

Japanese Semantic Role Labeling with Hierarchical Tag Context Trees

Yasuhiro Ishihara^(✉) and Koichi Takeuchi

Graduate School of Natural Science and Technology,
Okayama University, Okayama, Japan
{ishihara,koichi}@cl.cs.okayama-u.ac.jp

Abstract. In this paper we describe that the hierarchical tag context tree (HTCT) approach improves the accuracy of semantic role labeling on Japanese text. In Japanese language there are functional multiword expressions such as *no-tame-ni* and *yotte* that have potential to designate semantic relations between a predicate and its arguments. Since these expressions come to the end part of each argument, the performance of the CRF-based semantic role labeler can be improved by taking into account the last morphemes of each argument as features. We apply our proposed system to the annotated corpus of semantic role labels on a balanced Japanese corpus. The experimental results show that the CRF-based labeler with features extracted by HTCT approach outperforms the normal CRF-based labeler.

Keywords: Hierarchical Tag Context Trees · Semantic role labeling · CRFs

1 Background Issues

Analyzing semantic roles of arguments for a predicate must be a fundamental technology to capture deeper semantic relations between sentences. Since annotated corpora of semantic role labels (i.e., SRLs) and their frames are well developed in English, e.g., FrameNet [1] and PropBank [2], a lot of SRL detection systems have been developed mainly on English language [3–5]. In contrast to this, the most of the recent annotated corpora of predicate-argument structure in Japanese [6–8] are not on the level of semantic roles but on the level of surface case marker level.

In this situation, recently several language resources such as Japanese FrameNet [9] and Predicate Thesaurus (PT) [10] containing annotated semantic role information are constructed on Balanced Corpus of Contemporary Written Japanese (BCCWJ) [11]. Since the balanced corpus contains various text genres, the annotated data of SRLs on BCCWJ must be a profitable language resource for constructing a robust SRLer for Japanese. Currently the annotated corpus

[*Reason* 悪天候 のため-に] [*Theme* フライト-は] 中止-され-た
 akutenkou no-tame-ni furaito-ha chuushi-sa-re-ta
 bad weather because of flight-TOP was cancelled
 The flight was cancelled because of the bad weather.

Fig. 1. An example of multiword expression *no-tame-ni*.

based on PT is available¹ thus we use PT-based annotated corpus as a gold standard of SRLs that contains 72 types of SRLs².

The previous work on constructing English semantic role labeling system [3, 4] reveals that syntactic information is indispensable feature for recognizing SRLs, however, Japanese case markers, which are main clues of syntactic structure, do not have enough variety compared with prepositions in English; for example, English prepositions *in*, *at*, *with*, *by* can be mapped to a Japanese case marker *de*. Thus it must not be possible to apply the approaches of English SRL systems to a Japanese SRL system.

Besides case markers, functional multiword expressions (e.g., *no-tame-ni* (because of), *to-shi-te* (as), and so on) can be clues to estimate semantic relation types between a predicate and its arguments. The example of *no-tame-ni* (because of) is shown in Fig. 1.

In Fig. 1 the brackets indicates arguments for the predicate *chuushi-sa-re-ta*, and *Reason* and *Theme* are SRLs in Fig. 1. Functional multiword *no-tame-ni* (because of) indicates that the SRL of the first argument must be *Reason*. Functional multiwords are manually collected and distributed as a dictionary Tsutsuji³, however,

- (1) Japanese dependency parser (e.g., cabocha+mecab) does not detect the functional multiwords; and then the functional multiwords are separated into morphemes and are sometimes wrongly POS-annotated depending on the context, and
- (2) even though the functional multiword dictionary is available, there is still possibility to exist unrecognized functional multiwords.

Therefore we propose an approach to improve performance of SRL system by capturing the functional multiword expressions in each argument. In this paper, we apply hierarchical tag context tree model (HTCT) [13] that can extract automatically effective sequences of morphemes and/or POSes. The extracted sequences of morphemes and/or POSes are applied to a CRF-based SRL system as features. In the experimental results we show that the CRF with HTCT system outperformed a simple CRF-based SRL system.

¹ <http://pth.cl.cs.okayama-u.ac.jp>.

² The EDR corpus [12] also contains SRLs on Japanese texts, however, the texts are not balanced, thus we select PT corpus.

³ <http://kotoba.nuee.nagoya-u.ac.jp/tsutsuji/>.

2 Hierarchical Tag Context Tree Approach for Extracting Effective Sequences

The basic idea of the HTCT-based proposed approach is almost the same as an approach to construct a context tree from input sequences. The context tree is a framework to capture frequent sequences, and the characteristics of HTCT is that a context tree is constructed not only for input words but also for tags (e.g., POSes) taking into account a hierarchy of tags [13]. In the rest of the section, we describe the information-theoretical framework of how we find effective sequences from input sequences with hierarchical tags, and then describe how we adapt the HTCT framework for finding effective feature sequences of SRLs.

The key issue of constructing a context tree from input sequences is to define a criteria where a new context should be added to a tree or not. Assuming the situation to add a new tag b to a context sequence s at a leaves of a context tree, we define $\delta(sb)$ as an evaluation measure that indicates the gain of expanding the context s to a new context sb on the basis of Kullback-Leibler divergence between the probability distributions given the context sequences sb and s . The equation is shown in Eq. (1).

$$\begin{aligned}\delta(sb) &= n(sb) \sum_{a \in A} \frac{n(a|sb)}{n(sb)} \log \frac{P(a|sb)}{P(a|s)} \\ &= n(sb) \sum_{a \in A} P(a|sb) \log \frac{P(a|sb)}{P(a|s)} \\ &= n(sb) D_{KL}(P(\cdot|sb), P(\cdot|s))\end{aligned}\tag{1}$$

Where $n(sb)$, $P(\cdot|sb)$ denote the number of occurrence of sb and the conditional probability of a target given by sb , respectively. The idea of the evaluation measure is that the new tag b should be added to the tree node s when the new context sb gives enough information gain compared with the base context s . Then a new tag will be added when the measure $\delta(sb)$ is larger than the threshold we define in Sect. 3.

The algorithm of construction of a context tree for input sequences is processed by a greedy algorithm, i.e., the possibility of adding new tags are evaluated only on the leaves of the context tree that have already been fixed. This situation is shown in Fig. 2, where each node indicates a context tag sequence and each arrow indicates a tag added to a leaf of the context tree; the first node ϵ denotes an empty context; the dashed nodes and arrows denote not generated nodes and arrows, respectively.

In Fig. 2 once a new arrow r is rejected to add the context s by the above evaluation using Eq. (1), our approach will not take the context sr into account any more. This indicates that even if the longer context src was an effective context sequence, our approach would not take the context src because the context sr was not registered in the base context tree.

In the above description, tags are flat, and now we incorporate tags that have a hierarchical structure. Since the unit of the tags are morphemes in Japanese,

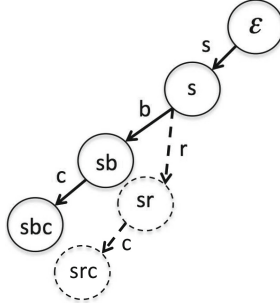


Fig. 2. Making a context tree with a greedy algorithm.

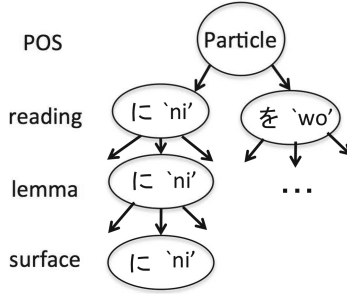


Fig. 3. Hierarchical structure of tags.

we assume that four-layer is-a hierarchy, i.e., part-of-speech, reading, lemma and surface are annotated to all the morphemes in the corpus (see Sect. 3).

Figure 3 shows an example of a hierarchy of Japanese case marker ‘ni’ whose POS is Particle, reading is ‘ni’, lemma is ‘ni’ and surface is ‘ni’. Since the hierarchical structure expresses abstraction levels of a morpheme, our approach takes only one element from the four hierarchical levels for a morpheme on the basis of Eq. 1. This indicates that the elements (i.e., tags) in a context sequence contain surface expressions, lemmas, readings and POSes of morphemes, and then our approach takes the best tag from all of the possible morphemes with hierarchical tags in extending a context tag sequence.

In the above explanation, we describe the theoretical framework of HTCT approach, and now we describe how we adapt the HTCT framework for finding effective tag sequences for SRLs.

As described in Sect. 1, case markers and multiwords i.e., functional morpheme sequences at the end of arguments can be effective for disambiguation of SRLs. Thus we apply the HTCT approach to extraction of effective tag sequences of ending morphemes in each argument of sentences.

To realize this adaptation, we prepare an annotated corpus of SRLs such as Fig. 1 and apply the HTCT approach to the annotated corpus by the following modification steps.

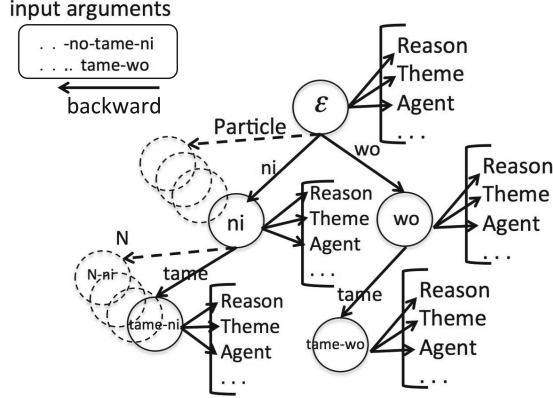


Fig. 4. Construction of an HTCT for finding effective tag sequence of SRLs.

- (1) An HTCT is constructed not for morphemes but for SRLs, and
- (2) An HTCT is constructed backward from the last morpheme of each argument.

In the modification step (1), we define the target $a \in A$ in Eq. 1 as SRLs. Thus we need to evaluate the conditional probability of SRLs given by various context tags such as $P(\text{Theme}|\cdot)$ and $P(\text{Reason}|\cdot)$ using the annotated corpus.

To describe the details of the modification (2), Fig. 4 shows the situation of constructing an HTCT for *no-tame-ni* and *tame-wo*. The left bracket denotes the target SRLs and each node has a conditional probability of SRLs given the context tag sequence. For example, at the top node ϵ , the conditional probabilities are defined as $P(\text{Theme})$ and $P(\text{Agent})$ which indicate no context tags, while at the second node, e.g., *ni*, the conditional probability is defined as $P(\text{Theme}|\text{ni})$. Since these conditional probabilities of SRLs are used in the evaluation measure of Eq. (1), the extracted context tag tree can be effective for predicting SRLs.

Figure 4 shows that the context tree is constructed backward, i.e., the first level of the context tags are evaluated on the morphemes *ni* and *wo*, and then the second level of the context is evaluated on *tame*. Since a leaf node of the context tree will be extended if the information gain of the new node is enough large, the context tag tree is expected to capture characteristic tag sequences as long as possible.

In every step of adding a new tag to a leaf of context tree, the proposed approach takes into account hierarchical tags for the target morphemes. Figure 4 shows the case that the surface level, i.e., *ni* and *wo* are selected at the second nodes of the context tree. Since hierarchical consistency will be kept in the context tree, if the POS level, i.e., *PARTICLE* were selected at the second nodes, the surface level nodes *ni* and *wo* might be merged into *PARTICLE* node; and then the third node *tame* would be evaluated in the context of *PARTICLE-tame*.

Table 1. Top 10 SRLs in the annotated corpus

Name of SRL	Freq. of SRLs
Theme	1391
Agent	567
Experiencer	242
Time	233
Manner	223
Goal	178
Adverbial	165
Reason	161
Modificant	152
Method	140
Total	4844

3 Experiments of Semantic Role Labeling and Discussions

3.1 Experimental Set Up

The PT corpus contains 2662 annotated sentences and head verbs and their arguments (4844) are annotated with the 62 types of SRLs in the sentences of BCCWJ. In the corpus all of the sentences are broken down to morphemes, and then SRLs are annotated to the morphemes with IOB2 tag format for arguments. The statistics of the top 10 SRLs are shown in Table 1⁴. The top two frequent SRLs are *Theme* and *Agent* that are the same as an English SRL annotated corpus PropBank [5].

Since each chunk, i.e., argument is annotated, the proposed CRF-based SRL system recognize (1) boundary of each argument and (2) SRL for each argument. The performance of the system is evaluated by precision, recall and f-measure. Let an output of the system be correct if the system correctly detect both a boundary and its SRL.

To evaluate the SRL system, we divide the TP corpus in half, i.e., training corpus and test corpus. In the both corpora all of the morphemes are correctly annotated with surface, lemma, reading, and POS on the basis of UniDic [11].

3.2 CRF Model and CRF with HTCT Model

We execute three types of experiments. The first is applying normal CRFs⁵ with taking into account a fixed number of morphemes at the end of each argument (Table 7); the second is CRFs with variable length features from the HTCT

⁴ See more details of the annotated corpus at <http://pth.cl.cs.okayama-u.ac.jp>.

⁵ We use CRF++ <http://crfpp.googlecode.com/svn/trunk/doc/index.html?source=navbar>.

Table 2. Base features of CRF

No.	Description of feature
1	Surface of the target morpheme
2	Lemma of the target morpheme
3	Reading of the target morpheme
4	POS of the target morpheme
5	Surface of the final Noun morpheme in the argument
6	Lemma of the final Noun morpheme in the argument
7	Case marker of the argument
8	Lemma of the head verb

Table 3. Contextual features of CRF

No.	Description of feature
t1	Surface of the next morpheme
t2	Surface of the previous morpheme
t3	Reading of the next morpheme
t4	Reading of the previous morpheme
t5	POS of the next morpheme
t6	POS of the previous morpheme

model⁶; and the third is CRFs with the first experiments’ settings and the features extracted from the HTCT model. In the rest of the section, we describe how we utilize the HTCT to a CRF model as well as the details of the features in each CRF model.

We prepare three types of CRF-based models with different features; they are (1) normal CRF (denoted as CRF), (2) CRF taking into account the features of the last a few morphemes in arguments (denoted as CRF+2suf and CRF+3suf, respectively), and (3) CRF that extends the second model by adding the combinations of features of the last two morphemes of arguments (denoted as CRF+3suf+c).

Table 2 shows the base features of the CRF model. The features No. 7 and 8 must be key information to decide SRLs. The features of the normal CRF model has also the contextual information as seen in Table 3 and the combinatorial features in Table 4. The features defined in Tables 2, 3 and 4 are used in the all of CRF models.

Table 5 shows the features of the last three morphemes used in the CRF+3suf model, while the CRF+2suf model uses the features of the last two morphemes in arguments, i.e., the features from No. 15 to 112 in Table 5. These features consist of four attributes, that are, surface, lemma, reading, and POS, and thus

⁶ We set the threshold to 0 in these experiments.

Table 4. Combination of base features in CRF

No.	Description of feature
c1	Combination of 1 and t1
c2	Combination of 1 and t2
c3	Combination of 4 and t5
c4	Combination of 4 and t6
c5	Combination of 5 and 8
c6	Combination of 6 and 8
c7	Combination of 7 and 8
c8	Combination of 5, 7 and 8
c9	Combination of 6, 7 and 8

Table 5. Features of enhancing the last three morphemes in arguments for CRF+3suf

No.	Description of feature
11	Surface of the third last morpheme in the argument
12	Lemma of the third last morpheme in the argument
13	Reading of the third last morpheme in the argument
14	POS of the third last morpheme in the argument
15	Surface of the second last morpheme in the argument
16	Lemma of the second last morpheme in the argument
17	Reading of the second last morpheme in the argument
18	POS of the second last morpheme in the argument
19	Surface of the last morpheme in the argument
110	Lemma of the last morpheme in the argument
111	Reading of the last morpheme in the argument
112	POS of the last morpheme in the argument

the CRF models can learn various kinds of abstracted levels of the characteristics of ending multiwords of arguments.

The features of the CRF+3suf+c model consist of combinations of the features in Table 6 and the features used in the CRF+3suf. Table 6 shows all of the binary combinations between the second last morpheme and the last morpheme; the base features are surface, lemma, reading, and POS, and then the combinations are 16 features in total. Thus the CRF+3suf+c can capture effective combined features for SRLs.

Next, we describe the CRF with HTCT models. The first model is the CRF model with the features using the output of the HTCT model for characterizing the ending multiwords of arguments instead of using fixed length features of the last a few morphemes, i.e., the features of the CRF with HTCT model are the

Table 6. Combination of features at the last two morphemes in CRF+3suf+c

No.	Description of feature
f1	Combination of l5 and l9
f2	Combination of l5 and l10
f3	Combination of l5 and l11
f4	Combination of l5 and l12
f5	Combination of l6 and l9
f6	Combination of l6 and l10
f7	Combination of l6 and l11
f8	Combination of l6 and l12
f9	Combination of l7 and l9
f10	Combination of l7 and l10
f11	Combination of l7 and l11
f12	Combination of l7 and l12
f13	Combination of l8 and l9
f14	Combination of l8 and l10
f15	Combination of l8 and l11
f16	Combination of l8 and l12

base features of CRF in Tables 2, 3 and 4 with a feature of the best context tag sequence outputted by the HTCT. Several HTCT models are constructed with varying different maximum depth of context tag trees from two to five, and then they are denoted as HTCT-2 to HTCT-5 learned from the training corpus. The second models of the CRF with HTCT take all the features of the CRF model and the HTCT model that shows the best performance among HTCT models in the experiments of Sect. 3.3.

3.3 Experimental Results and Discussions

In this section we will show the preliminary experimental results of detecting SRLs for the test data; that is, all of learning CRF models and construction of HTCT models are done on the training corpus, and the following scores are evaluated on the test corpus described in Sect. 3.1.

Table 7 shows the experimental results of detecting SRLs by the CRF models. In the table, the normal CRF without the features of the ending morphemes of arguments does not work well compared with the cases taking care of the ending morphemes of arguments. Note that the normal CRF also takes into account all of the morphemes in arguments, that is, multiwords at the end of arguments are contained in the features; however, the functional multiwords are separated to individual morphemes then it must be hard for the CRF model to associate the morphemes with the SRLs. In consract, the CRF+2suf model

Table 7. Experimental results of CRF + fixed length of the last a few morphemes in arguments

Model	Precision (%)	Recall (%)	F-measure
CRF	46.74	19.61	27.63
CRF+2suf	47.74	33.96	39.69
CRF+3suf	48.77	37.26	42.25
CRF+3suf+c	47.90	37.22	41.89

Table 8. Experimental Results of CRF + HTCT

Model	Precision (%)	Recall (%)	F-measure
HTCT-2	48.85	35.51	41.12
HTCT-3	51.05	34.53	41.20
HTCT-4	51.09	31.55	39.01
HTCT-5	50.35	29.27	37.02

Table 9. Experimental Results of CRF + fixed + HTCT

Model	Precision (%)	Recall (%)	F-measure
CRF+3suf+HTCT-3	49.71	37.87	42.99
CRF+3suf+c+HTCT-3	49.42	39.79	44.08

and the CRF+3suf model take two or three morpheme sequences as one new features, then the performance of recognizing SRLs is significantly improved.

Comparing the results between the CRF+2suf and the CRF+3suf models, we found that the length of the effective morpheme sequences would be three. Besides, comparing the CRF+3suf with the CRF+3suf+c, the simple application of the combinatorial features of the last two morphemes in arguments does not work well in SRL detection.

The experimental results of the CRF with HTCT model are shown in Table 8. Comparing the different length models of HTCT in F-measure, the HTCT-3 model shows the best performance. This indicates that the HTCT model estimates the effective morpheme length for SRLs must be three on the training corpus, which is the same results in Table 7. Comparing the HTCT models with the CRF+3suf in F-measure, however, the CRF+3suf outperforms all of the HTCT models. If we focus on the precision rates, the HTCT-3 model performs 51.05 % in precision rate whose score is better than the CRF+3suf. This indicates that the arguments annotated correctly by the HTCT model might be different from those by the CRF+3suf, and thus there might be room for improvement of the performance of detecting SRLs by using both features.

The experimental results of the CRF+3suf model or the CRF+3suf+c model combined with HTCT-3 are shown in Table 9. The table shows that the both

combined models outperforms the original models, i.e., CRF+3suf, CRF+3suf+c and HTCT-3 in F-measure. Especially comparing the results in Table 9 with those in Table 7, both the precision and recall rates are improved. These are preliminary results, however, these improvements must indicate that the context tag sequences extracted by HTCT would be different characteristics from manually defined features, and those must be effective for annotating SRLs.

4 Conclusion

We proposed a hierarchical tag context tree approach for capturing the multi-word expressions in arguments of Japanese sentences and show the effectiveness of extracting SRLs by applying the extracted hierarchical context tag sequences to the feature of CRFs. In the future work we will do more detailed analysis of these results.

Acknowledgments. This research received support from JSPS KAKENHI Grant Number 26370485.

References

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, pp. 86–90 (1998)
2. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: an annotated corpus of semantic roles. *Computat. Linguist.* **31**(1), 71–105 (2005)
3. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. *Comput. Linguist.* **28**(3), 1–45 (2002)
4. Surdeanu, M., Johansson, R., Meyers, A., Marquez, L., Nivre, J.: The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In: Proceedings of the 12th Conference on Computational Natural Language Learning, pp. 159–177 (2008)
5. Palmer, M., Gildea, D., Xue, N.: *Semantic Role Labeling*. Morgan & Claypool Publishers, San Rafael (2009)
6. Kawahara, D., Kurohashi, S., Hashida, K.: Construction of a Japanese relevance-tagged corpus. In: Proceedings of the 8th Annual Meeting of the Association for Natural Language Processing, pp. 495–498 (2007) (in Japanese)
7. Iida, R., Komachi, M., Inui, K., Matsumoto, Y.: Annotating a Japanese text corpus with a predicate-argument and coreference relations. In: Proceedings of the 1st Linguistic Annotation Workshop, pp. 132–139 (2007)
8. Komachi, M., Iida, R.: Annotating a Japanese balanced corpus (BCCWJ) with a predicate-argument and coreference relations. In: Workshop for Japanese Corpus, pp. 352–330 (2011) (in Japanese)
9. Ohara, K., Kato, J., Saito, H.: Annotation of Japanese framenet to BCCWJ. In: Proceedings of the Workshop of Japanese Corupus in Grant-in-Aid For Scientific Research on Priority Areas, pp. 513–518 (2011) (in Japanese)
10. Takeuchi, K., Ueno, M., Takeuchi, N.: Annotating semantic role information to Japanese balanced corpus. In: Proceedings of MAPLEX 2015 (2015)

11. Maekawa, K.: Balanced corpus of contemporary written Japanese. In: Proceedings of the 6th Workshop on Asian Language Resources (ALR), pp. 101–102 (2008)
12. EDR, EDR: Electric Dictionary the Second Edition, Japan Electronic Dictionary Research Institute, Ltd. (1995)
13. Haruno, M., Matsumoto, Y.: Mistake-driven mixture of hierarchical tag context trees. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pp. 230–237 (1997)

Computational Linguistics

14th International Conference of the Pacific Association

for Computational Linguistics, PACLING 2015, Bali,

Indonesia, May 19-21, 2015, Revised Selected Papers

Hasida, K.; Purwarianti, A. (Eds.)

2016, X, 263 p. 82 illus., Softcover

ISBN: 978-981-10-0514-5