of Tokyo Publisher. (in Japanese).

Ray Jackendoff. 1990. *Semantic structures*. The MIT Press.

Christian Jacquemin, Judith L. Klavans, and Evelyne Tzoukermann. 1997. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics* (*ACL-EACL*), pp. 24–31.

Hiroshi Kabaya, Yoshikazu Kawaguchi, and Megumi Sakamoto. 1998. *Polite expressions*. Taishukan Shoten. (in Japanese).

Taro Kageyama. 1996. *Verb semantics*. Kurosio Publishers. (in Japanese).

Taro Kageyama (Ed.). 2001. *Semantics and syntax of verb: comparable study between Japanese and English*. Taishukan Shoten. (in Japanese).

Taro Kageyama. 2002. Unaccusative transitive verbs: the interface between semantics and syntax. In Takane Ito (Ed.), *Grammatical Theory: Lexicon and Syntax*, pp. 119–145. University of Tokyo Publisher. (in Japanese).

Nobuhiro Kaji, Sadao Kurohashi, and Satoshi Sato. 2001. Paraphrase into plain sentence based on machine readable dictionary. In *Information Processing Society of Japan SIG Notes, NL-144-23*, pp. 167–174. (in Japanese).

Nobuhiro Kaji, Daisuke Kawahara, Sadao Kurohashi, and Satoshi Sato. 2002. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 215–222.

Nobuhiro Kaji, Masashi Okamoto, and Sadao Kurohashi. 2004. Paraphrasing predicates from written language to spoken language using the Web. In *Proceedings of the 2004 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (*HLT-NAACL*).

Nobuhiro Kaji and Sadao Kurohashi. 2004. Recognition and paraphrasing of periphrastic and overlapping verb phrases. In *Proceedings of the 4th International Conference on Language Resources and Evaluation* (*LREC*) *Workshop on Methodologies and Evaluation of Multiword Units in Real-world Application*.

Hiroshi Kanayama. 2003. Paraphrasing rules for automatic evaluation of translation into Japanese. In *Proceedings of the 2nd International Workshop on Paraphrasing: Paraphrase Acquisition and Applications* (*IWP*), pp. 88–93.

Akira Kataoka, Shigeru Masuyama, and Kazuhide Yamamoto. 2000. Paraphrasing a Japanese verbal noun phrase into an expression “*N1 no N2*”. *Journal of Natural Language Processing*, 7(4):79–98. (in Japanese).

Daisuke Kawahara and Sadao Kurohashi. 2001. Japanese case frame construction by coupling the verb and its closest case component. In *Proceedings of the Human Language Technology Conference* (*HLT*), pp. 204–210.

Daisuke Kawahara and Sadao Kurohashi. 2002. Fertilization of case frame dictionary for robust Japanese case analysis. In *Proceedings of the 19th International Conference on Computational Linguistics* (*COLING*), pp. 425–431.

Frank Keller, Maria Lapata, and Olga Ourioupina. 2003. Using the Web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3):459–484.

Yeun-Bae Kim and Terumasa Ehara. 1994. An automatic sentence breaking and subject supplement method for J/E machine translation. *IPSJ Journal*, 35(6):1018–1028. (in Japanese).

Yumiko Kinjo, Kunio Aono, Keishi Yasuda, Toshiyuki Takezawa, and Genichiro Kikui. 2003. Collection of Japanese paraphrases of basic expressions on travel conversation. In *Proceedings of the 9th Annual Meeting of the Association for Natural Language Processing*, pp. 101–104.

Satoshi Kinsui, Mayumi Kudo, and Yoshiko Numata. 2000. *Tense, aspect, negation, and emphasis*. Iwanami Shoten. (in Japanese).

Hideki Kishimoto. 2001. Painting construction. In Taro Kageyama (Ed.), *Comparable study of Japanese and English: Semantics and Syntax of Verb*, pp. 100–126. Taishukan Shoten.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *Proceedings of the 12th National Conference on Artificial Intelligence* (*AAAI*), pp. 779–784.

Kevin Knight and Vasileios Hatzivassiloglou. 1995. Two-level, many-paths generation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 252–260.

Keiko Kondo and Manabu Okumura. 1997. Summarization with dictionary-based paraphrasing. In *Proceedings of the 4th Natural Language Processing Pacific Rim Symposium* (*NLPRS*), pp. 649–652.

Keiko Kondo, Satoshi Sato, and Manabu Okumura. 1999. Paraphrasing of “sahen-noun + suru”. *IPSJ Journal*, 40(11):4064–4074. (in Japanese).

Keiko Kondo, Satoshi Sato, and Manabu Okumura. 2001. Paraphrasing by case alternation. *IPSJ Journal*, 42(3):465–477. (in Japanese).

Tadahisa Kondo and Shigeaki Amano. 2000. Lexical properties of Japanese, “Nihongo-no Goitokusei”: significance and problems. In *Technical Report of Institute of the Electronics, Information and Communication Engineers, TL-2000-14*, pp. 1–8. (in Japanese), http://www.kecl.ntt.co.jp/icl/mtg/goitokusei/.

Shin'ya Kouda, Atsushi Fujita, and Kentaro Inui. 2001. Issues in sentence-dividing paraphrasing: a empirical study. In *Proceedings of the 15th Annual Conference of Japanese Society for Artificial Intelligence*. (in Japanese).

Raymond Kozlowski, Kathleen F. MaCoy, and K. Vijay-Shanker. 2003. Generation of single-sentence paraphrases from predicate/argument structure using lexico-grammatical resources. In *Proceedings of the 2nd International Workshop on Paraphrasing: Paraphrase Acquisition and Applications* (*IWP*), pp. 1–8.

Sadao Kurohashi and Makoto Nagao. 1998. Building a Japanese parsed corpus while improving the parsing system. In *Proceedings of the 1st International Conference on Language Resources and Evaluation* (*LREC*), pp. 719–724. http://www.kc.t.u-tokyo.ac.jp/nl-resource/corpus.html.

Sadao Kurohashi and Yasuyuki Sakai. 1999. Semantic analysis of Japanese noun phrases: a new approach to dictionary-based understanding. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 481–488.

Kazuya Kurokawa. 2003. Classification of correct and incorrect usages of functional expressions in Japanese and its application to learning support. Master thesis, Department of Information and Computer Sciences, Toyohashi University of Technology. (in Japanese).

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics* (*COLING-ACL*), pp. 704–710.

Maria Lapata. 2001. A corpus-based account of regular polysemy: the case of context-sensitive adjectives. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics* (*NAACL*), pp. 63–70.

Maria Lapata, Frank Keller, and Scott McDonald. 2001. Evaluating smoothing algorithms against plausibility judgements. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 346–353.

Benoit Lavoie, Richard Kittredge, Tanya Korelsky, and Owen Rambow. 2000. A framework for MT and multilingual NLG systems based on uniform lexico-structural processing. In *Proceedings of the 6th Applied Natural Language Processing Conference and the 1st Meeting of the North American Chapter of the Association for Computational Linguistics* (*ANLP-NAACL*), pp. 60–67.

Lillian Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 25–32.

Lillian Lee. 2001. On the effectiveness of the skew divergence for statistical language analysis. In *Proceedings of the 8th International Workshop on Artificial Intelligence and Statistics*, pp. 65–72.

Beth Levin. 1993. *English verb classes and alternations: a preliminary investigation*. Chicago Press.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (*HLT-NAACL*), pp. 150–157.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Inderjeet Mani, Barbara Gates, and Eric Bloedorn. 1999. Improving summaries by revising them. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 558–565.

Suguru Matsuyoshi, Satoshi Sato, and Takehito Utsuro. 2004. Paraphrasing a functional word "nara" for machine translation. In *Information Processing Society of Japan SIG Notes, NL-159-28*, pp. 201–208. (in Japanese).

Kathleen R. McKeown, Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: progress and prospects. In *Proceedings of the 16th National Conference on Artificial Intelligence and 11th Conference on Innovative Application s of Artificial Intelligence* (*AAAI-IAAI*), pp. 453–460.

I. Dan Melamed. 2001. *Empirical methods for exploiting parallel texts*. The MIT Press.

Igor Mel'čuk and Alain Polguère. 1987. A formal lexicon in meaning-text theory (or how to do lexica with words). *Computational Linguistics*, 13(3-4):261–275.

Maria Meteer and Varda Shaked. 1988. Strategies for effective paraphrasing. In *Proceedings of the 12th International Conference on Computational Linguistics* (*COLING*), pp. 431–436.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244. http://www.cogsci.princeton.edu/˜wn/.

Teruko Mitamura and Eric Nyberg. 2001. Automatic rewriting for controlled language translation. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium* (*NLPRS*) *Workshop on Automatic Paraphrasing: Theories and Applications*, pp. 1–12.

Tadao Miyamoto. 1999. *The light verb construction in Japanese: the role of the verbal noun*. John Benjamins Publishing Company.

Yoshiyuki Morita and Masae Matsuki. 1989. *Expressions and patterns in Japanese — meanings and usages of complex expressions based on example*. ALC. (in Japanese).

Shinjiro Muraki. 1991. *Various aspects of Japanese verbs*. Hitsuji Syobo. (in Japanese).

Hidetsugu Nanba and Manabu Okumura. 2000. Producing more readable extracts by revising them. In *Proceedings of the 18th International Conference on Computational Linguistics* (*COLING*), pp. 1071–1075.

NIST (Ed.). 2002. *Automatic evaluation of machine translation quality using n-gram co-occurrence statistics*. http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf.

Eric Nyberg and Teruko Mitamura. 2000. The KANTOO machine translation environment. In *Proceedings of AMTA 2000 Conference*, pp. 192–195.

Shuta Ogawa and Shun Ishizaki. 2004. Interaction between deep cases in conceptual dictionary — a case study on painting sentences. In *Proceedings of the 10th Annual Meeting of the Association for Natural Language Processing*, pp. 572–575. (in Japanese).

Kazuteru Ohashi and Kazuhide Yamamoto. 2004. Paraphrasing "sahen-verb + noun" into compound nouns. In *Proceedings of the 10th Annual Meeting of the Association for Natural Language Processing*, pp. 693–696. (in Japanese).

Kiyonori Ohtake and Kazuhide Yamamoto. 2001. Paraphrasing honorifics. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium* (*NLPRS*) *Workshop on Automatic Paraphrasing: Theories and Applications*, pp. 13–20.

Kiyonori Ohtake and Kazuhide Yamamoto. 2003. Applicability analysis of corpus-derived paraphrases toward example-based paraphrasing. In *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation* (*PACLIC*), pp. 380–391.

Hiroyuki Okamoto, Kengo Sato, and Hiroaki Saito. 2003. Preferential presentation of Japanese near-synonyms using definition statements. In *Proceedings of the 2nd International Workshop on Paraphrasing: Paraphrase Acquisition and Applications* (*IWP*), pp. 17–24.

Masahiro Oku. 1990. Analysis methods for Japanese idiomatic predicates. *IPSJ Journal*, 31(12):1727–1734. (in Japanese).

Susumu Ôno and Masando Hamanishi. 1981. *Kadokawa synonym new dictionary*. Kadokawa Shoten Publishing. (in Japanese).

Manabu Ori and Satoshi Sato. 2002. Composition of Japanese texts. In *Proceedings of the 8th Annual Meeting of the Association for Natural Language Processing*, pp. 367–370. (in Japanese).

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (*HLT-NAACL*), pp. 102–109.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 311–318.

Darren Pearce. 2001. Synonymy in collocation extraction. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics* (*NAACL*) *Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, pp. 41–46.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 183–190.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), pp. 142–149.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 215–222.

Fabio Rinaldi, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Mollá. 2003. Exploiting paraphrases in a question answering system. In *Proceedings of the 2nd International Workshop on Paraphrasing: Paraphrase Acquisition and Applications* (*IWP*), pp. 25–32.

Jacques Robin and Kathleen McKeown. 1996. Empirically designing and evaluating a new revision-based model for summary generation. *Artificial Intelligence*, 85(1-2):135–179.

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via EM-based clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (*ACL*), pp. 104–111.

RWC, 1998. *RWC text database version 2 with Iwanami Kokugo Jiten*. (in Japanese).

Hiroyuki Sakai and Shigeru Masuyama. 2003. Automatic acquisition of noun and its abbreviation correspondences from corpus. In *Proceedings of the 9th Annual Meeting of the Association for Natural Language Processing*, pp. 226–229. (in Japanese).

Satoshi Sato. 1999. Automatic paraphrase of technical papers' titles. *IPSJ Journal*, 40(7):2937–2945. (in Japanese).

Satoshi Sato and Hiroshi Nakagawa (Eds.). 2001. *Workshop on Automatic Paraphrasing: Theories and Applications*. NLPRS-2001 Workshop.

Mitsuo Shimohata and Eiichiro Sumita. 2002a. Automatic paraphrasing based on parallel corpus for normalization. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation* (*LREC*), pp. 453–457.

Mitsuo Shimohata and Eiichiro Sumita. 2002b. Identifying synonymous expressions from a bilingual corpus for example-based machine translation. In *Proceedings of the 19th International Conference on Computational Linguistics* (*COLING*) *Workshop on Machine Translation in Asia*, pp. 20–25.

Mitsuo Shimohata. 2004. *Acquiring paraphrases from corpora and its application to machine translation*. Ph.D. thesis, Graduate School of Information Science, Nara Institute of Science and Technology.

Yusuke Shinyama and Satoshi Sekine. 2003. Paraphrase acquisition for information extraction. In *Proceedings of the 2nd International Workshop on Paraphrasing: Paraphrase Acquisition and Applications* (*IWP*), pp. 65–71.

Tamotsu Shirado, Satoko Marumoto, and Hitoshi Isahara. 2003. Quantitative analyses of misusage of polite expressions. *Mathematical Linguistics*, 24(2):65–80. (in Japanese).

Satoshi Shirai, Satoru Ikehara, and Tsukasa Kawaoka. 1993. Effects of automatic rewriting of source language within a Japanese to English MT system. In *Proceedings of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation* (*TMI*), pp. 226–239.

Satoshi Shirai, Kazuhide Yamamoto, and Francis Bond. 2001. Japanese-English paraphrase corpus. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium* (*NLPRS*) *Workshop on Language Resources in Asia*, pp. 23–30.

Nobuyuki Shiraki and Sadao Kurohashi. 2000. Book retrieval system based on flexible matching between natural language queries and tables of contents. *IPSJ Journal*, 41(4):1162–1170. (in Japanese).

Advaith Siddharthan. 2003. Preserving discourse structure when simplifying text. In *Proceedings of the 9th European Workshop on Natulal Language Generation* (*EWNLG*), pp. 103–110.

Yuriko Sunagawa. 1995. Principles of functions and wordings of Japanese cleft sentences. In Yoshio Nitta (Ed.), *Research on Complex Sentences* (*2nd half*), pp. 353–388. Kurosio Publishers. (in Japanese).

Tetsuro Takahashi, Tomoya Iwakura, Ryu Iida, Atsushi Fujita, and Kentaro Inui. 2001. KURA: a transfer-based lexico-structural paraphrasing engine. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium* (*NLPRS*) *Workshop on Automatic Paraphrasing: Theories and Applications*, pp. 37–46. http://cl.naist.jp/kura/doc/.

Tetsuro Takahashi, Kozo Nawata, Kentaro Inui, and Yuji Matsumoto. 2003. Effects of structural matching and paraphrasing in question answering. *IEICE Transactions on Information and Systems*, E86-D(9):1677–1685.

Zenbun Takahashi and Kazuo Ushijima. 1991. Measuring clarity of computer manuals. *IPSJ Journal*, 32(4):460–469. (in Japanese).

Eiji Takeishi and Yoshihiko Hayashi. 1992. Dividing Japanese complex sentences based on conjunctive expressions analysis. *IPSJ Journal*, 33(5):652–663. (in Japanese).

Koichi Takeuchi, Kyo Kageura, and Teruo Koyama. 2002. An LCS-based approach for analyzing Japanese compound nouns with deverbal heads. In *Proceedings of the 2nd International Workshop on Computational Terminology* (*CompuTerm*), pp. 64–70. http://cl.it.okayama-u.ac.jp/rsc/lcs/.

Koichi Takeuchi. 2004. Developing an LCS dictionary for verb. In *Proceedings of the 10th Annual Meeting of the Association for Natural Language Processing*, pp. 576–579. (in Japanese).

Akira Terada and Takenobu Tokunaga. 2001. Automatic disabbreviation by using context information. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium* (*NLPRS*) *Workshop on Automatic Paraphrasing: Theories and Applications*, pp. 21–28.

Yasuhiro Tokunaga. 2002. Paraphrasing and semantic analysis of negative expressions focusing on emphasized expressions. Graduation thesis of Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology. (in Japanese).

Kentaro Torisawa. 2001. A nearly unsupervised learning method for automatic paraphrasing of Japanese noun phrases. In *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium* (*NLPRS*) *Workshop on Automatic Paraphrasing: Theories and Applications*, pp. 63–72.

Kentaro Torisawa. 2002. An unsupervised learning method for associative relationships between verb phrases. In *Proceedings of the 19th International Conference on Computational Linguistics* (*COLING*), pp. 1009–1015.

Masatoshi Tsuchiya, Satoshi Sato, and Takehito Utsuro. 2004. Automatic paraphrasing rule acquisition from a paraphrase corpus for functional expressions. In *Proceedings of the 10th Annual Meeting of the Association for Natural Language Processing*, pp. 492–495. (in Japanese).

Kiyoko Uchiyama and Shun Ishizaki. 2003. A disambiguation method for Japanese compound verbs. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics* (*ACL*) *Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pp. 81–88.

Wolfgang Wahlster (Ed.). 2000. *Verbmobil: foundations of speech-to-speech translation*. Springer.

Masaya Yamaguchi, Nobuo Inui, Yoshiyuki Kotani, and Hirohiko Nishimura. 1998. Acquisition of automatic pre-edition rules from results of pre-edition. *IPSJ Journal*, 39(1):17–28. (in Japanese).

Kazuhide Yamamoto. 2001. Present state and issues on paraphrasing. In *Proceedings of Workshop on Automatic Paraphrasing of the 7th Annual Meeting of the Association for Natural Language Processing*, pp. 93–96. (in Japanese).

Kazuhide Yamamoto. 2002a. Machine translation by interaction between paraphraser and transfer. In *Proceedings of the 19th International Conference on Computational Linguistics* (*COLING*), pp. 1107–1113.

Kazuhide Yamamoto. 2002b. Acquisition of lexical paraphrases from texts. In *Proceedings of the 2nd International Workshop on Computational Terminology* (*CompuTerm*), pp. 22–28.

Keyword Index