# Incorporating Complementary Annotation to a CCGbank for Improving Derivations for Japanese

**Sumire Uematsu** and **Yusuke Miyao**
National Institute of Informatics / JST, PRESTO
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
{uematsu,yusuke}@nii.ac.jp

## Abstract

Wide-coverage resources for lexicalized grammars have been obtained by converting the existing treebanks into collections of derivations. Additional annotations to the source treebank can be used to improve these derivations. A treebank annotation called the NTT treebank was used for this paper to improve a CCGbank for Japanese. The source treebank of the CCGbank itself is created by automatically converting chunk-dependencies, but the CCGbank contains errors caused by noisier phrase structures and a lack of linguistic information, which is difficult to represent in chunk-dependency. The NTT treebank provides cleaner trees and functional and semantic information, e.g., coordinations and predicate-argument structures. The effect of the improvement process is empirically evaluated in terms of the changes in the dependency relations extracted from the resulting derivations.

## 1 Introduction

Wide-coverage resources for lexicalized grammars have been created by converting the existing treebanks into collections of derivations for the target grammars (Miyao and Tsujii, 2008; Hockenmaier and Steedman, 2007; Hockenmaier, 2006). However, the source corpora, such as the Penn Treebank (Marcus et al., 1993), often lack the necessary linguistic information for constructing these derivations, and this can create noise in the resulting derivations. Therefore, complementary annotations to the source treebank, e.g., NP bracketing and semantic roles, have been used used to improve the derivations (Honnibal et al., 2010; Vadas and Curran, 2008).

It is especially important to reduce the amount of noise in the derivations in the CCGbank for Japanese (Uematsu et al., 2015). Since the source treebank itself was created by converting the chunk-dependency, it potentially includes even more errors in the phrase structures and other types of information such as the functional tags. For instance, the chunk-dependency is often insufficient for correctly deciding on the phrase structure of coordinated NPs with modifiers. Since the dependencies do not encode the left boundary of each NP (Asahara, 2013), a manual annotation is needed for the precise structure. In fact, it essentially lacks any linguistic information which is difficult to represent in the chunk-dependency, e.g., the coordinated arguments.

The NTT treebank (Tanaka and Nagata, 2013) was used to improve the Japanese CCGbank for this paper. Basically the treebank is a manually corrected version of the source treebank used to create the CCGbank, but we treat the treebank as a collection of additional annotations to the source treebank. We specifically use its cleaner phrase structures to correct the phrase structure errors and its functional tags to properly deal with the coordinations in the derivations. Moreover, we use the annotations of causer roles in causative constructions in the NTT treebank, which are not available in the original syntactic resources, to recognize the arguments in the derivations. We show how the functional and semantic annotations on the treebank will be used for improving the CCGbank.

The improvement process together with the conversion in our previous work (Uematsu et al., 2015) can be regarded as a framework for obtaining a Japanese treebank and a derivation bank at a lower cost. That is, we can obtain a clean treebank containing rich linguistic information by 1) translating the existing syntactic resources, e.g., the chunk-dependency annotation, to a treebank, 2) manually correcting the phrase structures, and 3) using the cleaner treebank as a base for additional annotations. The treebank and the related



*Proceedings of the 14th International Conference on Parsing Technologies*, pages 20–29,
Bilbao, Spain; July 22–24, 2015. 

$$\begin{array}{llll} \mathrm{X/Y}: f \quad \mathrm{Y}: a & \rightarrow & \mathrm{X}: fa & (>) \\ \mathrm{Y}: a \quad \mathrm{X\backslash Y}: a & \rightarrow & \mathrm{X}: fa & (<) \\ \mathrm{X/Y}: f \quad \mathrm{Y/Z}: g & \rightarrow & \mathrm{X/Z}: \lambda x.f(gx) & (>\mathrm{B}) \\ \mathrm{Y\backslash Z}: g \quad \mathrm{X\backslash Y}: f & \rightarrow & \mathrm{X\backslash Z}: \lambda x.f(gx) & (<\mathrm{B}) \end{array}$$

Figure 1: Combinatory rules in Japanese CCGbank.

resources are hopefully applicable to other grammar formalisms.

## 2 CCGbank for Japanese

### 2.1 Combinatory Categorial Grammar

Combinatory Categorial Grammar (CCG) is a lexicalized grammar formalism that is widely accepted in the NLP fields (Steedman, 2001). We briefly introduce its basic elements below.

A CCG grammar has two elements: *categories* for expressing the syntactic characteristic of the words and phrases and *combinatory rules* for combining the categories. There are two types of categories, *ground* and *complex*. *Ground categories* include S and NP, and *complex categories* are either $\mathrm{X/Y}$ or $\mathrm{X\backslash Y}$, where X and Y are the categories. Category $\mathrm{X/Y}$ means that it becomes a category X when it is combined with another category Y to its right, and $\mathrm{X\backslash Y}$ means it takes on a category Y to its left. For example, categories $\mathrm{S\backslash NP}$ and $\mathrm{S/NP\backslash NP}$ represent an English intransitive verb and a transitive one, respectively.

Combinatory rules (Fig. 1) are applied to the categories to form categories for larger phrases. For example, a subject NP and intransitive verb $\mathrm{S\backslash NP}$ are combined to form a sentence S by applying the backward application rule ($<$ in Fig. 1). Figure 2 shows a CCG analysis of a simple Japanese sentence, which is called a *derivation*.

### 2.2 CCG-based syntactic theory for Japanese

We briefly describe Bekki's theoretical work on Japanese syntax (Bekki, 2010), which is the basis of the analysis in the Japanese CCGbank (Fig. 2). Based on CCG, his theory provides a comprehensive description for a variety of morphological and syntactic constructions, such as agglutination, scrambling, and long-distance dependencies.

There are three types of ground categories in his theory: S for sentences, NP for noun and postposition phrases, and CONJ for conjunctions. Categories S and NP have the syntactic features of *form* and *case*, respectively. Table 1 itemizes the values of these syntactic features.

| Cat. | Feature | Value | Interpretation |
|---|---|---|---|
| NP | case | ga | nominative |
| | | o | accusative |
| | | ni | dative |
| | | to | comitative, complementizer, etc. |
| | | nc | none |
| S | form | stem | stem |
| | | base | base |
| | | neg | imperfect or negative |
| | | cont | continuative |
| | | vo_s | causative |

Table 1: Features for Japanese syntax in (Uematsu et al., 2015).

Predicative words, such as verbs and adjectives, are represented as $\mathrm{S\backslash NP_{ga}}$, $\mathrm{S\backslash NP_{ni}\backslash NP_{ga}}$, etc., depending on their mandatory arguments. For example, $\mathrm{S\backslash NP_{ga}}$ is for intransitive verbs and for adjectives, and $\mathrm{S\backslash NP_{ga}\backslash NP_{o}}$ represents a transitive verb. Postpositions that work as argument markers include $\mathrm{NP_{ga}\backslash NP_{nc}}$ $\mathrm{NP_{ni}\backslash NP_{nc}}$, etc. For example, "が NOM ga" is represented as $\mathrm{NP_{ga}\backslash NP_{nc}}$ as it takes on the left NP to form a nominative NP. Postpositions can be used to form modifier phrases to verbal and adjective phrases. For example, "に ni" is $\mathrm{S/S\backslash NP}$ if it takes on the left NP to form a temporal or a locative modifier.

The treatment of auxiliary verbs differs here from the English CCG. In Japanese, auxiliary verbs follow right after the main verb and express the semantic information, such as the tense and modality. For example, a verb "選ば/choose-NEG" and auxiliaries "なかっ/not-CONT" and "た/PAST-BASE" form a VP "選ばなかった", which means "did not choose". Auxiliary verbs are expressed as the category $\mathrm{S\backslash S}$, and the category is combined with a main verb via the function composition rule ($<$B in Fig. 1), as shown in Fig. 2.

Bekki's theory treats the coordination in a similar way as in (Steedman, 2001). There is a special rule for coordination $\Phi$, with the restriction that X must be in a form of $\mathrm{T/(T\backslash\$)}$, e.g., $\mathrm{NP_{nc}/NP_{nc}}$ and $\mathrm{S/NP\backslash(S/NP)}$.

$$\mathrm{X_1} \quad \ldots \quad \mathrm{CONJ} \quad \mathrm{X}_m \rightarrow \quad \mathrm{X} \quad (\Phi) \qquad (1)$$

### 2.3 Japanese CCGbank

We proposed an algorithm in our previous work (Uematsu et al., 2015) to convert the existing chunk-dependency resources into CCG derivations for Japanese sentences. We refer to the collection of derivations obtained using this method as the original CCGbank for Japanese. Two steps

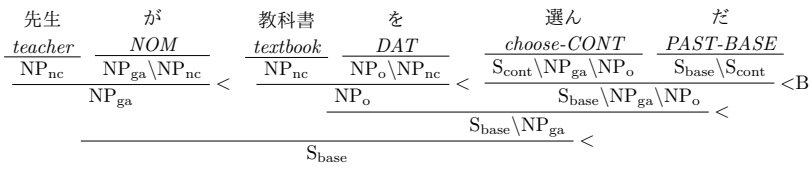


Figure 2: CCG derivation for Japanese sentence "The teacher chose the textbook."

are needed to complete the conversion. First, we integrates chunk-dependencies from the Kyoto corpus (Kurohashi and Nagao, 2003) and a type of semantic role annotations from the NAIST corpus (Iida et al., 2007) to create tree structures with predicate argument structures (PASs). The trees are then translated into CCG derivations. 94% of the sentences in the source corpus were considered to be successfully converted into derivations. The lexicon extracted from the CCGbank has a lexical coverage of 99.4% and a sentential coverage of 87.0% on unseen text.

One of the obstacles in the conversion is often insufficient information for recovering the phrase structures. Several heuristic rules were used to complement for this lack, but we must make manual annotations, especially for the types of information that are difficult to represent in the chunk-dependency, e.g., coordinated arguments. The conversion errors due to the lack o information resulted in erroneous substructures in the trees and derivations and this leads to noises in the obtained grammar.

As a result, the grammar of the CCGbank is simplified for some constructions, specifically the coordinations. It treats the NP coordinations as noun-noun modifications. VP and ADJP coordinations are implicitly handled as a type of continuous clauses. By incorporating additional annotations into the derivation, our new procedure identifies the NP coordination and improves the substructure. On the other hand, we kept the VP coordinations as a type of continuous clauses, because it is difficult to distinguish between the VP coordination and other types of continuous clauses based only on the shallow semantic information.

## 3 NTT treebank

The phrase structures and supplementary information in NTT treebank (NTB) are annotated to news-wire text (Tanaka and Nagata, 2013). As supplementary annotations, the treebank contains functional tags and predicate argument structures.

| Grammatical role for mandatory argument | |
|---|---|
| -SBJ | Subjective case |
| -OBJ | Objective case |
| -OB2 | Indirect object case |
| -COORD | Coordination |
| -APPOS | Apposition |

Table 2: Function tags in NTT treebank.

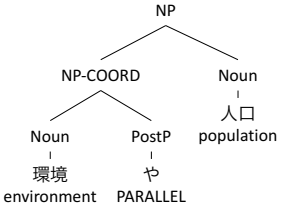


Figure 3: NP coordination in NTT treebank.

Table 2 lists some function tag examples that are annotated to the tree nodes. SBJ and OBJ represent the grammatical roles of phrases, and COORD shows the annotated node and sibling node are coordinated (Fig. 3). Predicate argument structures are presented as relations between the predicate words and their argument phrases, which is similar to the annotation style of PropBank (Palmer et al., 2005) (Fig. 5).

The treebank is created by manually correcting and updating the base annotation, which is actually the source treebank used to build the original Japanese CCGbank. It is based on the dependency between chunks or *bunsetsu* of the Kyoto corpus (Kurohashi and Nagao, 2003), but manual annotation made the treebank cleaner and richer. In addition to fixing the apparent errors such as the tokenization errors and erroneous POS tags, the manual annotation includes modifying the subtrees for specific constructions (e.g., coordinated phrases), a clause with a formal noun, and a PP with a compound postposition. Moreover, PAS annotations for specific voices, such as causatives and benefic ials were added to the treebank.

Compound postposition is a type of multi-word expressions in which a combination of postpositions, verbs, and auxiliaries works as one post-

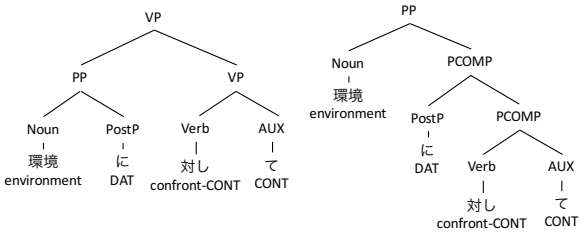


Figure 4: Subtrees with compound postposition “に対して” before (left) and after manual annotation (right).

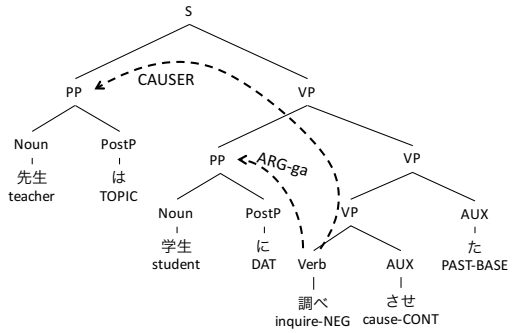


Figure 5: Causative sentence “The teacher has the student inquire” and PAS annotated for “調べ inquire”. The dotted arc labeled with ARG-ga denotes that the agent of “inquire” is “the student”

position. For example, a compound postposition ” に/DAT/ni 対し/confront-CONT/taishi て/aux-CONT/te” typically functions similarly to the postposition “に/DAT/ni”. Fig. 4 shows the subtrees before and after the update. Since the structure on the left is the same as the structure for a continuous clause, it is difficult to distinguish compound postpositions and VPs. After the manual annotation, the compound postpositions are marked with “PCOMP” tags and have a specific structure, as shown on the right in Fig. 4

The PAS annotation on the base treebank, which is obtained by converting the word-to-word annotation of the NAIST corpus (Iida et al., 2007), was also manually corrected and populated. An important addition to the PAS is the annotation of causative and beneficial constructions. The original treebank (and the NAIST corpus) also identifies causative constructions, but there are very few annotations for the causer role, which typically occurs with the case marker “が/ga” or its topicalized form. Fig. 5 shows an example of the annotation of a causative construction with a causer role.

## 4 Related work

The corpus-based acquisition of wide-coverage CCG resources has been very successful for English (Hockenmaier and Steedman, 2007). Their method converts the Penn Treebank (Marcus et al., 1993) into CCGbank, which is a collection of CCG derivations, and extracts a wide-coverage lexicon from the derivations. The CCGbank is also used to train a robust CCG parser (Clark and Curran, 2007).

Complementary resources on the Penn Treebank are used to improved the derivations because the treebank does not contain some of the linguistic information necessary for a CCG derivation. Boxwell and White (2008) augmented the English CCGbank with the semantic roles in PropBank (Palmer et al., 2005). Honnibal et al. (2010) integrated several types of additional annotations such as PropBank and NP structure annotation (Vadas and Curran, 2007), to improve the CCGbank. Our work basically follows these methods, but we have to deal with noises in the treebank that are caused by the dependency-to-tree conversion errors.

Our previous work (Uematsu et al., 2015) extended the method used for the English CCGbank, and obtained wide-coverage CCG resources for Japanese. A treebank is created in this method by converting chunk-based annotation resources. The treebank is then converted into a CCGbank for Japanese, which can be used to obtain wide-coverage lexicon and parsers for Japanese CCG.

Other than the one mentioned above, there are several other studies on Japanese deep parsing. The theoretical work by Gunji (1987) describes Japanese phenomenon based HPSG. Komagata (1999) proposed a CCG-based theory and implementation, but the focus is not on processing real world texts. JACY (Siegel and Bender, 2002) is a type of hand-crafted Japanese grammar based on HPSG that can compute a detailed semantic representation. One of our future goals is to obtain CCG resources that allow for a more precise and detailed description by incorporating additional annotations into CCGbank.

## 5 Incorporating additional annotation into CCGbank

We describe the two steps needed to incorporate the annotations of the NTT treebank (NTB) into the Japanese CCGbank. First, we reconstruct the CCGbank according to the clean phrase structure

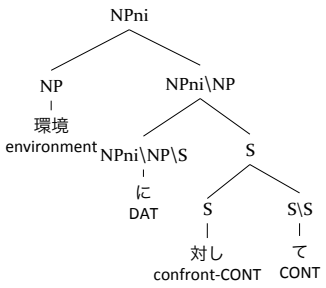


Figure 6: Substructure for argument PPs with compound postpositions.

$$\begin{array}{llll} X & CONJ & \rightarrow & X_{conj} \quad (Coord1) \\ X_{conj} & X & \rightarrow & X \quad (Coord2) \end{array}$$

Figure 7: Special rules for coordination.

in the treebank, and then, change the substructures of the CCG derivations based on the functional tags and predicate-argument annotations of the treebank.

### 5.1 Reconstruction of CCGbank

As stated in Section 3, the trees in NTB are a manually corrected version of those used in (Uematsu et al., 2015). The reconstruction of the Japanese CCGbank is basically done by following the conversion rules used in (Uematsu et al., 2015). A drastic change of the conversion rules is not necessary because most of the changes in NTB are error corrections. However, we have to add a conversion rule for a compound postposition in order to handle the structure change illustrated in Sec. 3.

The treatment of compound postpositions is not explicitly described in (Bekki, 2010), so we defined two types of structures for these compounds. As with a normal postposition, a compound postposition either works as an argument marker or forms an adjunct PP. Fig. 6 shows the defined substructure for an argument PP using the compounds. The structure for case markers (left in the figure) is designed so that the node for the compound postposition (right child of the top node in the figure) is assigned the category for argument marker ($NP_{ni}\backslash NP_{nc}$ in the example).

We added a conversion rule that first detects a PP with a compound postposition by searching for a node with a PP label whose right daughter is PCOMP, and then, checks whether the node is identified as an argument by the original conversion rules, and finally assigns categories to the nodes in the PP according to the substructure shown in Fig. 6 if the node is found to be an argument.

### 5.2 Incorporation of the additional annotation to CCGbank

The process to incorporate complementary linguistic information into the CCGbank follows (Honnibal et al., 2010), but we have to deal with constructions that are specific to Japanese. We describe how to improve the treatment of the coordination as an example of handling functional tag annotations, and present how to identify the causer roles by processing the semantic annotations. Finally, we check the consistency of the changes.

#### 5.2.1 Coordination

We added a new syntactic feature *conj* and two special rules that are presented in Fig. 7 to the grammar to deal with the argument coordination in derivations. The new feature indicates whether the phrase includes a conjunction. The new rules are the result of adapting the coordination rule ($\Phi$) into binary-branching of the CCGbank. Similar rules were used to deal with the coordinations in the English CCGbank (Hockenmaier and Steedman, 2007).

We replace the analyses of the coordinations with ones incorporated with the special rules by the following process. First, a noun phrase including a coordination is detected by a subtree where an top NP node contains an NP node with a COORD tag as its left child, and the NP node with the COORD tag has a punctuation or parallel particle as its right child (the left tree in Fig. 8). The categories corresponding to the top node (a basic category NP in the example) are then checked to see if the condition for $\Phi$ is satisfied. If satisfied, the rule combining the left and right daughters is changed to Coord2. For the example in the figure, the category for "食料 / food" will be NP, and that of the left daughter will be $NP_{conj}$ after the change. The rule to form the left daughter is also replaced by Coord1, so the category for the conjunction changes to CONJ.

#### 5.2.2 Causer argument

The grammar for the original CCGbank can handle causer arguments as well as other types of arguments. An example analysis of a causative sentence is shown in Fig. 9. The semantic representation is omitted in this figure, but the causative verb "調べ / inquire" has a causer argument in its

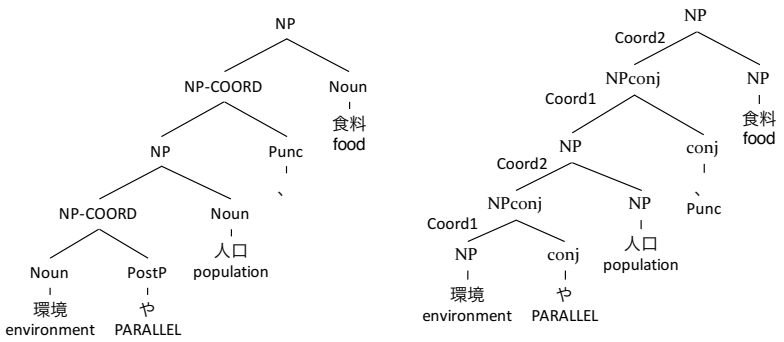


Figure 8: Subtree involving coordination (left) and new analysis for phrase (right).

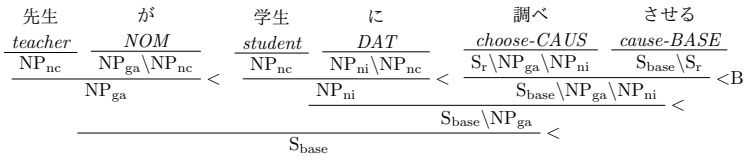


Figure 9: CCG derivation for Japanese sentence "The teacher has the student inquire."

| | Train. | Devel. | Test |
|---|---|---|---|
| sentence | 6,800 | 800 | 2,400 |
| tokens | 178,732 | 24,159 | 64,824 |

Table 3: Statistics of input linguistic resources

predicate-argument structure, and the causer argument is co-indexed with the $\mathrm{NP_{ga}}$. We reanalyzed the causative constructions in the original CCGbank based on the annotation instances of the causers added in the NTB.

First, we describe the changes to the argument phrases in the causative constructions. In Japanese, an argument to a verb is typically followed by a case marker particle ("が / ga / NOM", "に / ni / DAT", etc.) or a binding particle ("は / wa / TOPIC", "も / mo "). Phrases headed by a binding particle are used when an argument is topicalized and the case for the topicalized argument must be estimated. On the other hand, a phrase with a case marker or a binding particle can be used as a modifier to a verb. Therefore, properly distinguishing the arguments and modifiers is important for building derivations. Moreover, a causative is a construction involving case alternations (see Fig. 5), so the surface and deep cases of each argument must be decided according to the PAS annotations.

Concretely speaking, we changed the substructure for a causative in the following process. A candidate for a causer argument is detected as a phrase headed by a case marker or binding particle that is combined with a VP headed by a causative verb in the treebank. For example, the PP "先生は / teacher-TOPIC" in Fig. 5 satisfies these conditions. Next, the argument / adjunct distinction of the phrase is updated by simply checking the new PAS annotations of NTB. If the phrase is found to be a causer argument, the category for the phrase is changed to $\mathrm{NP_{ga}}$ because the surface case of the causer is always *ga*. The category for the VP is also changed to the one with the added argument. Fig. 9 shows an example of the change in causative constructions. The category for the PP "先生は / teacher-TOPIC" changes from a modifier $\mathrm{S/S}$ to an argument $\mathrm{NP_{ga}}$. We transfer the changes to the descendant nodes and obtain the new derivation, as shown on the right in the figure.

### 5.2.3 Consistency check

Finally, we check to see if the modified parts in a derivation are consistent with each other. This is done by applying the combinatory rule assigned to the each branching in the derivation in a top-down order. We discard the modifications to a derivation if they are found to be inconsistent.

## 6 Evaluation

We actually applied the improvement process to the Japanese CCGbank to evaluate it. The Japanese CCGbank we used was the version in December 2014. We used the preliminary version for the NTT treebank (NTB) that contained 10,000 trees with functional tags and PAS annota-

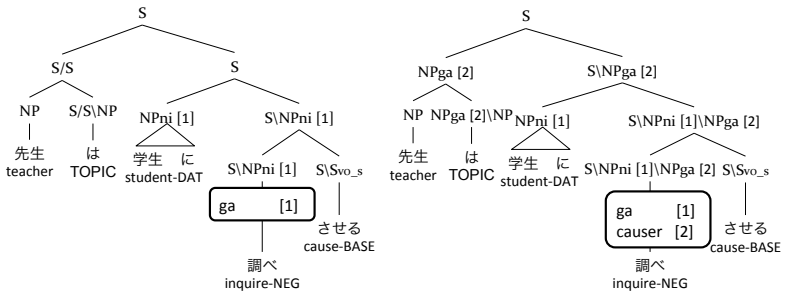


Figure 10: Causative construction with causer argument (left) and its reanalysis (right).

| | Training | | Development | | Test | |
|---|---|---|---|---|---|---|
| | Annot. | Changed | Annot. | Changed | Annot. | Changed |
| Causative | 195 | – | 20 | – | 80 | – |
| Causer | 34 | 22 | 3 | 3 | 15 | 10 |
| NP Coord | 2,632 | 2,148 | 323 | 259 | 826 | 652 |

Table 4: Statistics of annotated and reanalyzed instances

tions. We divided the 10,000 sentences into three sets: training, development, and test. Table 3 classifies the statistics of the sets. Since the original CCGbank consists of approximately 38,400 derivations, out experiments were performed on only 26% of the derivations.

### 6.1 Evaluation of the derivation changes

Table 4 lists the numbers of annotation instances and the number of the changes made in each set. We also measured the similarity between the original CCGbank and the new ones following (Honnibal et al., 2010). In other words, we used the difference in dependencies as the difference metrics, where a dependency is defined as a 4-tuple: a head of a functor, a functor category, an argument slot, and a head of an argument. Table 5 lists the percentages of the labeled and unlabeled dependencies left unchanged after the reanalysis processes. A labeled dependency is marked as unchanged if the four elements match a tuple in the original. An unlabeled dependency is correct if the heads of the functor and the argument appear together in the original.

We see from Table 5 that the most influential change was the correction of the phrase structures, which includes the changes for the compound postpositions. On the other hand, the effect of the causer arguments was fairly limited partly because there are only a small number of annotation instances for the causers.

In order to evaluate the quality of the changes, we randomly sampled fifty derivations from those

| Corpus | L.Deps | U.Deps | Cat |
|---|---|---|---|
| + Correction | 81.3 | 86.7 | 85.9 |
| + Causer | 81.0 | 86.6 | 85.7 |
| + NP Coord | 77.9 | 84.1 | 83.2 |

Table 5: Rate of dependencies and categories left unchanged in development set.

| Num. | Type of change |
|---|---|
| 32 | Change in subcategorization |
| 19 | From modifier to argument |
| 18 | Change to CONJ |
| 11 | From modifying noun NP/NP to NP |
| 8 | From S/S\NP to NP/NP\NP |

Table 6: Most frequent changes in lexical categories.

that underwent the reanalysis, and manually investigated the samples. We first checked the changes in lexical categories and category dependencies, and then referred to the derivations for the cause.

#### 6.1.1 Changes in lexical categories

The lexical categories for 135 tokens in the fifty sentences changed after the reanalysis[1]. Table 6 classifies the most frequent types of category changes.

The most and second-most frequent types are related to the adjunct versus argument decisions. Due to the corrected PASs and the additional causer annotations in the NTB, some postposition phrases previously marked as adjuncts are now arguments in the new analysis, and vice versa.

[1] We excluded tokens with different word boundaries after the reconstruction from the number.

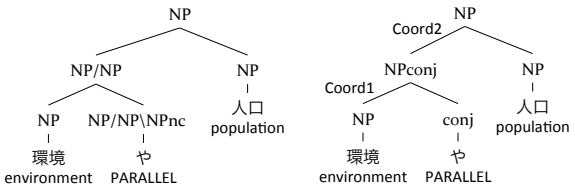


Figure 11: Subtree involving coordination (left) and new analysis for phrase (right).

In the example shown in Fig. 10, the PP "先生 は / teacher-TOPIC" is changed to an argument. This has the category for "調べ / inquire-CONT" change from $S\backslash NP_{ni}$ to $S\backslash NP_{ni}\backslash NP_{ga}$, and the category for "は / TOPIC" shifted from $S/S\backslash NP_{nc}$ to $NP_{ga}\backslash NP_{nc}$ The former belongs to the most frequent type and the latter belongs to the second. These types of changes are also caused by corrections in the morphological information, e.g., POS tags. For example, a PP headed by a postposition "と" turned into an argument by fixing the erroneous POS "conjunctive particle" into the correct "case marker particle" one.

The third and fourth ones are due to the introduction of noun coordinations. For these types, the conjunctions previously treated as NP/NP\NP etc. improved to CONJ (see Fig. 11). The fourth type also resulted from corrections in the internal structures of the NPs.

We marked each of the changes in the lexical category as "good", "bad" (for deletion of obvious arguments etc.), and "cannot decide" (for cases where categories are not correct before or after the change). We found that 79% of the changes (107 categories) were judged as good.

#### 6.1.2 Changes in dependencies

Next, we investigated the difference in dependency relations. For the fifty sampled derivations, the dependencies were extracted from the original CCGbank and our resulting CCGbank, and the two sets of dependencies were then manually compared. We focused on the dependency relations that were not shared by the two sets. In other words, we examined 348 dependency tuples unique to the original CCGbank, and 337 tuples that were only extracted from our resulting CCGbank. Tables 7 and 8 list the most frequent causes of the changes in category dependency counted in the original CCGbank and ours.

In both sets, over 90 relations only differ in the word boundaries to their counterparts. This is due to the correction of the word boundary given by the NTB, and suggests that the rate of the virtually unchanged relations is larger than those listed in Table 5.

The second most frequent causes in both sets is related to the change between the adjuncts and arguments described in Sec. 6.1.1. If a PP changed from an adjunct to an argument like in the example shown in Fig. 10, the category for the postposition ("は / TOPIC" in the figure) turns into $NP_{ga}\backslash NP_{nc}$ from $S/S\backslash NP_{nc}$. This resulted in a deletion of the relations that the old category had with the left noun ("先生 / teacher" in the figure) and the main verb ("調べ / inquire "), and the addition of a relation between the new category and the left noun. The change between the adjuncts and arguments also affected the subcategorization of the predicatives, and this is counted as a change in the subcategorization (fifth in both Tables 7 and 8)

As in Sec. 6.1.1, we also marked each of the unshared dependencies as "good", "bad", or "cannot decide". We excluded the relations changed by the correction in the word boundary. Note that any dependency in the original CCGbank is marked as "good" when its deletion or change is desirable for improving the derivation. Table 9 lists the number of relations for each of the marks. More than 80% of the changes in dependency are considered to be desirable in both sets.

| | Original | Results |
|---|---|---|
| Good | 207 | 196 |
| Bad | 34 | 34 |
| Other | 14 | 15 |
| Total | 255 | 245 |

Table 9: Judgment on unshared dependencies in original and resulting CCGbank.

| CCGbank | # Cat. | Sent. cov. |
|---|---|---|
| Original | 606 | 78.1 |
| + Correction | 690 | 76.6 |
| + Causer | 702 | 75.4 |
| + Coord. | 693 | 75.2 |

Table 10: No. of category types and sentential coverage of lexicon extracted from each CCGbank version.

### 6.2 Evaluation of the obtained resources

#### 6.2.1 Lexical categories

Table 10 lists the number of category types in the CCGbank and the coverage of the lexicon on the

| Num. | Type of change |
|---|---|
| 93 | Change in word boundary |
| 49 | From verb modifier to argument |
| 26 | From modifying noun NP/NP to NP |
| 22 | Head change in argument NP |
| 11 | Change in subcategorization |
| 11 | From noun modifier to $S\backslash NP_{ga}\backslash NP$ |

Table 7: Most frequent types of change in category dependency (counted in original CCGbank)

| Num. | Type of change |
|---|---|
| 92 | Change in word boundary |
| 47 | From verb modifier to argument |
| 22 | Head change in argument NP |
| 19 | Dependency for coordinated arguments |
| 16 | Change in subcategorization |

Table 8: Most frequent types of change in category dependency (counted in resulting CCGbank)

| | Development | | | | Test | | | |
|---|---|---|---|---|---|---|---|---|
| | LP | LR | UP | UR | LP | LR | UP | UR |
| Original | 84.54 | 81.02 | 90.78 | 87.00 | 85.00 | 81.03 | 91.15 | 86.90 |
| + Correction | 85.05 | 82.24 | 91.33 | 88.32 | 85.24 | 81.44 | 91.99 | 87.89 |
| + Causer | 84.84 | 82.13 | 91.34 | 88.41 | 84.87 | 81.22 | 91.83 | 87.88 |
| + Coord. | 82.71 | 79.78 | 89.39 | 86.22 | 82.60 | 78.71 | 89.81 | 85.59 |

Table 11: Parsing accuracy

unseen text for each CCGbank version. We measured the coverage in the same way as in (Uematsu et al., 2015), that is, we obtained improved CCG derivations for the test set by applying our method to the original CCGbank, and used them as the "gold-standard". The lexical coverage was around 98.8% for all the versions. The sentential coverage indicates the number of sentences in which all the words were assigned gold-standard categories.

After applying the change to the derivations, the numbers of category types increased and the coverage dropped 2-3%. This is due to the distinctions we added to the grammar, such as the compound postpositions versus the continuous clauses.

#### 6.2.2 Parsing accuracy

We trained a statistical parser on different versions of the augmented CCGbank, and tested on the unseen text. Table 11 itemizes the performance of the parsers trained on each version of the CCGbank. The parser we used is the same one as that used in our previous work (Uematsu et al., 2015), and no tuning was performed. The evaluation measures were the recall and precision over the category dependency. As we stated above, the size of the NTB used in this experiment was 26% of the original CCGbank, so the numbers are not directly comparable to the results using the original. Note that the table suggests how hard it is to recover the structures in each CCGbank, rather than how good the CCGbank is, because each line represents a different gold standard. A possible explanation for the slight improvement in the performance after correcting the trees is that the manual correction increased the consistency in the structures.

## 7 Conclusion

A method for improving the Japanese CCGbank by integrating CCG derivations and additional annotations to the source treebank was presented in this paper. For the Japanese CCGbank, the source treebank itself is a result of converting chunk-dependencies, and the treebank potentially includes more errors in the phrase structures and other types of information such as functional tags. Moreover, it essentially lacks linguistic information which is difficult to represent in the chunk-dependency, e.g., coordinated arguments.

We showed how functional and semantic annotations on the treebank can be used for improvement of the Japanese CCGbank by incorporating annotations of the NTT treebank, especially those for coordinations and causer roles. The process first reconstructs the CCGbank by using cleaner trees from the NTT treebank and the original tree-to-derivation conversion. The new derivations are then modified according to the functional tags and PAS information of the NTT treebank.

The empirical evaluation on the dependency relations shows that the improving process changed 23% of the dependencies extracted from the original CCGbank. A manual investigation of the changes suggests that approximately 80% of the changes are desirable and that the resulting derivations are more accurate in recognizing arguments and coordinations.

## References


Masayuki Asahara. 2013. Comparison of syntactic dependency annotation schemata. In *Proceedings of the third Japanese Corpus Linguistics Workshop*. (In Japanese).

Daisuke Bekki. 2010. *Formal Theory of Japanese Syntax*. Kuroshio Shuppan. (In Japanese).

Stephen Boxwell and Michael White. 2008. Projecting Propbank roles onto the CCGbank. In *Proceedings of LREC 2008*.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4).

Takao Gunji. 1987. *Japanese Phrase Structure Grammar: A Unification-based Approach*. D. Reidel.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Julia Hockenmaier. 2006. Creating a CCGbank and a wide-coverage CCG lexicon for German. In *Proceedings of COLING/ACL 2006*.

Matthew Honnibal, James R. Curran, and Johan Bos. 2010. Rebanking CCGbank for improved NP interpretation. In *Proceedings of ACL 2010*, pages 207–215.

Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007. Annotating a Japanese text corpus with predicate-argument and coreference relations. In *Proceedings of Linguistic Annotation Workshop*.

Nobo Komagata. 1999. *Information Structure in Texts: A Computational Analysis of Contextual Appropriateness in English and Japanese*. Ph.D. thesis, University of Pennsylvania.

Sadao Kurohashi and Makoto Nagao. 2003. Building a Japanese parsed corpus. In *Treebanks*, volume 20 of *Text, Speech and Language Technology*, pages 249–260. Springer Netherlands.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34(1):35–80.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Melanie Siegel and Emily M. Bender. 2002. Efficient deep processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization*.

Mark Steedman. 2001. *The Syntactic Process*. MIT Press.

Takaaki Tanaka and Masaaki Nagata. 2013. Constructing a practical constituent parser from a Japanese treebank with function labels. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 108–118.

Sumire Uematsu, Takuya Matsuzaki, Hiroki Hanaoka, Yusuke Miyao, and Hideki Mima. 2015. Integrating multiple dependency corpora for inducing wide-coverage Japanese CCG resources. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 14(1):1–24.

David Vadas and James Curran. 2007. Adding noun phrase structure to the Penn Treebank. In *Proceedings of ACL 2007*, pages 240–247.

David Vadas and James R. Curran. 2008. Parsing noun phrase structure with CCG. In *Proceedings of ACL-08: HLT*, pages 335–343.