Standardizing Complex Functional Expressions in Japanese Predicates: Applying Theoretically-Based Paraphrasing Rules

Tomoko Izumi[†]   **Kenji Imamura**[†]   **Genichiro Kikui**[†]   **Satoshi Sato**[‡]

[†]NTT Cyber Space Laboratories,
NTT Corporation
{izumi.tomoko, imamura.kenji,
kikui.genichiro}@lab.ntt.co.jp

[‡]Graduate School of Engineering,
Nagoya University
ssato@nuee.nagoya-u.ac.jp

Abstract

In order to accomplish the deep semantic understanding of a language, it is essential to analyze the meaning of predicate phrases, a content word *plus* functional expressions. In agglutinating languages such as Japanese, however, sentential predicates are multi-morpheme expressions and all the functional expressions including those unnecessary to the meaning of the predicate are merged into one phrase. This triggers an increase in surface forms, which is problematic for NLP systems. We solve this by introducing simplified surface forms of predicates that retain only the crucial meaning of the functional expressions. We construct paraphrasing rules based on syntactic and semantic theories in linguistics. The results of experiments show that our system achieves the high accuracy of 77% while reducing the differences in surface forms by 44%, which is quite close to the performance of manually simplified predicates.

1 Introduction

The growing need for text mining systems such as opinion mining and sentiment analysis requires the deep semantic understanding of languages (Inui et al., 2008). In order to accomplish this, one needs to not only focus on the meaning of a single content word such as *buy* but also the meanings conveyed by function words or functional expressions such as *not* and *would like to*. In other words, to extract and analyze a *predicate,* it is critical to consider both the content word *and* the functional expressions (Nasukawa, 2001). For example, the functional expressions *would like to* as in the predicate *"would like to buy"* and *can't* as in *"can't install"* are key expressions in detecting the customer's needs and complaints, providing valuable information to marketing research applications, consumer opinion analysis etc.

Although these functional expressions are important, there have been very few studies that extensively deal with these functional expressions for use in natural language processing (NLP) systems (e.g., Tanabe et al., 2001; Matsuyoshi and Sato, 2006, 2008). This is due to the fact that functional expressions are syntactically complicated and semantically abstract and so are poorly handled by NLP systems.

In agglutinating languages such as Japanese, functional expressions appear in the form of suffixes or auxiliary verbs that follow the content word without any space. This sequence of a content word (*c* for short) plus several of functional expressions (*f* for short) forms a *predicate* in Japanese (COMP for completive aspect marker, NOM for nominalizer, COP for copular verb).

(1) kat -chai -takat -ta -n -da
    buy -COMP -want -PAST -NOM -COP
    c -f_1 -f_2 -f_3 -f_4 -f_5
    "(I) wanted to buy (it)"

The meaning of "want to" is expressed by *-tai* (f_2) and the past tense is expressed by *-ta* (f_3).

The other functional expressions, -*chai*(f_1), -*n*(f_4), and -*da*(f_5), only slightly alter the predicative meaning of "wanted to buy," as there is no direct English translation. Therefore, (1) expresses the same fact as (2).

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

As shown, in Japanese, once one extracts a predicate phrase, the number of differences in surface forms increases drastically regardless of their similarities in meaning. This is because sentential predicates are multi-word or multi-morpheme expressions and there are two different types of functional expressions, one which is crucial for the extraction of predicative meaning and the other, which is almost unnecessary for NLP applications. This increase in surface forms complicates NLP systems including text mining because they are unable to recognize that these seemingly different predicates actually express the same *fact*.

In this study, we introduce paraphrasing rules to transform a predicate with complex functional expressions into a simple predicate. We use the term *standardize* to refer to this procedure. Based on syntactic and semantic theories in linguistics, we construct a simple predicate structure and categorize functional expressions as either necessary or unnecessary. We then paraphrase a predicate into one that only retains the crucial meaning of the functional expression by deleting unnecessary functional expressions while adding necessary ones.

The paper is organized as follows. In Section 2, we provide related work on Japanese functional expressions in NLP systems as well as problems that need to be solved. Section 3 introduces several linguistic theories and our standardizing rules that we constructed based on these theories. Section 4 describes the experiments conducted on our standardization system and the results. Section 5 discusses the results and concludes the paper. Throughout this paper, we use the term *functional expressions* to indicate not only a single function word but also compounds (e.g., *would like to*).

2 Previous Studies and Problems

Shudo et al. (2004) construct abstract semantic rules for functional expressions and use them in order to find whether two different predicates mean the same. Matsuyoshi and Sato (2006, 2008) construct an exhaustive dictionary of functional expressions, which are hierarchically organized, and use it to produce different functional expressions that are semantically equivalent to the original one.

Although these studies provide useful insights and resources for NLP systems, if the intention is to extract the meaning of a *predicate*, we find there are still problems that need to be solved. There are two problems that we focus on.

The first problem is that many functional expressions are unnecessary, i.e., they do not actually alter the meaning of a predicate.

(3) yabure -teshimat -ta   -no -dearu
      rip   -COMP -PAST -NOM -COP
      c   -f_1 -f_2 -f_3 -f_4
      "(something) ripped."

(3) can be simply paraphrased as (4)

(4) yabure -ta
      rip   -PAST
      c   -f_1

In actual NLP applications such as text mining, it is essential that the system recognizes that (3) and (4) express the same event of something "*ripped*." In order to achieve this, the system needs to recognize -*teshimat, -no,* and -*dearu* as unnecessary ($f_1, f_3, f_4 \rightarrow \emptyset$). Previous studies that focus on paraphrasing of one functional expression to another ($f \rightarrow f'$) cannot solve this problem.

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

(5) (Hawai-ni) _{P1}iki, _{P2}nonbirishi -takat -ta
    (Hawaii-to) go   relax   -want -PAST
                c_1 c_2 f_1 f_2
    "I wanted to go to Hawaii and relax."

(5) has a coordinate structure, and two verbal predicates, *iki* (P1) "go" and *nonbirishi-takat-ta* (P2) "wanted to relax", are coordinated.

As the English translation indicates, the first predicate in fact means *iki-takat-ta* "wanted to

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

As shown, there are two problems that need to be solved in order for a system to extract the actual meaning of a predicate.

i. Several functional expressions are necessary for sustaining the meaning of the *event* expressed by a predicate while others barely alter the meaning ($f \rightarrow \emptyset$).
ii. Several predicates in coordinate sentences lack necessary functional expressions at the surface level ($\emptyset \rightarrow f$) and this results in a wrong extraction of the predicate meaning.

Based on syntactic and semantic theories in linguistics, we construct paraphrasing rules and solve these problems by standardizing complex functional expressions.

3 Construction of Paraphrasing Rules

The overall flow of our standardizing system is depicted in Figure 1. The system works as follows.

i. Given a parsed sentence as an input, it extracts a predicate(s) and assigns a semantic label to each functional expression based on Matsuyoshi and Sato (2006).
ii. As for an intermediate predicate, necessary functional expressions are added if missing ($\emptyset \rightarrow f$).
iii. From each predicate, delete unnecessary functional expressions that do not alter the meaning of the predicate ($f \rightarrow \emptyset$).
iv. Conjugate each element and generate a simplified predicate.

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

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

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

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

3.1 Categorization of Functional Expressions

As discussed in Section 1 and in Inui et al. (2008), what is crucial in the actual NLP applications is to be able to recognize whether two seemingly different predicates express the same *fact*.

This perspective of factuality is similar to the truth-value approach of an *event* denoted by predicates as discussed in the field of formal semantics (e.g., Chierchia and Mcconnel-Ginet, 2000; Portner, 2005). Although an extensive investigation of these theories is beyond the scope of this paper, one can see that expressions such as *tense (aspect)*, *negation* as well as *modality*, are often discussed in relation to the meaning of an *event* (Partee et al., 1990; Portner, 2005).

Tense (Aspect): Expresses the time in (at/for) which an event occurred.

Negation: Reverses the truth-value of an event.

Modality: Provides information such as possibility, obligation, and the speaker's eagerness with regard to an event and relate it to what is true in reality.

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

(6) kat -chai -takat -ta -n -da
    buy -COMP -want   -PAST -NOM -COP
       aspect modality tense(aspect)

(7) kai -takat   -ta
    buy -want   -PAST
       modality   tense(aspect)
       "wanted to buy"

The predicate "*kat-chai-takat-ta-n-da*" in (6) and "*kai-takat-ta*" in (7) express the same event because they share the same tense (past), negation (none), and modality (want). Although (6) has the completive aspect marker *-chai* while (7) does not, they still express the same fact. This is because the Japanese past tense marker *-ta* also has a function to express the completive aspect. The information expressed by *-chai* in (6) is re-
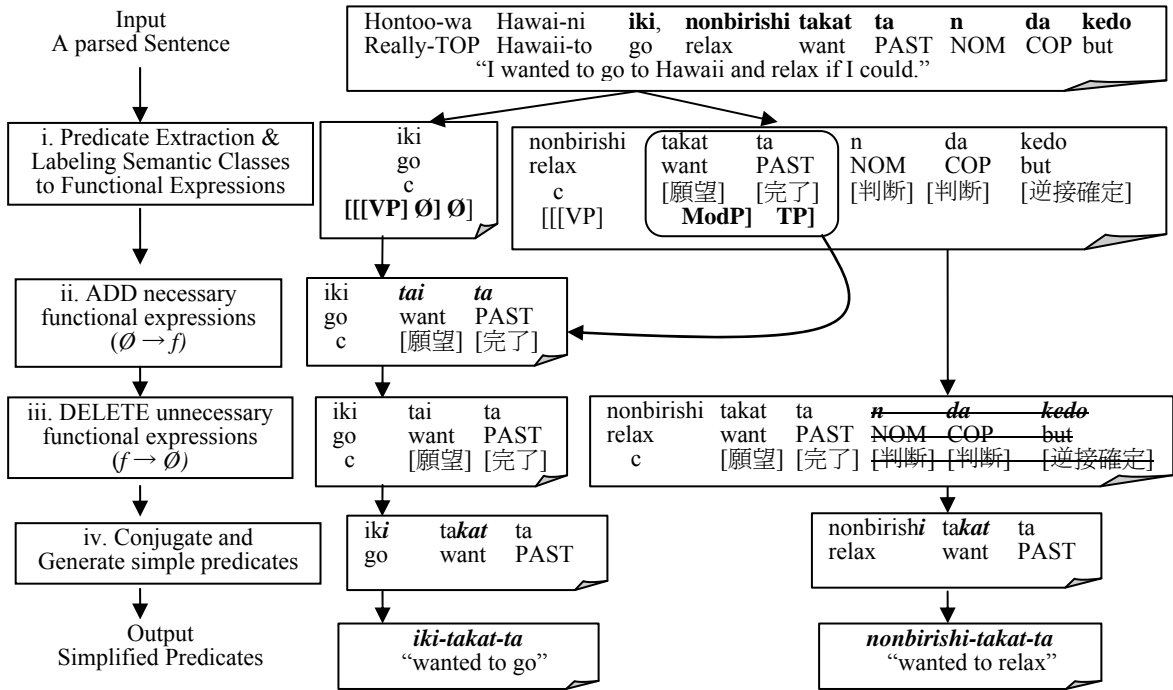
Figure 1. The flow of Standardization.

dundant and so unnecessary.

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

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

3.2 Adding Necessary Functional Expressions

The next question that we need to answer is how we find which functional expressions are missing when standardizing an intermediate predicate in a coordinate structure (e.g., (5)). We solve this based on a detailed analysis of the syntactic structure of predicates.

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

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

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

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

(8) <u>nemutai-mitai-de</u> <u>kaeri -tagatte -tei -ta</u>
sleepy-seems-COP gohome-want-CONT-PAST
   "He seemed sleepy and wanted to go home."

In (8), the first predicate *nemutai-mitai-de* "seem to be sleepy" should be paraphrased as *nemutai-mitai-dat-ta*, "seem<u>ed</u> to be sleepy," in which only the functional expression indicating *past* is required. The other functional expressions such as *tagatte* "want," and the aspect marker *tei* (CONTinuation) should *not* be added (*nemutai-mitai-de-tagat*(want)-*tei*(CONT)-*ta*(PAST) is completely ungrammatical).

67

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

(9) (imawa) <u>yasui-ga</u> (mukashiwa) <u>takakat-ta</u>
(today) inexpensive-but (in old days) expensive-PAST

"(They) <u>are inexpensive</u> (today), (but) <u>used to be very expensive</u> (in the old days.)"

In (9), the first predicate *yasui* "inexpensive" should not be paraphrased as *yasukat-ta* "was inexpensive" since this would result in the ungrammatical predicate of "*(they) *were* inexpensive (today)."

In order to add necessary functional expressions to an intermediate predicate, one needs to solve the following two problems.

  i. Find whether the target predicate indeed lacks necessary functional expressions.
  ii. If such a shortfall is detected, decide which functional expressions should be added to the predicate.

We solve these problems by turning to the *incompleteness* of the syntactic structure of a predicate.

Studies such as Cinque (2006) and Rizzi (1999) propose detailed functional phrases such as TopP (Topic Phrase) in order to fully describe the syntactic structures of a language. We adopt this idea and construct a phrase structure of Japanese predicates which borrows from the functional phrases of TP, ModP, and FocP (Figure 2).

ModP stands for a *modality* phrase and this is where modality expressions can appear.[1] FocP stands for a *focus* phrase. This is the phrase where the copula *da* appears. This phrase is needed because several modality expressions syntactically need the copula *da* in either the following or preceding position (Kato, 2007). The existence of FocP also indicates that the modality expressions within the phrase are complete (no more modality phrase is attached). TP stands for a *tense* phrase and this is where the tense marker appears.

Note that this structure is constructed for the purpose of Standardization and other functional

---

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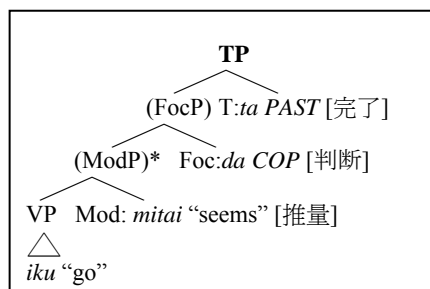

Figure 2. Structure of a predicate.

projections such as NegP (negation phrase) will not be discussed although we assume there must be one. Based on the predicate structure in Figure 2, we solve the two problems as follows.

The first problem: Detecting whether the target predicate lacks necessary functional expressions.

➢ If the predicate has the past tense marker *ta* or if the coordinate conjunction following the predicate is for combining phrases with tense, then consider the predicate as *complete* and do not add any functional expressions. Otherwise, consider the predicate as *incomplete* and add the appropriate functional expressions.

The underlying principle of this rule is that *if a predicate is tensed, then its syntactic structure is complete.* As often described in syntactic theories (e.g., Adger, 2003), a sentence can be said to be a phrase with tense (i.e., TP). In other words, if a predicate is tensed, then it can stand alone as a sentence.

By adopting this idea, we judge the completeness of a predicate by the existence of tense. Because Japanese marks past tense by the past tense marker *-ta*, if a predicate has *-ta*, it is complete and no functional expressions need be added.

However, Japanese does not hold an explicit *present* tense marker; the base form of a verb is also a present form. We solve this by looking at which conjunction follows the predicate. As discussed in Minami (1993), the finite state and the type of conjunction are related; some conjunctions follow tensed phrases while others follow infinitival phrases. Following this, we categorize all the coordinate conjunctions based on whether they can combine with a tensed phrase. These conjunctions are listed as *tensed* in Table 1. If

Not tensed	Tensed
gerundive form, *te*	*shi, dakedenaku, ueni, bakarika, hoka(ni)(wa), keredo, ga, nonitai-shi(te),ippou(de),hanmen*

Table 1. Coordinate conjunctions.

the target phrase is followed by one of those conjunctions, then we do not add any functional expressions to them because they are *complete*.

The second problem: Finding the appropriate functional expressions for incomplete intermediate predicates.

As discussed, we assume that predicates are coordinated at one of the functional phrase levels in Figure 2. Functional expressions that need to be added are, therefore, those of the *outer* phrases of the target phrase.

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

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

3.3 Implementing the Standardization

In this final subsection, we describe how we actually implement our theoretical observations in our standardization system.

CATEGORIZE functional expressions
First, we selected functional expressions that belong to our syntactic and semantic categories from those listed in Matsuyoshi and Sato (2006), a total of about 17,000 functional expressions with 95 different semantic labels. We use abstract semantic labels, such as "completion," "guess," and "desire" for the categorization (Table 2).

We divided those that did not belong to our syntactic and semantic categories into *Deletables* and *Undeletables*. *Deletables* are those that do

not alter the meaning of an event and are, therefore, unnecessary. *Undeletables* are those that are a part of content words, and so cannot be deleted (e.g., *kurai* [程度] "about" as in *1-man-en-kurai-da* "is *about* one million yen"). Based on the categorization of semantic labels as well as surface forms of functional expressions, our system works as follows;

ADD necessary functional expressions
A-1: Examine whether the target predicate has the tense marker *ta* or it is followed by the conjunctions categorized as *tensed*. If not, then go to Step A-2.
A-2: Based on the semantic label of the target predicate, decide which level of syntactic phrase the predicate projects. Add functional expressions from the last predicate that belongs to outer phrases.

DELETE unnecessary functional expressions
D-1: Delete all the functional expressions that are categorized as *Deletables*.
D-2: Leave only one functional expression if there is more than one same semantic label. For those categorized as *Negation*, however, delete all if the number of negations is even. Otherwise, leave one.
D-3: Delete those categorized as *Focus* if they do not follow or precede a functional expression categorized as *Modality*.

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

4 Experiments and Evaluations

4.1 Constructing Paraphrase Data

We selected 2,000 sentences from newspaper and blog articles in which more than one predicate were coordinated.[2] We manually extracted predicates (c-f_1-f_2..f_n). Half of them were those in which the last predicate had *three* or more functional expressions ($n \geq 3$).

Syntactic	Semantic	Semantic Labels
T if the surface is *ta*	*Tense (Aspect)*	完了(completion), 継起, 付帯, 回避, 経験, 事後, 習慣, 継続, 発継続, 着継続, 最中, 事前, 放置, 傾向
	Negation	否定(negation), 放置, 否定意志, 否定推量, 不可能, 不必要, 不許可, 不可避, 無意味
Mod	*Modality*	推量(guess), 願望(desire), 疑問, 許可, 当為, 意志, 依頼, 勧め, 勧誘, 可能, 比況, 順接 必要, 不可能, 不必要, 不許可, 回想, 不可避, 無意味
Foc	*Focus*	判断(affirmation), 名詞化, 同格
	Deletables	丁寧(politeness), 他-授与, 伝聞, 相応, 内-授与, 自然発生, 添加, 理由, 逆接確定, 感嘆, 不満, 順接確定, 順接仮定, 想外, 限定, 極端例
	Undeletables	程度(about), 終点, 根拠, は観点, も観点, 割合, 基準, 起点, 場合, 状態, 想外無視, 相関, 対象, 仲介, 定義, 範囲, 非限定, 不均衡, 立場, 同時性, 順接限定, 逆接仮定, 目的, 反復, 因状況, 対比, 適当, 状況, 話題, 並立, 相手, 目標, 主体, 強調

Table 2. Syntactic and semantic categorization of semantic labels.

We then asked one annotator with a linguistic background to paraphrase each predicate into the simplest form possible while retaining the meaning of the event.[3] We asked another annotator, who also has a background in linguistics, to check whether the paraphrased predicates made by the first annotator followed our criterion, and if not, asked the first annotator to make at least one paraphrase. 424 out of 4,939 predicates (8.5%) were judged as *not following the criterion* and were re-paraphrased. This means that the accuracy of 91.5% is the gold standard of our task.

One of the authors manually assigned a correct semantic label to each functional expression. Procedure *i* in Figure 1 is, therefore, manually implemented in our current study.

4.2 Experiments and Results

Based on the standardization rules discussed in Section 3, our system automatically paraphrased functional expressions of test predicates into simple forms. We excluded instances that had segmentation errors and those that were judged as *inappropriate as a predicate*.[4] A total of 1,501 intermediate predicates (287 for development and 1,214 for test) and 1,958 last predicates (391 for development and 1,567 for test) were transformed into simple predicates.

The accuracy was measured based on the exact match in surface forms with the manually constructed paraphrases. For comparison, we used the following baseline methods.

➢ No Add/Delete: Do not add/delete any functional expression.
➢ Simp Add: Simply add *all* functional expressions that the intermediate phrase does not have from the last predicate.

Table 3 indicates the results. Our standardizing system achieved high accuracy of around 77% and 83 % in open (against the test set) and closed tests (against the development set) compared to the baseline methods (No Add/Delete (open), 55%; Simp Add (open), 33%).

We also measured the reduced rate of differences in surface forms. We counted the number of types of functional expressions in the last predicates (a sequence of f_1-f_2-f_3 is counted as one) before and after the standardization.

For comparison, we also counted the number of functional expressions of the manually paraphrased predicates. Table 4 lists the results. As shown, our standardizing system succeeded in reducing surface differences in predicates from the original ones at the rate of 44.0%, which is quite close to the rate achieved by the human annotators (52.0%).

5 Discussion and Conclusion

Our standardization system succeeded in generating simple predicates in which only functional expressions crucial for the factual meaning of the predicate were retained.

The predicates produced by our system showed fewer variations in their surface forms while around 77% of them exactly matched the simplified predicates produced by human annotators, which is quite high compared to the baseline systems.

---

[3] We asked to delete or add functional expressions from each predicate when paraphrasing. Only the surface forms (and not semantic labels) were used for annotation.
[4] In Japanese, a gerundive form of a verb is sometimes used as a postposition. The annotators excluded these examples as "not-paraphrasable."

	Normalization	No Add/Delete	Simp Add
Open (Intermediate)	**77.7%(943/1214)**	**57.8%(702/1214)**	**32.8%(398/1214)**
Closed (Intermediate)	82.9%(238/287)	62.0%%(178/287)	35.2%(101/287)
Open (Last)	**76.2%(1194/1567)**	**51.9% (203/391)**	n.a
Closed (Last)	83.4%(326/391)	48.1%(188/391)	n.a.

Table 3. Results of our normalization system.

Original	943 types	Reduced Rate
Normalization	530 types	**44.0%**
Human Annotation	448 types	52.0%

Table 4. Reduced rate of surface forms.

This was achieved because we constructed solid paraphrasing rules by applying linguistic theories in semantics and syntax. The quite low accuracy of the baseline method, especially SimpAdd, further supports our claim that implementing linguistic theories in actual NLP applications can greatly improve system performance.

Unlike the study by Inui et al. (2008), we did not include the meaning of a *content* word for deciding the factuality of the event nor did we include it in the paraphrasing rules. This lowers the accuracy. Several functional expressions, especially those expressing *aspect,* can be deleted or added depending on the meaning of the content word. This is because content words inherently hold aspectual information, and one needs to compare it to the aspectual information expressed by functional expressions. Because we need a really complicated system to compute the abstract semantic relations between a content word and functional expressions, we leave this problem for future research.

Regardless of this, our standardizing system is useful for a lot of NLP applications let alone text mining. As mentioned in Inui et al. (2008), bag-of-words-based feature extraction is insufficient for conducting statistically-based deep semantic analysis, such as factual analysis. If standardized predicates were used instead of a single content word, we could expect an improvement in those statistically-based methods because each predicate holds important information about *fact* while differences in surface forms are quite limited.

In conclusion, we presented our system for standardizing complex functional expressions in Japanese predicates. Since our paraphrasing

rules are based on linguistic theories, we succeeded in producing simple predicates that have only the functional expressions crucial to understanding of the meaning of an *event.* Our future research will investigate the relationship between the meaning of content words and those of functional expressions in order to achieve higher accuracy. We will also investigate the impact of our standardization system on NLP applications.

References

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

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

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

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

Inui, Kentaro, Shuya Abe, Kazuo Hara, Hiraku Morita, Chitose Sao, Megumi Eguchi, Asuka Sumida, Koji Murakami, & Suguru Matsuyoshi (2008). Experience mining: Building a large-scale database of personal experiences and opinions from web documents. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 1.,* 314-321.

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

Matsuyoshi, Suguru, & Satoshi Sato (2006). Compilation of a dictionary of Japanese functional expressions with hierarchical organization. *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages*

(ICCPOL), Lecture Notes in Computer Science, Vol. 4285, 395-402.

Matsuyoshi, Suguru, & Satoshi Sato (2008). Automatic paraphrasing of Japanese functional expressions using a hierarchically organized dictionary. *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP), Vol. 1,* 691-696.

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

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

Partee, Barbara H., Alice ter Meulen, & Robert E. Wall (1990). *Mathematical methods in Linguistics.* Dordrecht, The Netherland: Kluwer.

Portner, Paul H. (2005). *What is meaning?: Fundamentals of formal semantics.* Malden, MA: Blackwell.

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

Shudo, Kosho, Toshifumi Tanabe, Masahito Takahashi, & Kenji Yoshimura (2004). MWEs as non-propositional content indicators. *Proceedings of second Association for Computational Linguistics (ACL) Workshops on Multiword Expressions: Integrating Processing,* 32-39.

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

Tsujimura, Natsuko. (2007). *An Introduction to Japanese Linguistics (2nd Ed.).* Malden, MA: Blackwell.