Constructing a Practical Constituent Parser from a Japanese Treebank with Function Labels

Takaaki Tanaka and **Masaaki Nagata**
NTT Communication Science Laboratories
Nippon Telegraph and Telephone Corporation

{tanaka.takaaki, nagata.masaaki}@lab.ntt.co.jp

Abstract

We present an empirical study on constructing a Japanese constituent parser, which can output function labels to deal with more detailed syntactic information. Japanese syntactic parse trees are usually represented as unlabeled dependency structure between bunsetsu chunks, however, such expression is insufficient to uncover the syntactic information about distinction between complements and adjuncts and coordination structure, which is required for practical applications such as syntactic reordering of machine translation. We describe a preliminary effort on constructing a Japanese constituent parser by a Penn Treebank style treebank semi-automatically made from a dependency-based corpus. The evaluations show the parser trained on the treebank has comparable bracketing accuracy as conventional bunsetsu-based parsers, and can output such function labels as the grammatical role of the argument and the type of adnominal phrases.

1 Introduction

In Japanese NLP, syntactic structures are usually represented as dependencies between grammatical chunks called *bunsetsus*. A bunsetsu is a grammatical and phonological unit in Japanese, which consists of an independent-word such as noun, verb or adverb followed by a sequence of zero or more dependent-words such as auxiliary verbs, postpositional particles or sentence final particles. It is one of main features of Japanese that bunsetsu order is much less constrained than phrase order in English. Since dependency between bunsetsus can treat flexible bunsetsu order, most publicly available Japanese parsers including CaboCha (Kudo et al., 2002) and KNP (Kawahara et al., 2006) return bunsetsu-based dependency as syntactic structure. Such bunsetsu-based parsers generally perform with high accuracy and have been widely used for various NLP applications.

However, bunsetsu-based representations also have serious shortcomings for dealing with Japanese sentence hierarchy. The internal structure of a bunsetsu has strong morphotactic constraints in contrast to flexible bunsetsu order. A Japanese predicate bunsetsu consists of a main verb followed by a sequence of auxiliary verbs and sentence final particles. There is an almost one-dimensional order in the verbal constituents, which reflects the basic hierarchy of the Japanese sentence structure including voice, tense, aspect and modality. Bunsetsu-based representation cannot provide the linguistic structure that reflects the basic sentence hierarchy.

Moreover, bunsetsu-based structures are unsuitable for representing such nesting structure as coordinating conjunctions. For instance, bunsetsu representation of a noun phrase "技術-の (technology-GEN) / 向上-と (improvement-CONJ) / 経済-の (economy-GEN) / 発展 (growth) " *technology improvement and economic growth* does not allow us to easily interpret it, which means *((technology improvement) and (economic growth))* or *(technology (improvement and economic growth))*, because bunsetsu-based dependencies do not convey information about left boundary of each noun phrase (Asahara, 2013). This drawback complicates

Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages, pages 108–118,
Seattle, Washington, USA, 18 October 2013.

operating syntactically meaningful units in such applications as statistical machine translation, which needs to recognize syntactic units in building a translation model (e.g. tree-to-string and tree-to-tree) and in preordering source language sentences.

Semantic analysis, such as predicate-argument structure analysis, is usually done as a pipeline process after syntactic analysis (Iida et al., 2011 ; Hayashibe et al., 2011); but in Japanese, the discrepancy between syntactic and semantic units cause difficulties integrating semantic analysis with syntactic analysis.

Our goal is to construct a practical constituent parser that can deal with appropriate grammatical units and output grammatical functions as semi-semantic information, e.g., grammatical or semantic roles of arguments and gapping types of relative clauses. We take an approach to deriving a grammar from manually annotated corpora by training probabilistic models like current statistical constituent parsers of de facto standards (Petrov et al., 2006; Klein et al., 2003 ; Charniak, 2000; Bikel, 2004). We used a constituent-based treebank that Uematsu et al. (2013) converted from an existing bunsetsu-based corpus as a base treebank, and retag the non-terminals and transform the tree structures in described in Section 3. We will present the results of evaluations of the parser trained with the treebank in Section 4, and show some analyses in Section 5.

2 Related work

The number of researches on Japanese constituent-based parser is quite few compared to that of bunsetsu-dependency-based parser. Most of them have been conducted under lexicalized grammatical formalism.

HPSG (Head-driven Phrase Structure Grammar) (Sag et al., 2003) is a representative one. Gunji et al. (1987) proposed JPSG (Japanese Phrase Structure Grammar) that is theoretically precise to handle the free word order problem of Japanese. Nagata et al. (1993) built a spoken-style Japanese grammar and a parser running on it. Siegel et al (2002) constructed a broad-coverage linguistically precise grammar JACY, which integrates semantics, MRS (Minimal Recursion Semantics) (Copestake, 2005). Bond et al. (2008) built a large-scale Japanese treebank Hinoki based on JACY and used it for parser training.

Masuichi et al.(2003) developed a Japanese LFG (Lexicalized-Functional Grammar) (Kaplan et al., 1982) parser whose grammar is sharing the design with six languages. Uematsu et al. (2013) constructed a CCG (Combinatory Categorial Grammar) bank based on the scheme proposed by Bekki (2010), by integrating several corpora including a constituent-based treebank converted from a dependency-base corpus.

These approaches above use a unification-based parser, which offers rich information integrating syntax, semantics and pragmatics, however, generally requires a high computational cost. We aim at constructing a more light-weighted and practical constituent parser, e.g. a PCFG parser, from Penn Treebank style treebank with function labels. Gabbard et al. (2006) introduced function tags by modifying those in Penn Treebank to their parser. Even though Noro et al. (2005) built a Japanese corpus for deriving Japanese CFG, and evaluated its grammar, they did not treat the predicate-argument structure or the distinction of adnominal phrases.

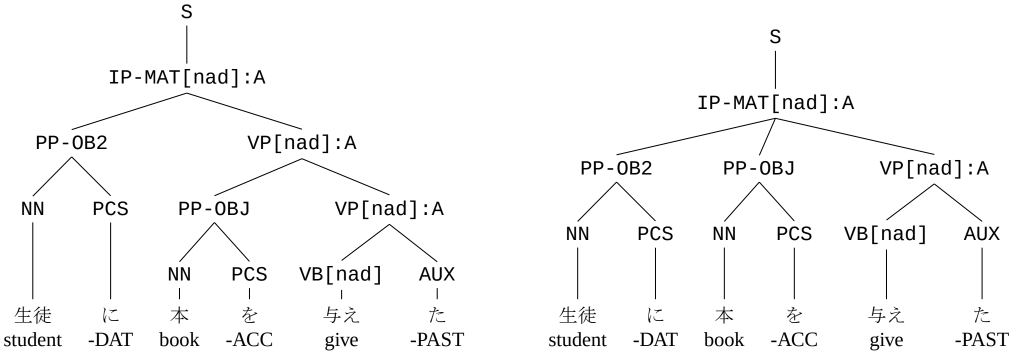
This paper is also closely related to the work of Korean treebank transformations (Choi et al., 2012). Most of the Korean corpus was built using grammatical chunks *eojeols*, which resemble Japanese bunsetsus and consist of content words and morphemes that represent grammatical functions. Choi et al. transformed the eojeol-based structure of Korean treebanks into entity-based to make them more suitable for parser training. We converted an existing bunsetsu-based corpus into a constituent-based one and integrating other information into it for training a parser.

3 Treebank for parser training

In this section, we describe the overview of our treebank for training a parser.

3.1 Construction of a base treebank

Our base treebank is built from a bunsetsu-dependency-based corpus, the Kyoto Corpus (Kurohashi et al., 2003), which is a collection of newspaper articles, that is widely used for training data for Japanese parsers and other applications. We



(I) gave the student a book.

binary tree

n-ary (flattened) tree

Figure 1: Verb Phrase with subcategorization and voice information

NN	General noun
NNP	Proper noun
NPR	Pronoun
NV	Verbal noun
NADJ	Adjective noun
NADV	Adverbial noun (incl. temporal noun)
NNF	Formal noun (general)
NNFV	Formal noun (adverbial)
PX	Prefix
SX	Suffix
NUM	Numeral
CL	Classifier
VB	Verb
ADJ	Adjective
ADNOM	Adnominal adjective
ADV	Adverb
PCS	Case particle
PBD	Binding particle
PADN	Adnominal particle
PCO	Parallel particle
PCJ	Conjunctive particle
PEND	Sentence-ending particle
P	Particle (others)
AUX	Auxiliary verb
CONJ	Conjunction
PNC	Punctuation
PAR	Parenthesis
SYM	Symbol
FIL	Filler

Table 1: Preterminal tags

automatically converted from dependency structure to phrase structure by the previously described method (Uematsu et al., 2013), and conversion errors of structures and tags were manually corrected.

We adopted the annotation schema used in Japanese Keyaki treebank (Butler et al., 2012) and Annotation Manual for the Penn Historical Corpora and the PCEEC (Santorini, 2010) as reference to retag the nonterminals and transform the tree structures.

The original Kyoto Corpus has fine-grained part-of-speech tags, which we converted into simpler preterminal tags shown in Table 1 for training by lookup tables. First the treebank's phrase tags except function tags are assigned by simple CFG rule sets, then, function tags are added by integrating the information from the other resources or manually annotated. We integrate predicate-argument information from the NAIST Text Corpus (NTC) (Iida et al., 2007) into the treebank by automatically converting and adding tag suffixes (e.g. -SBJ, -ARG0 described in section 3.3) to the original tags of the argument phrases. The structure information about coordination and apposition are manually annotated.

3.2 Complementary information

We selectively added the following information as tag suffixes and tested their effectiveness.

Inflection We introduced tag suffixes for inflection as clues to identify the attachment position of the verb and adjective phrases, because Japanese verbs and adjectives have inflections, which depends

(no label)	base form
cont	continuative form
attr	attributive form
neg	negative form
hyp	hypothetical form
imp	imperative form
stem	stem

Table 2: Inflection tag suffixes

on their modifying words and phrases (e.g. noun and verb phrases). Symbols in Table 2 are attached to tags VB, ADJ and AUX, based on their inflection form. The inflection information is propagated to the phrases governing the inflected word as a head. We adopted these symbols from the notation of Japanese CCG described in (Bekki, 2010).

Subcategorization and voice Each verb has a subcategorization frame, which is useful for building verb phrase structure. For instance, 掴む *tsukamu* "grasp" takes two arguments, nominative and accusative cases, 与える *ataeru* "give" takes three arguments: nominative, accusative and dative cases. We also added suffixes to verb tags to denote which arguments they require (n:nominative, a:accusative and d: dative). For instance, the verb 与える "give" takes three arguments (nominative, accusative and dative cases), it is tagged with VB[nad].

We retrieve this information from a Japanese case frame dictionary, Nihongo Goitaikei (Ikehara et al., 1997), which has 14,000 frames for 6,000 verbs and adjectives. As an option, we also added voice information (A:active, P:passive and C:causative) to the verb phrases, because it effectively helps to discriminate cases.

3.3 Annotation schema

We introduce phrase and function tags in Table 3 and use them selectively based on the options described below.

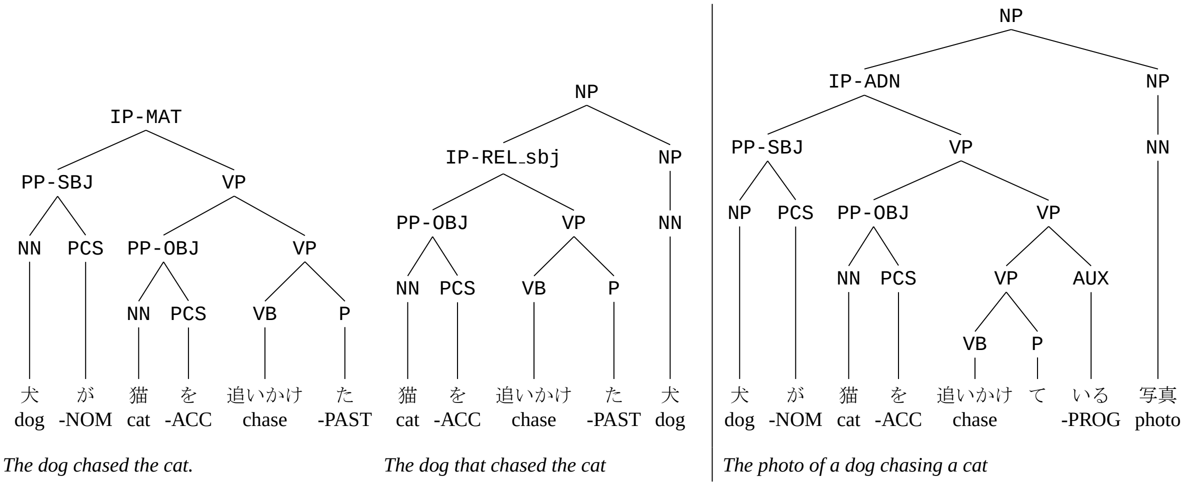
Tree Structure We first built a treebank with binary tree structure (except the root and terminal nodes), because it is comparably easy to convert the existing Japanese dependency-based corpus to it. We converted the dependency-based corpus by a previously described method in (Uematsu et al., 2013). The binary tree's structure has the following characteristics about verb phrase (VP) and postposition phrase (PP): VP from the same bunsetsu is a left-branching subtree and the PP-VP structure (roughly corresponding to the argument-predicate structure) is a right-branching subtree. Pure binary trees tend to be very deep and difficult to annotate and interpret by humans. We also built an n-ary tree version by flattening these structures.

NP	Noun phrase
PP	Postposition phrase
VP	Verb phrase
ADJP	Adjective phrase
ADVP	Adverbial phrase
CONJP	Conjunction phrase
S	Sentence (=root)
IP	Inflectional phrase
IP-MAT	Matrix clause
IP-ADV	Adverb clause
IP-REL	Gapping relative clause
IP-ADN	Non-gapping adnominal clause
CP	Complementizer phrase
CP-THT	Sentential complement
Function tags	
semantic role for mandatory argument (gap notation)	
-ARG0 (_arg0)	
-ARG1 (_arg1)	
-ARG2 (_arg2)	
grammatical role for mandatory argument (gap notation)	
-SBJ (_sbj)	Subjective case
-OBJ (_obj)	Objective case
-OB2 (_ob2)	Indirect object case
arbitrary argument	
-TMP	Temporal case
-LOC	Locative case
-COORD	Coordination (for n-ary)
-NCOORD	Left branch of NP coord. (for binary)
-VCOORD	Left branch of VP coord. (for binary)
-APPOS	Apposition
-QUE	Question

Table 3: Phrase tags

The predicate-argument structure, which is usually represented by PPs and a VP in the treebank, particularly tends to be deep in binary trees based on the number of arguments. To flatten the structure, we remove the internal VP nodes by intermediately re-attaching all of the argument PPs to the VP that dominates the predicate. Figure 1 shows an example of flattening the PP-VP structure.

For noun phrases, since compound nouns and numerals cause deep hierarchy, the structure that includes them is flattened under the parent NP. The coordinating structure is preserved, and each NP element of the coordination is flattened



The dog chased the cat.

The dog that chased the cat

The photo of a dog chasing a cat

Figure 2: Leftmost tree shows annotation of grammatical roles in a basic inflectional phrase. Right two trees show examples of adnominal phrases.

Predicates and arguments The predicate's argument is basically marked with particles, which represent cases in Japanese; thus, they are represented as a postpositional phrase, which is composed of a noun phrase and particles. The leftmost tree in Figure 2 is an example of the parse result of the following sentence: 犬-が *inu-ga* "dog-NOM" 猫-を *neko-o* "cat-ACC" 追いかけた *oikaketa* "chased" (*The dog chased the cat.*)

We annotated predicate arguments by two different schemes (different tag sets) in our treebank: grammatical roles and semantic roles. In using a tag set based on grammatical roles, the arguments are assigned with the suffixes based on their syntactic roles in the sentence, like Penn Treebank: SBJ (subject), OBJ (direct object), and OB2 (indirect object). Figure 2 is annotated by this scheme.

Alternatively, the arguments are labeled based on their semantic roles from case frame of predicates, like PropBank (Palmer et al., 2005): ARG0, ARG1 and ARG2. These arguments are annotated by converting semantic roles defined in the case frame dictionary Goitaikei into simple labels, the labels are not influenced by case alternation.

In both annotation schemes, we also annotated two types of arbitrary arguments with semantic role labels: LOC (locative) and TMP (temporal), which can be assigned consistently and are useful for various applications.

Adnominal clauses Clauses modifying noun phrases are divided into two types: (gapping) relative and non-gapping adnominal clauses. Relative clauses are denoted by adding function tag -REL to phrase tag IP. Such a gap is directly attached to IP-REL tag as a suffix consisting of an underscore and small letters in our treebank, e.g., IP-REL_sbj for a subject-gap relative clause, so that the parser can learn the type of gap simultaneously, unlike the Penn Treebank style, where gaps are marked as trace '*T*'. For instance, note the structure of the following noun phrase, which is shown in the middle tree in Figure 2: 猫-を *neko-o* "cat-ACC" 追いかけた *oikake-ta* "to chase" 犬 *inu* "dog" "neko-o (cat-ACC) oikaketa (chase) inu" (*The dog that chased the cat.*). We also adopt another type of gap notation that resembles the predicate-argument structure: semantic role notation. In the example above, tag IP-REL_arg0 is attached to the relative clause instead.

We attach tag IP-ADN to another type of adnominal clauses, which has no gap, the modified noun phrase is not an argument of the predicate in the adnominal clause. The rightmost in Figure 2 is an example of a non-gapping clause: 犬-が *inu-ga* "dog-NOM" 猫-を *neko-o* "cat-ACC" 追いかけている *oikake-teiru* "chasing" 写真 *shashin* "photo" (*A photo of a dog chasing a cat.*), where there is no predicate-argument relation between the verb 追いかける *chase* and the noun 写真 *photo*.

Coordination and apposition The notation of such parallel structure as coordination and apposition differs based on the type of tree structure. For binary trees, the coordination is represented by a left-branching tree, which is a conjunction or a conjunction particle that first joined a left hand constituent; the phrase is marked as a modifier consisting of coordination (`-NCOORD` and `-VCOORD` for NP and VP coordinations), as shown on the left side of Figure 3. On the other hand, in n-ary trees, all the coordination elements and conjunctions are aligned flatly under the parent phrase with suffix `-COORD`. The apposition is represented in the same way using tag `-APPOS` instead.

Phrase and sentential elements Since predicate arguments are often omitted in Japanese, discrimination between the fragment of larger phrases and sentential elements is not clear. In treebank, we employ `IP` and `CP` tags for inflectional and complementizer phrases, assuming that tags with function tag suffixes to the phrase correspond to the maximum projection of the predicate (verb or adjective). The matrix phrase and the adverbial phrase have `IP-MAT` and `IP-ADV` tags respectively. This annotation schema is adopted based on the Penn Historical Corpora (Santorini, 2010) and Japanese Keyaki treebank (Butler et al., 2012) as previously described, while IP in our treebank is not so flat as them.

Such sentential complements as that-clauses in English are tagged with `CP-THT`. In other words, the sentential elements, which are annotated with SBAR, S, and trace *T* in the Penn Treebank, are tagged with `CP` or `IP` in our treebank.

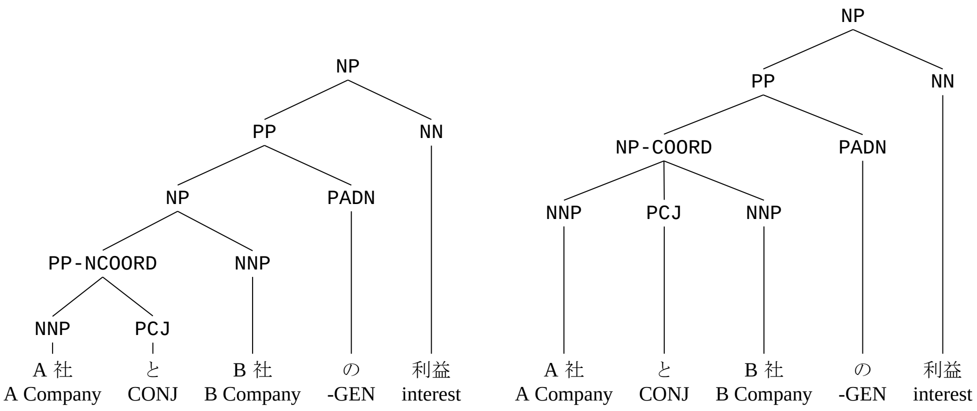
4 Evaluation

The original Kyoto Corpus has 38,400 sentences and they were automatically converted to constituent structures. The function tags are also added to the corpus by integrating predicate-argument information in the NAIST Text corpus. Since the conversion contains errors of structures and tags, about half of them were manually checked to avoid the effects of the conversion errors.

We evaluated our treebank's effectiveness for parser training with 18,640 sentences, which were divided into three sets: 14,895 sentences for a train-

Tag set	LF$_1$	Comp	UF$_1$	Comp
binary tree				
Base	88.4	34.0	89.6	37.9
Base_inf	88.5⋆	33.5	90.0⋆	39.3
Full$_{\mathbf{sr}}$	80.7	13.6	88.4	35.9
Full$_{\mathbf{sr}}$**_inf**	**81.1**⋆	**15.5** ⋆	**88.7**⋆	**36.9**
Full$_{\mathbf{sr}}$**_lex**	79.8⋆	13.1	87.7⋆	34.3
Full$_{\mathbf{sr}}$**_vsub**	80.3⋆	12.5	87.9⋆	35.1
Full$_{\mathbf{sr}}$**_vsub_alt**	78.6⋆	13.3	86.7⋆	32.5⋆
Full$_{\mathbf{gr}}$	81.0	**15.6**	88.5	**37.3**
Full$_{\mathbf{gr}}$**_inf**	**81.3**⋆	15.3	**88.8**	37.2
Full$_{\mathbf{gr}}$**_lex**	80.3⋆	14.2	87.9⋆	33.6⋆
Full$_{\mathbf{gr}}$**_vsub**	81.2	15.5	88.5	35.2
Full$_{\mathbf{gr}}$**_vsub_alt**	77.9⋆	11.7⋆	86.0⋆	29.9⋆
n-ary tree				
Full$_{\mathbf{sr}}$	76.7	11.4	85.3	28.0
Full$_{\mathbf{sr}}$**_inf**	**76.9**	**11.6**	**85.4**	**28.7**
Full$_{\mathbf{sr}}$**_lex**	76.5	11.1	84.7⋆	27.9
Full$_{\mathbf{sr}}$**_vsub**	76.5	10.8	84.9⋆	26.2
Full$_{\mathbf{sr}}$**_vsub_alt**	76.6	11.0	84.8⋆	27.2
Full$_{\mathbf{gr}}$	77.2	**13.2**	85.3	**29.2**
Full$_{\mathbf{gr}}$**_inf**	77.4	12.0⋆	**85.5**	28.3
Full$_{\mathbf{gr}}$**_lex**	**77.6**	12.2⋆	85.0	28.5
Full$_{\mathbf{gr}}$**_vsub**	77.1	12.7⋆	84.8⋆	28.8
Full$_{\mathbf{gr}}$**_vsub_alt**	76.9	12.2⋆	84.7⋆	26.3⋆

Table 4: Parse results displayed by labeled and unlabeled F$_1$ metrics and proportion of sentences completely matching gold standard (**Comp**). **Base** contains only basic tags, not grammatical function tags. Figures with '⋆' indicate statistically significant differences ($\alpha = 0.05$) from the results without complementary information, i.e., **Full**$_{\mathbf{sr}}$ or **Full**$_{\mathbf{gr}}$.



the interests of A Company and B Company

Figure 3: Noun phrase coordination

tag set	**UAS**
binary tree	
Base	89.1
Base_inf	89.4
Full$_{\mathbf{sr}}$	87.9
Full$_{\mathbf{sr}}$**_inf**	88.3
Full$_{\mathbf{gr}}$	88.0
Full$_{\mathbf{gr}}$**_inf**	88.5⋆
n-ary (flattened) tree	
Full$_{\mathbf{sr}}$	82.8
Full$_{\mathbf{sr}}$**_inf**	83.3
Full$_{\mathbf{gr}}$	82.9
Full$_{\mathbf{gr}}$**_inf**	83.0

Table 5: Dependency accuracies of the results converted into bunsetsu dependencies.

ing set, 1,860 sentences for a test set, and the remainder for a development set.

The basic evaluations were under the condition of using the original tag sets: the basic set **Base**, which contains all the preterminal tags in Table 1 and the phrase tags in Table Table 3, except the IP and CP tags, and the full set **Full**, which has **Base** + IP, CP tags, and all the function tags. The basic set **Base** is provided to evaluate the constituent parser performance in case that we need better performance at the cost of limiting the information.

We used two types of function tag sets: **Full** $_{\mathbf{sr}}$ for semantic roles and **Full** $_{\mathbf{gr}}$ for grammatical roles.

We added the following complementary information to the tags and named the new tag sets **Base** or **Full** and suffix:

_inf: add inflection information to the POS tag (verbs, adjectives, and auxiliary verbs) and the phrase tags (Table 2).

_lex: lexicalize the closed words, i.e., auxiliary verbs and particles.

_vsub: add verb subcategorization to the verb and verb phrase tags.

_vsub_alt: add verb subcategorization and case alternation to the verb and verb phrase tags.

In comparing the system output with the gold standard, we remove the complementary information to ignore different level of annotation, thus, we do not discriminate between `VB[na]` and `VB[nad]` for example.

We used the Berkeley parser (Petrov et al., 2006) for our evaluation and trained with six iterations for latent annotations. In training the n-ary trees, we used a default Markovization parameter ($h = 0, v = 1$), because the parser performed the best with the development set.

Table 4 shows the parsing results of the test sets. On the whole, the binary tree outperformed the n-ary tree. This indicates that the binary tree structure was converted from bunsetsu-based dependencies, whose characteristics are described in Section 3.3, and is better for parser training than the partially flattened structure.

As for additional information, the inflection suffixes slightly improved the F_1-metrics. This is mainly because the inflection information gives the category of the attached phrase (e.g., the attributive form for noun phrases). The others did not provide any improvement, even though we expected the subcategorization and case alternation information to help the parser detect and discriminate the grammatical roles, probably because we simply introduced the information by concatenating the suffixes to the base tags to adapt an off-the-shelf parser in our evaluation. For instance, `VB[n]` and `VB[na]` are recognized as entirely independent categories; a sophisticated model, which can treat them hierarchically, would improve the performance.

For comparison with a bunsetsu-based dependency parser, we convert the parser output into unlabeled bunsetsu dependencies by the following simple way. We first extract all bunsetsu chunks in a sentence and find a minimum phrase including each bunsetsu chunk from a constituent structure. For each pair of bunsetsus having a common parent phrase, we add a dependency from the left bunsetsu to the right one, since Japanese is a head-final language.

The unlabeled attachment scores of the converted dependencies are shown as the accuracies in Table 5, since most bunsetsu-based dependency parsers output only unlabeled structure.

The **Base_inf** results are comparable with the bunsetsu-dependency results (90.46%) over the same corpus (Kudo et al., 2002)[1], which has only the same level of information. Constituent parsing with treebank almost matched the current bunsetsu parsing.

5 Analysis

In this section, we analyze the error of parse results from the point of view of the discrimination of grammatical and semantic roles, adnominal clause and coordination.

Grammatical and semantic roles Predicate arguments usually appeared as PP, which is composed of noun phrases and particles. We focus on PPs with function labels. Table 6 shows the PP results with

tag	P	R	F_1
PP-ARG0	64.9	75.0	69.6
PP-ARG1	70.6	80.1	75.1
PP-ARG2	60.3	68.5	64.1
PP-TMP	40.1	43.6	41.8
PP-LOC	23.8	17.2	20.0

tag	P	R	F_1
PP-SBJ	69.6	81.5	75.1
PP-OBJ	72.6	83.5	77.7
PP-OB2	63.6	71.4	67.3
PP-TMP	45.0	48.0	46.5
PP-LOC	21.3	15.9	18.2

Table 6: Discrimination of semantic role and grammatical role labels (upper: semantic roles, lower: grammatical role)

system \ gold	PP-SBJ	PP-OBJ	PP-OB2
PP-SBJ	*74.9	6.5	2.3
PP-OBJ	5.8	*80.1	0.5
PP-OB2	1.7	0.3	*68.5
PP-TMP	0.2	0.0	0.5
PP-LOC	0.2	0.0	0.4
PP	6.5	2.0	16.8
other labels	0.5	0.2	0.3
no span	10.2	10.9	11.0

system \ gold	PP-TMP	PP-LOC
PP-SBJ	4.7	4.1
PP-OBJ	0.0	0.0
PP-OB2	6.0	13.8
PP-TMP	*43.6	2.8
PP-LOC	2.0	*17.2
PP	37.6	49.7
other labels	1.4	5.0
no span	4.7	7.4

Table 7: Confusion matrix for grammatical role labels (recall). Figures with '*' indicate recall.(binary tree, $\textbf{Full}_{\textbf{gr}}$)

tag	P	R	F_1
IP-REL_sbj	48.4	54.3	51.1
IP-REL_obj	27.8	22.7	24.9
IP-REL_ob2	17.2	29.4	21.7
IP-ADN	50.9	55.4	53.1
CP-THT	66.1	66.6	66.3

Table 8: Results of adnominal phrase and sentential element (binary tree, $\textbf{Full}_{\textbf{gr}}$)

[1] The division for the training and test sets is different.

grammatical and semantic labels under the $\mathbf{Full_{sr}}$ and $\mathbf{Full_{gr}}$ conditions respectively.

The precision and the recall of mandatory arguments did not reach a high level. The results are related to predicate argument structure analysis in Japanese. But, they cannot be directly compared, because the parser in this evaluation must output a correct target phrase and select it as an argument, although most researches select a word using a gold standard parse tree. Hayashibe et al. (2011) reported the best precision of ARG0 discrimination to be 88.42 % [2], which is the selection results from the candidate nouns using the gold standard parse tree of NTC. If the cases where the correct candidates did not appear in the parser results are excluded (10.8 %), the precision is 72.7 %. The main remaining error is to label to non-argument PP with suffix -ARG0 (17.0%), thus, we must restrain the overlabeling to improve the precision.

The discrimination of grammatical role is higher than that of semantic role, which is more directly estimated by case particles following the noun phrases. The confusion matrix for the recall in Table 7 shows main problem is parse error, where correct phrase span does not exist (no span), and marks 10-11%. The second major error is discrimination from bare PPs (PPs without suffixes), mainly because the clues to judge whether the arguments are mandatory or arbitrary lack in the treebank. Since even the mandatory arguments are often omitted in Japanese, it is not facilitate to identify arguments of predicates by using only syntactic information.

Adnominal phrases We need to discriminate between two types of adnominal phrases as described in Section 3.3: IP-REL and IP-ADN. Table 8 shows the discrimination results of the adnominal phrase types. The difference between IP-REL (gapped relative clauses) and IP-ADN is closely related to the discrimination of the grammatical role: whether the antecedent is the argument of the head predicate of the relative clause.

Table 8 shows the discrimination results of the adnominal phrases. The results indicate the difficulties of discriminating the type of gaps of relative clause IP-REL. The confusion matrix in Table 9 shows that the discrimination between gaps and non-gaps, i.e., IP-REL and IP-ADN, is moderate as for IP-REL_sbj and IP-REL_obj. However, IP-REL_ob2 is hardly recognized, because it is difficult to determine whether the antecedent, which is marked with particle 'ni', is a mandatory argument (IP-REL_ob2) or not (IP-ADN). Increasing training samples would improve the discrimination, since there are only 290 IP-REL_ob2 tags for 8,100 IP-ADN tags in the training set.

Naturally discrimination only by syntactic information has limitation; this baseline can be improved by incorporating semantic information.

Coordination Figure 10 shows the coordination results, which are considered the baseline for only using syntactic information. Improvement is possible by incorporating semantic information, since the disambiguation of coordination structure essentially needs semantic information.

6 Conclusion

We constructed a Japanese constituent-based parser to be released from the constraints of bunsetsu-based analysis to simplify the integration of syntactic and semantic analysis. Our evaluation results indicate that the basic performance of the parser trained with the treebank almost equals bunsetsus-based parsers and has the potential to supply detailed syntactic information by grammatical function labels for semantic analysis, such as predicate-argument structure analysis.

Future work will be to refine the annotation scheme to improve parser performance and to evaluate parser results by adapting them to such NLP applications as machine translation.

Acknowledgments

We would like to thank Associate Professor Yusuke Miyao, Associate Professor Takuya Matsuzaki of National Institute of Informatics and Sumire Uematsu of the University of Tokyo for providing us the language resources and giving us valuable suggestions.

[2] The figure is calculated only for the arguments that appear as the dependents of predicates, excluding the omitted arguments.

system \ gold	IP-REL_sbj	IP-REL_obj
IP-REL_sbj	*55.0	30.3
IP-REL_obj	8.5	*33.3
IP-REL_ob2	0.5	0.0
IP-ADN	10.0	9.0
IP-ADV	0.3	0.0
VP	8.5	7.6
other labels	1.2	6.2
no span	16.0	13.6
system \ gold	IP-REL_ob2	IP-ADN
IP-REL_sbj	29.4	7.5
IP-REL_obj	0.0	0.6
IP-REL_ob2	*11.8	0.0
IP-ADN	23.5	*57.3
IP-ADV	0.5	0.3
VP	17.6	9.3
other labels	5.4	3.0
no span	11.8	22.0

Table 9: Confusion matrix for adnominal phrases (recall). Figures with '*' indicate recall.(binary tree, $\mathbf{Full_{gr}}$)

tag	P	R	F_1
NP-COORD	62.6	60.7	61.6
VP-COORD	57.6	50.0	53.5
NP-APPOS	46.0	40.0	42.8

Table 10: Results of coordination and apposition (binary tree, $\mathbf{Full_{gr}}$)

References

Masayuki Asahara. 2013. Comparison of syntactic dependency annotation schemata . In *Proceedings of the 3rd Japanese Corpus Linguistics Workshop*, In Japanese.

Daisuke Bekki. 2010. Formal theory of Japanese syntax. Kuroshio Shuppan, In Japanese.

Daniel M. Bikel. 2004. A distributional analysis of a lexicalized statistical parsing model. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP 2004)*, Vol.4, pp. 182–189.

Francis Bond, Sanae Fujita and Takaaki Tanaka. 2008. The Hinoki syntactic and semantic treebank of Japanese. In *Journal of Language Resources and Evaluation*, Vol.42, No. 2, pp. 243–251

Alastair Butler, Zhu Hong, Tomoko Hotta, Ruriko Otomo, Kei Yoshimoto and Zhen Zhou. 2012. Keyaki Treebank: phrase structure with functional information for Japanese. In *Proceedings of Text Annotation Workshop*.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, (NAACL 2000)*, pp. 132–139.

DongHyun Choi, Jungyeul Park and Key-Sun Choi. 2012. Korean treebank transformation for parser training. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pp. 78-88.

Ann Copestake, Dan Flickinger, Carl Pollard and Ivan A. Sag. 2005. Minimal recursion semantics: an introduction. *Research on Language and Computation*, Vol. 3, No. 4, pp. 281-332.

Ryan Gabbard, Mitchell Marcus and Seth Kulick. 2006. Fully parsing the Penn Treebank. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL 2006)*, pp. 184–191.

Takao Gunji. 1987 Japanese phrase structure grammar: a unification-based approach. D.Reidel.

Yuta Hayashibe, Mamoru Komachi and Yujzi Matsumoto. 2011. Japanese predicate argument structure analysis exploiting argument position and type. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pp. 201-209.

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*, pp. 132–139.

Ryu Iida, Massimo Poesio. 2011. A cross-lingual ILP solution to zero anaphora resolution. In *Proceedings* *of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pp. 804-813.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Kentaro Ogura, Yoshifumi Ooyama and Yoshihiko Hayashi. 1998. Nihongo Goitaikei. Iwanami Shoten, In Japanese.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: a formal system for grammatical representation. In *the Mental Representation of Grammatical Relations* (Joan Bresnan ed.), pp. 173–281. The MIT Press.

Daisuke Kawahara and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL 2006)*, pp. 176–183.

Dan Klein and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language processing. *Advances in Neural Information Processing Systems*, 15:3–10.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL-2002)*, Volume 20, pp. 1–7.

Sadao Kurohashi and Makoto Nagao. 2003. Building a Japanese parsed corpus – while improving the parsing system. In Abeille (ed.), *Treebanks: Building and using parsed corpora*, Chap. 14, pp. 249–260. Kluwer Academic Publishers.

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. In *Journal of Computational Linguistics*. Vol.19, No.2, pp. 313–330.

Hiroshi Masuichi, Tomoko Okuma, Hiroki Yoshimura and Yasunari Harada. 2003 Japanese parser on the basis of the Lexical-Functional Grammar formalism and its evaluation. In *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation (PACLIC 17)*, pp. 298-309.

Masaaki Nagata and Tsuyoshi Morimoto,. 1993. A unification-based Japanese parser for speech-to-speech translation. In *IEICE Transaction on Information and Systems*. Vol.E76-D, No.1, pp. 51–61.

Tomoya Noro, Taiichi Hashimoto, Takenobu Tokunaga and Hotsumi Tanaka. 2005. Building a large-scale Japanese syntactically annotated corpus for deriving a CFG. in *Proceedings of Symposium on Large-Scale Knowledge Resources (LKR2005)*, pp..159 – 162.

Matha Palmer, Daniel Gildea and Paul Kingsbury. 2005. The Proposition Bank: n annotated corpus of semantic roles. *Computational Linguistics*, Vol.31 No. 1, pp. 71–106.

Slav Petrov, Leon Barrett, Romain Thibaux and Dan Klein.. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pp. 433-440.

Ivan A. Sag, Thomas Wasow and Emily M. Bender,. 2003. Syntactic theory: a formal introduction. *2nd Edition*, *CSLI Publications*.

Beatrice Santorini. 2010. Annotation manual for the Penn Historical Corpora and the PCEEC (Release 2). Department of Linguistics, University of Pennsylvania.

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 at the 19th International Conference on Computational Linguistics*, Vol. 12, pp. 1–8.

Sumire Uematsu, Takuya Matsuzaki, Hiroaki Hanaoka, Yusuke Miyao and Hideki Mima. 2013. Integrating multiple dependency corpora for inducing wide-coverage Japanese CCG resources. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pp. 1042–1051.