Empty Category Detection using Path Features and Distributed Case Frames

Shunsuke Takeno[†], **Masaaki Nagata**[‡], **Kazuhide Yamamoto**[†]
[†]Nagaoka University of Technology,
1603-1 Kamitomioka, Nagaoka, Niigata, 940-2188 Japan
{takeno, yamamoto}@jnlp.org
[‡]NTT Communication Science Laboratories, NTT Corporation,
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
nagata.masaaki@labs.ntt.co.jp

Abstract

We describe an approach for machine learning-based empty category detection that is based on the phrase structure analysis of Japanese. The problem is formalized as tree node classification, and we find that the path feature, the sequence of node labels from the current node to the root, is highly effective. We also find that the set of dot products between the word embeddings for a verb and those for case particles can be used as a substitution for case frames. Experiments show that the proposed method outperforms the previous state-of the art method by 68.6% to 73.2% in terms of F-measure.

1 Introduction

Empty categories are phonetically null elements that are used for representing dropped pronouns ("pro" or "small pro"), controlled elements ("PRO" or "big pro") and traces of movement ("T" or "trace"), such as WH-questions and relative clauses. They are important for pro-drop languages such as Japanese, in particular, for the machine translation from pro-drop languages to non-pro-drop languages such as English. Chung and Gildea (2010) reported their recover of empty categories improved the accuracy of machine translation both in Korean and in Chinese. Kudo et al. (2014) showed that generating zero subjects in Japanese improved the accuracy of preordering-based translation.

State-of-the-art statistical syntactic parsers had typically ignored empty categories. Although Penn Treebank (Marcus et al., 1993) has annotations on PRO and trace, they provide only labeled bracketing. Johnson (2002) proposed a statistical pattern-matching algorithm for post-processing the results of syntactic parsing based on minimal unlexicalized tree fragments from empty node to its antecedent. Dienes and Dubey (2003) proposed a machine learning-based "trace tagger" as a preprocess of parsing. Campbell (2004) proposed a rule-based post-processing method based on linguistically motivated rules. Gabbard et al. (2006) replaced the rules with machine learning-based classifiers. Schmid (2006) and Cai et al. (2011) integrated empty category detection with the syntactic parsing.

Empty category detection for pro (dropped pronouns or zero pronoun) has begun to receive attention as the Chinese Penn Treebank (Xue et al., 2005) has annotations for pro as well as PRO and trace. Xue and Yang (2013) formalized the problem as classifying each pair of the location of empty category and its head word in the dependency structure. Wang et al. (2015) proposed a joint embedding of empty categories and their contexts on dependency structure. Xiang et al. (2013) formalized the problem as classifying each IP node (roughly corresponds to S and SBAR in Penn Treebank) in the phrase structure.

In this paper, we propose a novel method for empty category detection for Japanese that uses conjunction features on phrase structure and word embeddings. We use the Keyaki Treebank (Butler et al., 2012), which is a recent development. As it has annotations for pro and trace, we show our method has substantial improvements over the state-of-the-art machine learning-based method (Xiang et al., 2013) for Chinese empty category detection as well as linguistically-motivated manually written rule-based method similar to (Campbell, 2004).

2 Baseline systems

The Keyaki Treebank annotates the phrase structure with functional information for Japanese sentences following a scheme adapted from the Annotation manual for the Penn Historical Corpora and

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1347–1352,
Lisbon, Portugal, 17-21 September 2015.

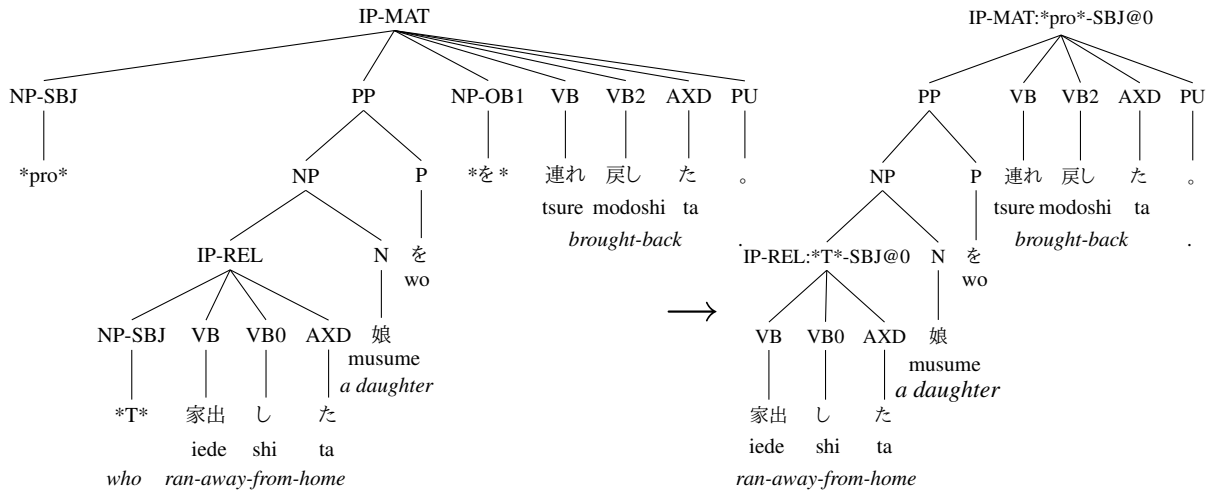

Figure 1: An annotation example of 家出した娘を 連れ戻した。*(*pro* brought back a daughter who ran away from home.)* in Keyaki Treebank. (The left tree is the original tree and the right tree is a converted tree based on Xiang et al.'s (2013) formalism)

the PCEEC (Santorini, 2010). There are some major changes: the VP level of structure is typically absent, function is marked on all clausal nodes (such as IP-REL and CP-THT) and all NPs that are clause level constituents (such as NP-SBJ). Disambiguation tags are also used for clarifying the functions of its immediately preceding node, such as NP-OBJ *を *(wo) for PP, however, we removed them in our experiment.

Keyaki Treebank has annotation for trace markers of relative clauses (*T*) and dropped pronouns (*pro*), however, it deliberately has no annotation for control dependencies (PRO) (Butler et al., 2015). It has also fine grained empty categories of *pro* such as *speaker* and *hearer*, but we unified them into *pro* in our experiment.

HARUNIWA(Fang et al., 2014) is a Japanese phrase structure parser trained on the treebank. It has a rule-based post-processor for adding empty categories, which is similar to (Campbell, 2004). We call it RULE in later sections and use it as one of two baselines.

We also use Xiang et al's (2013) model as another baseline. It formulates empty category detection as the classification of IP nodes. For example, in Figure 1, empty nodes in the left tree are removed and encoded as additional labels with its position information to IP nodes in the right tree. As we can uniquely decode them from the extended IP labels, the problem is to predict the labels for the input tree that has no empty nodes.

Let $T = t_1 t_2 \cdots t_n$ be the sequence of nodes produced by the post-order traversal from root node, and e_i be the empty category tag associated with t_i. The probability model of (Xiang et al., 2013) is formulated as MaxEnt model:

$$P(e_1^n|T) = \prod_{i=1}^{n} P(e_i|e_1^{i-1}, T) = \prod_{i=1}^{n} \frac{\exp(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(e_i, e_1^{i-1}, T))}{Z(e_1^{i-1}, T)} \quad (1)$$

where $\boldsymbol{\phi}$ is a feature vector, $\boldsymbol{\theta}$ is a weight vector to $\boldsymbol{\phi}$ and Z is normalization factor:

$$Z(e_1^{i-1}, T) = \sum_{e \in \mathcal{E}} \exp(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(e, e_1^{i-1}, T))$$

where $\mathcal{E}$ represents the set of all empty category types to be detected.

Xiang et al. (2013) grouped their features into four types: tree label features, lexical features, empty category features and conjunction features as shown in Table 1. As the features for (Xiang et al., 2013) were developed for Chinese Penn Treebank, we modify their features for Keyaki Treebank: First, the traversal order is changed from post-order (bottom-up) to pre-order (top-down). As PROs are implicit in Keyaki Treebank, the decisions on IPs in lower levels depend on those on higher levels in the tree. Second, empty category features are extracted from ancestor IP nodes, not from descendant IP nodes, in accordance with the first change.

Table 2 shows the accuracies of Japanese empty category detection, using the original and our modification of the (Xiang et al., 2013) with ablation test. We find that the conjunction features

Tree label features	
1	current node label
2	parent node label
3	grand-parent node label
4	left-most child label or POS tag
5	right-most child label or POS tag
6	label or POS tag of the head child
7	the number of child nodes
8	one level CFG rule
9	left-sibling label or POS tag (up to two siblings)
10	right-sibling label or POS tag (up to two siblings)
Lexical features	
11	left-most word under the current node
12	right-most word under the current node
13	word immediately left to the span of the current node
14	word immediately right to the span of the current node
15	head word of the current node
16	head word of the parent node
17	is the current node head child of its parent? (binary)
Empty category features	
18	predicted empty categories of the left sibling
19*	the set of detected empty categories of ancestor nodes
Conjunction features	
20	current node label with parent node label
21*	current node label with features computed from ancestor nodes
22	current node label with features computed from left-sibling nodes
23	current node label with lexical features

Table 1: List of features of Xiang et al.'s (2013). (* indicates the features we changed for the Keyaki Treebank)

Features	F(Gold)	Δ
original (Xiang et al., 2013)	68.2	−0.40
modified (Xiang et al., 2013)	68.6	-
− Tree label	68.6	−0.00
− Empty category	68.3	−0.30
− Lexicon	68.6	−0.00
− Conjunction	58.5	−10.1

Table 2: Ablation result of (Xiang et al., 2013)

are highly effective compared to the three other features. This observation leads to the model proposed in the next section.

3 Proposed model

In the proposed model, we use combinations of path features and three other features, namely head word feature, child feature and empty category feature. Path feature (PATH) is a sequence of non-terminal labels from the current node to the ancestor nodes up to either the root node or the nearest CP node. For example, in Figure 1, if the current node is IP-REL, four paths are extracted; IP-REL, IP-REL → NP, IP-REL → NP → PP and IP-REL → NP → PP → IP-MAT.

Head word feature (HEAD) is the surface form of the lexical head of the current node. Child feature (CHILD) is the set of labels for the children of the current node. The label is augmented with the surface form of the rightmost terminal node if it is a function word. In the example of Figure 1, if the current node is IP-MAT, HEAD is 連れ (tsure) and CHILD includes: PP-を (wo), VB, VB2, AXD-た (ta) and PU-。. Empty category feature (EC) is a set of empty categories detected in the ancestor IP nodes. For example in Figure 1, if the current node is IP-REL, EC is *pro*.

We then combine the PATH with others. If the current node is the IP-MAT node in right-half of Figure 1, the combination of PATH and HEAD is:IP-MAT× 連れ (tsure) and the combinations of PATH and CHILD are: IP-MAT×PP-を (wo), IP-MAT×VB, IP-MAT×VB2, IP-MAT×AXD-た (ta) and IP-MAT×PU-。.

3.1 Using Word Embedding to approximate Case Frame Lexicon

A case frame lexicon would be obviously useful for empty category detection because it provides information on the type of argument the verb in question takes. The problem is that case frame lexicon is not usually readily available. We propose a novel method to approximate case frame lexicon for languages with explicit case marking such as Japanese using word embeddings. According to (Pennington et al., 2014), they designed their embedding model GloVe so that the dot product of two word embeddings approximates the logarithm of their co-occurrence counts. Using this characteristic, we can easily make a feature that approximate the case frame of a verb. Given a set of word embeddings for case particles $q_1, q_2, \cdots, q_N \in Q$, the distributed case frame feature (DCF) for a verb w_i is defined as:

$$\tilde{v_i} = w_i \cdot (q_1, q_2, \cdots, q_N) \quad (2)$$

$$v_i = \frac{\tilde{v_i}}{||\tilde{v_i}||} \quad (3)$$

In our experiment, we used a set of high frequency case particles が (ga), は (ha), も (mo), の (no), を (wo), に (ni), へ (he) and から (kara) as Q.

4 Experiment

4.1 Dataset

We divided the Keyaki Treebank into training, development and test sets. As of May 8, 2015, there

		ALL	development/test		
			transcript	blog	newswire
#pro	SBJ	13343	598	187	346
	OB1	1568	43	1	27
	OB2	59	2	0	0
#T	ADT	97	0	0	8
	LOC	164	9	7	7
	OB1	755	0	11	35
	OB2	15	0	1	2
	SBJ	3788	5	40	266
	TMP	53	0	0	3
	MSR	14	0	1	4
	TPC	10	0	0	0
	char/sent.	32.0	32.1	69.5	96.5
	#sent.	22649	591	109	303
	#IP node	46684	1129	544	1841

Table 3: Statistics of Keyaki Treebank

are 22,639 sentences in Keyaki Treebank. We used 1,000 sentences as the development set, 1,003 sentences as the test set. They were taken from the files blog_KNB.psd (blog), spoken_CIAIR.psd (transcript), newswire_MAINICHI-1995.psd (newswire) to balance the domain. The remaining 20,646 sentences are used for training. Further statistics are shown in Table 3.

We used GloVe as word embedding, Wikipedia articles in Japanese as of January 18, 2015, are used for training, which amounted to 660 million words and 23.4 million sentences. By using the development set, we set the dimension of word embedding and the window size for co-occurrence counts as 200 and 10, respectively.

4.2 Result and Discussion

We tested in two conditions: gold parse and system parse. In gold parse condition, we used the trees of Keyaki Treebank without empty categories as input to the systems. In system parse condition, we used the output of the Berkeley Parser model of HARUNIWA before rule-based empty category detection[1]. We evaluated them using the word-position-level identification metrics described in (Xiang et al., 2013). It projects the predicted empty category tags to the surface level. An empty node is regarded as correctly predicted surface position in the sentence, type (T or pro) and function (SBJ, OB1 and so on) are matched with the reference.

To evaluate the effectiveness of the proposed distributed case frame (DCF), we used an existing case frame lexicon (Kawahara and Kurohashi, 2006) and tested three different ways of encoding the case frame information: BIN encodes each case as binary features. SET encodes each combination of required cases as a binary feature. DIST is a vector of co-occurrence counts for each case particle, which can be thought of an unsmoothed version of our DCF.

Table 4 shows the accuracies of various empty category detection methods, for both gold parse and system parse. In the gold parse condition, the two baselines, the rule-based method (RULE) and the modified (Xiang et al., 2013) method, achieved the F-measure of 62.6% and 68.6% respectively.

We also implemented the third baseline based on (Johnson, 2002). Minimal unlexicalized tree fragments from empty node to its antecedent were extracted as pattern rules based on corpus statistics. For *pro*, which has no antecedent, we used the statistics from empty node to the root. Although the precision of the method is high, the recall is very low, which results in the F-measure of 38.1%.

Among the proposed models, the combination of path feature and child feature (PATH $\times$ CHILD) even outperformed the baselines. It reached 73.2% with all features. As for the result of system-parse condition, the F-measure dropped considerably from 73.2% to 54.7% mostly due to the parsing errors on the IP nodes and its function.

We find that there are no significant differences among the different encodings of the case frame lexicon, and the improvement brought by the proposed distributed case frame is comparable to the existing case frame lexicon.

Table 5 shows the ablation result of the proposed model. It indicates conjunction between PATH and CHILD feature is most effective.

	F(Gold)	Δ	F(System)	Δ
Proposed	72.1	-	53.9	-
$-$CHILD	47.4	-24.7	33.7	-20.2
$-$EC	70.8	-1.3	52.4	-1.5
$-$HEAD	70.0	-2.1	51.6	-2.3

Table 5: Ablation result of PATH $\times$ (CHILD + EC + HEAD) model

[1]There are two models available in HARUNIWA, namely the BitPar model (Schmid, 2004) and Berkeley Parser binary branching model (Petrov and Klein, 2007). The output of the later is first flattened, then added disambiguation tags and empty categories using tsurgeon script (Levy and Andrew, 2006).

Models		Gold parse			System parse			
	CF	P	R	F	P	R	F	#nonZ
RULE	-	54.4	73.7	62.6	57.4	50.5	53.7	-
modified (Xiang et al., 2013)	-	76.8	62.0	68.6	57.4	46.9	51.6	321k
modified (Johnson, 2002)	-	81.3	25.0	38.1	66.8	18.1	28.6	-
PATH × CHILD	-	71.0	67.7	69.3	55.9	49.1	52.3	108k
PATH × (CHILD + HEAD + EC)	-	74.8	69.5	72.1	56.2	51.8	53.9	123k
PATH × (CHILD + HEAD + EC)	+DCF	78.0	68.9	73.2	59.7	50.5	54.7	124k
PATH × (CHILD + HEAD + EC)	+BIN	77.1	70.2	73.5	58.8	51.6	55.0	124k
PATH × (CHILD + HEAD + EC)	+SET	77.5	70.0	73.6	58.5	51.4	54.7	126k
PATH × (CHILD + HEAD + EC)	+DIST	77.5	68.3	72.6	60.4	50.6	55.1	124k

Table 4: Result of our models with baselines. #nonZ means the amount of non-zero weight of model

5 Conclusion

In this paper, we proposed a novel model for empty category detection in Japanese using path features and the distributed case frames. Although it achieved fairly high accuracy for the gold parse, there is much room for improvement when applied to the output of a syntactic parser. Since the accuracy of the empty category detection implemented as a post-process highly depends on that of the underlying parser, we want to explore models that can solve them jointly, such as the lattice parsing approach of (Cai et al., 2011). We would like to report the results in the future version of this paper.

References

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

Alastair Butler, Shota Hhiyama, and Kei Yoshimoto. 2015. Coindexed null elements for a Japanese parsed corpus. In *Proceedings of the 21th Annual Meeting of the Association for Natural Language Processing*, pages 708–711.

Shu Cai, David Chiang, and Yoav Goldberg. 2011. Language-Independent Parsing with Empty Elements. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, volume 2, pages 212–216.

Richard Campbell. 2004. Using linguistic principles to recover empty categories. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 645–652.

Tagyoung Chung and Daniel Gildea. 2010. Effects of Empty Categories on Machine Translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 636–645.

Péter Dienes and Amit Dubey. 2003. Deep Syntactic Processing by Combining Shallow Methods. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 431–438.

Tsaiwei Fang, Alastair Butler, and Kei Yoshimoto. 2014. Parsing Japanese with a PCFG treebank grammar. In *Proceedings of The Twentieth Meeting of the Association for Natural Language Processing*, volume C, pages 432–435.

Ryan Gabbard, M Marcus, and Seth Kulick. 2006. Fully Parsing the Penn Treebank. In *Proceedings of the main conference on Human Language Technology of the North American Chapter of the Association of Computational Linguistics*, pages 184–191.

Mark Johnson. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 136–143.

Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 1344–1347.

Taku Kudo, Hiroshi Ichikawa, and Hideto Kazawa. 2014. A joint inference of deep case analysis and zero subject generation for Japanese-to-English statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 557–562.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of 5th International Conference on Language Resources and Evaluation*, pages 2231–2234.

Mitchell P Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated

Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Slav Petrov and Dan Klein. 2007. Improved Inferencing for Unlexicalized Parsing. In *Proceedings of HLT-NAACL 2007 - Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 404–411.

Santorini. 2010. Annotation Manual for the Penn Historical Corpora and the PCEEC (Release 2).

Helmut Schmid. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 162–168.

Helmut Schmid. 2006. Trace Prediction and Recovery with Unlexicalized PCFGs and Slash Features. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 177–184.

Bing Xiang, Xiaoqiang Luo, and Bowen Zhou. 2013. Enlisting the ghost: Modeling empty categories for machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 822–831.

Nianwen Xue and Yaqin Yang. 2013. Dependency-based empty category detection via phrase structure trees. In *Proceedings of HLT-NAACL 2013, Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings of the Main Conference*, pages 1051–1060.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238.