Automatically Acquired Lexical Knowledge Improves Japanese Joint Morphological and Dependency Analysis

Daisuke Kawahara and **Yuta Hayashibe**[1] and **Hajime Morita**[2] and **Sadao Kurohashi**
Graduate School of Informatics, Kyoto University
{dk, kuro}@i.kyoto-u.ac.jp
hayashibe@fairydevices.jp, hmorita@jp.fujitsu.com

Abstract

This paper presents a joint model for morphological and dependency analysis based on automatically acquired lexical knowledge. This model takes advantage of rich lexical knowledge to simultaneously resolve word segmentation, POS, and dependency ambiguities. In our experiments on Japanese, we show the effectiveness of our joint model over conventional pipeline models.

1 Introduction

Morphological analysis, i.e., word segmentation, POS tagging and lemmatization, is the first step for processing unsegmented languages such as Chinese and Japanese. Words segmented by a morphological analyzer are usually fed into subsequent analyzers, such as dependency parsers and predicate-argument structure (PAS) analyzers, in a pipeline manner. One problem with this pipeline process is that errors in morphological analysis are propagated to the subsequent steps. In morphological analysis, it is also difficult in some cases to determine word segmentations without syntactic and structural knowledge, which could be available at the step of dependency or PAS analysis.

For instance, the Japanese phrase “あるかない” in Sentence (1) can be segmented into (2a) or (2b).[3]

(1) 可能性 が あるかない か 分から ない
possibility NOM ___ or know not

(2) a. ある / か / ない
exist / or / not
b. あるか / ない
walk / not

In this case, (2a) is the correct segmentation, which means “whether a possibility exists,” while the incorrect segmentation (2b) is meaningless: “a possibility does not walk.” It might be possible to select the correct segmentation if a morphological analyzer could look up selectional preference knowledge of the predicates “exist” and “walk.”

Thus far, several models have been proposed for joint morphological and dependency analysis, but the performance improvement is not stable among target languages. For Chinese joint analysis, where the parsing accuracy of a baseline pipeline model is around 80%, an F1 improvement of around 2% was reported (Hatori et al., 2012; Zhang et al., 2014). For Japanese joint analysis, where the parsing accuracy of a pipeline model is around 90%, there have been no studies that report a significant improvement (Tawara et al., 2015). One of the reasons for such instability is that most of these joint models are trained only on a small-scale treebank, which consists of several tens of thousands of sentences. These models do not make use of large-scale external lexical knowledge. Since it is necessary to use lexical knowledge of selectional preferences to address the abovementioned ambiguities, these joint models cannot solve such ambiguities in many cases.

This paper proposes a joint model for morphological and dependency analysis based on automatically acquired lexical knowledge. The lexical knowledge includes case frames acquired from a large-scale raw corpus, which provide useful clues to resolve morphological and syntactic ambiguities. In our experiments on Japanese corpora, we show a significant improvement over conventional

[1]The second author is now affiliated with Fairy Devices Inc.

[2]The third author is now affiliated with Fujitsu Laboratories Ltd.

[3]In this paper, we use the following abbreviations: NOM (nominative), ACC (accusative), DAT (dative), LOC (locative), ABL (ablative), and TOP (topic marker).

Proceedings of the 15th International Conference on Parsing Technologies, pages 1–10,
Pisa, Italy; September 20–22, 2017. ©2017 Association for Computational Linguistics

pipeline models.

The remainder of this paper is organized as follows. Section 2 summarizes previous joint models for morphological and dependency analysis. Section 3 describes our method for constructing lexical knowledge. Section 4 illustrates our idea and describes our joint analysis model in detail. Section 5 is devoted to our experiments. Finally, section 6 gives the conclusions.

2 Related Work

Some variants of transition-based parsing methods have been proposed for joint POS tagging and parsing (Bohnet and Nivre, 2012; Bohnet et al., 2013; Wang and Xue, 2014) and joint Chinese word segmentation, POS tagging, and dependency parsing (Hatori et al., 2012; Zhang et al., 2014). As an external knowledge source, Hatori et al. (2012) used a word dictionary extracted mainly from Wikipedia, but it did not provide lexical knowledge for resolving syntactic ambiguities.

Lattice parsing methods have been proposed for Hebrew and Arabic (Goldberg and Tsarfaty, 2008; Goldberg et al., 2009; Green and Manning, 2010; Goldberg and Elhadad, 2011). These methods first generate a word lattice and then apply PCFG parsing to the word lattice. Starting with a word lattice, the methods of Wang et al. (2013) and Zhang et al. (2015) select the best parse using dual decomposition and the randomized greedy algorithm, respectively. Of these methods, Goldberg et al. (2009) incorporated an external morphological lexicon, which does not provide selectional preferences.

As a different method from lattice parsing, Qian and Liu (2012) trained separate models for Chinese word segmentation, POS tagging, and constituency parsing. They proposed a unified decoding algorithm that combines the scores from these three models. This is a purely supervised method that does not use lexical knowledge.

As dependency parsing models using lexical knowledge, there have been semi-supervised approaches that use knowledge of word classes, lexical preferences or selectional preferences acquired from raw corpora (e.g., (van Noord, 2007; Koo et al., 2008; Chen et al., 2009; Zhou et al., 2011; Bansal and Klein, 2011)). However, these dependency parsing models cannot be applied to joint morphological and dependency analysis.

For Japanese, Morita et al. (2015) proposed a morphological analyzer that jointly performs segmentation and POS tagging using recurrent neural network language models, but does not perform dependency parsing. We employ this morphological analyzer, JUMAN++[4], as a pre-processor to generate word lattice (described in Section 4.1). Kawahara and Kurohashi (2006) proposed a probabilistic model for Japanese dependency parsing and PAS analysis based on case frames automatically compiled from a large raw corpus, which are also used as a source of selectional preferences in our model (described in Section 3.1). Kudo and Matsumoto (2002), Sassano (2004), Iwatate (2012) and Yoshinaga and Kitsuregawa (2014) proposed supervised models for Japanese dependency parsing without using external knowledge sources. These models need a 1-best output of segmentation and POS tagging as an input, and are not a joint model of morphological analysis and dependency parsing. We adopt KNP[5] and CaboCha[6] as baseline dependency parsers, which are implementations of Kawahara and Kurohashi (2006) and Sassano (2004), respectively.[7]

Tawara et al. (2015) proposed a joint model for Japanese morphological analysis and dependency parsing without lexical knowledge. However, they failed to achieve significant improvements over conventional pipeline methods.

To the best of our knowledge, there have been no joint models of morphological and dependency analysis that use large-scale lexical knowledge which includes selectional preferences.

3 Lexical Knowledge Acquisition

In our joint analysis model, we use the following three types of lexical knowledge automatically acquired from a large raw corpus: case frames, cooccurrence probabilities of noun-noun / predicate-predicate dependencies, and word embeddings. We deeply utilize case frames in our joint model

[4] `http://nlp.ist.i.kyoto-u.ac.jp/EN/?JUMAN++`

[5] `http://nlp.ist.i.kyoto-u.ac.jp/?KNP`

[6] `https://taku910.github.io/cabocha/`

[7] As baseline parsers, we did not use J.DepP (`http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/jdepp/`) and the tournament model proposed by Iwatate (2012). J.DepP is an implementation of Yoshinaga and Kitsuregawa (2014), which uses the same shift-reduce model as CaboCha with a similar feature set. It was also empirically proved by the author of CaboCha that the tournament model of Iwatate (2012) did not significantly outperform the shift-reduce model. Iwatate (2012) further improved the performance of a single parser using parser stacking. This kind of parser combination technique is complementary to our model and can be incorporated into our model in the future.

and also consider these resources as features in our scoring function described in Section 4.2. Below, we describe the methods for constructing each of the resources, which are basically based on previous work.

3.1 Case Frames

We use case frames to evaluate the plausibility of PASs. Case frames are predicate-specific semantic frames like PropBank (Palmer et al., 2005), which are distinguished for each predicate sense or usage. Although PropBank was elaborated by hand and does not have frequency information, we automatically compile large-scale case frames that reflect real predicate uses.

Each predicate has several case frames that are semantically distinguished. Each case frame consists of case slots, each of which consists of word instances that can be filled. Examples of Japanese case frames are shown in Table 1. Case frames are the source of selectional preferences, which are compiled by aggregating PASs for each predicate usage.

We adapted the method of Kawahara et al. (2014) to Japanese case frame compilation. They proposed an unsupervised method for compiling English case frames from a large raw corpus. The procedure for inducing case frames is as follows:

1. apply dependency parsing to a raw corpus and extract PASs for each predicate from the automatic parses,
2. merge the PASs that have presumably the same meaning based on the assumption of one sense per collocation to get a set of initial frames, and
3. apply clustering to the initial frames based on the Chinese Restaurant Process (Aldous, 1985) to produce predicate-specific case frames.

While the original method used Stanford dependency labels as the representations of case slots, we use case-marking postpositions in Japanese, such as "が" (NOM), "を" (ACC), and "に" (DAT). At Step 1, we apply the morphological and dependency analyzers, JUMAN++ and KNP, to the raw corpus. To alleviate the influence of errors in segmentations, POS tags and dependencies, we extract only reliable PASs that have no syntactic ambiguities. At Step 2, the PASs that have presumably the same meaning are identified by coupling a predicate and the closest argument. That is, PASs are distinguished by predicate-argument pairs, such as "道を あるく" (walk road) and "町を あるく" (walk city).

	CS	instances
ある:1 (exist:1)	が	necessity:297865, case:190109, ···
	に	thing:40, me:29, trend:29, ···
	time	<time>:398
ある:2 (exist:2)	が	interest:34236, confidence:21326, ···
	に	point:702, way:490, me:442, ···
	で	feeling:70, aspect:58, ···
ある:3 (exist:3)	が	possibility:121867
	に	price:23, myself:20, you:18, ···
	で	step:4, influence:4, ···
⋮	⋮	⋮
あるく:1 (walk:1)	が	person:57, I:13, ···
	を	road:24236, trail:4066, ···
	から	parking:175, station:88, ···
あるく:2 (walk:2)	が	I:35, parade:27, ···
	を	city:13548, town:5336, park:3264, ···
	で	alone:464, feeling:74, ···
あるく:3 (walk:3)	が	person:60, cat:24, ···
	を	inside:18858, top:9969, bottom:1769, ···
	で	alone:216, barefoot:198, ···
⋮	⋮	⋮

Table 1: Acquired case frames for the Japanese verbs "ある" (exist) and "あるく" (walk). CS denotes case slots, such as "が" (NOM), "を" (ACC), "に" (DAT), "で" (LOC), and "から" (ABL). Instances in each CS are originally Japanese words but expressed only in English due to space limitation. The number following each word denotes its frequency.

We crawled the Web to obtain a large-scale Japanese Web corpus. As a result, we extracted 10 billion Japanese sentences without duplicates from three billion Web pages. We automatically compiled case frames from these sentences and acquired case frames for approximately 100,000 predicates. Examples of acquired case frames are shown in Table 1.

3.2 Cooccurrence Probabilities of Noun-noun / Predicate-predicate Dependencies

To evaluate dependencies that cannot be captured by PASs, cooccurrence statistics of these dependencies are collected from a large raw corpus. For such dependencies, we consider noun-noun and predicate-predicate dependencies. Noun-noun dependencies cover the dependency relations between nouns including compound nouns and predicate-predicate dependencies are the dependency relations between predicates.

We collect noun-noun and predicate-predicate dependencies from automatic parses and calculate cooccurrence probabilities of these dependencies. We acquired these probabilities from automatic parses of 1.6 billion Japanese Web sentences, which are a part of the Japanese Web corpus constructed for case frame compilation.

3.3 Word Embeddings

To detect coordinate structures, which cover a large proportion of dependency relations in a sentence, it is important to capture similarities between words and word sequences. In this paper, we employ word embeddings to calculate similarities between words and word sequences.

We trained word embeddings using 100 million Japanese Web sentences by word2vec[8] (Mikolov et al., 2013) using skip-gram and negative sampling. The dimension of word embeddings was set to 500. To calculate the similarity between two words, we compute the cosine measure between the embeddings of these words.

4 Joint Morphological and Dependency Analysis based on Automatically Acquired Lexical Knowledge

4.1 Joint Analysis Model

We deal with dependencies between base phrases, which are the dependency unit defined in the annotation guidelines of the Japanese treebanks described in Section 5.1. A base phrase consists of a content word and zero or more function words. Although the traditional dependency unit for Japanese is *bunsetsu*, which can contain more than one content word,[9] we adopt base phrase dependencies instead of *bunsetsu* dependencies. This is because base phrase dependencies are basically based on *bunsetsu* dependencies but extended to consider the dependencies inside compound nouns. Hereafter, we call base phrases simply *phrases*.

We adopt the widely used CKY algorithm for our joint analysis model. In our model, a cell in the CKY table corresponds to a span of characters in the input sentence. This model outputs the best parse tree, which contains all the disambiguated results of words, phrases, and dependencies. The procedure of our joint analysis model is described below.

1. Projection of candidate words onto the CKY table

First, a word lattice is generated using a morphological analyzer. All the words included in the word lattice are projected onto the CKY table.

For instance, in Figure 1, the input sentence is "可能性があるかないか." The possible words for this sentence are projected as described in Figure 1(a). For example, the span "あるか" has the following three possible word cells: "ある" (exist), "か" (or), and "あるか" (walk).

2. Generation of phrases

Possible phrases are generated on the CKY table using POS-based phrase chunking rules. These rules are extracted from the Japanese dependency parser KNP. Because there are ambiguities in words and POS tags, there are also ambiguities in phrases. Each of the generated phrases is regarded as the smallest sub-tree consisting of only one phrase.

In Figure 1, by concatenating words in Figure 1(a) using the phrase chunking rules, the light blue cells in Figure 1(b) are generated as candidate phrases. For example, the span "あるか" has the following two possible phrases: "あるか" (walk) and "ある / か" (exist or).

3. Merging neighboring sub-tree pairs

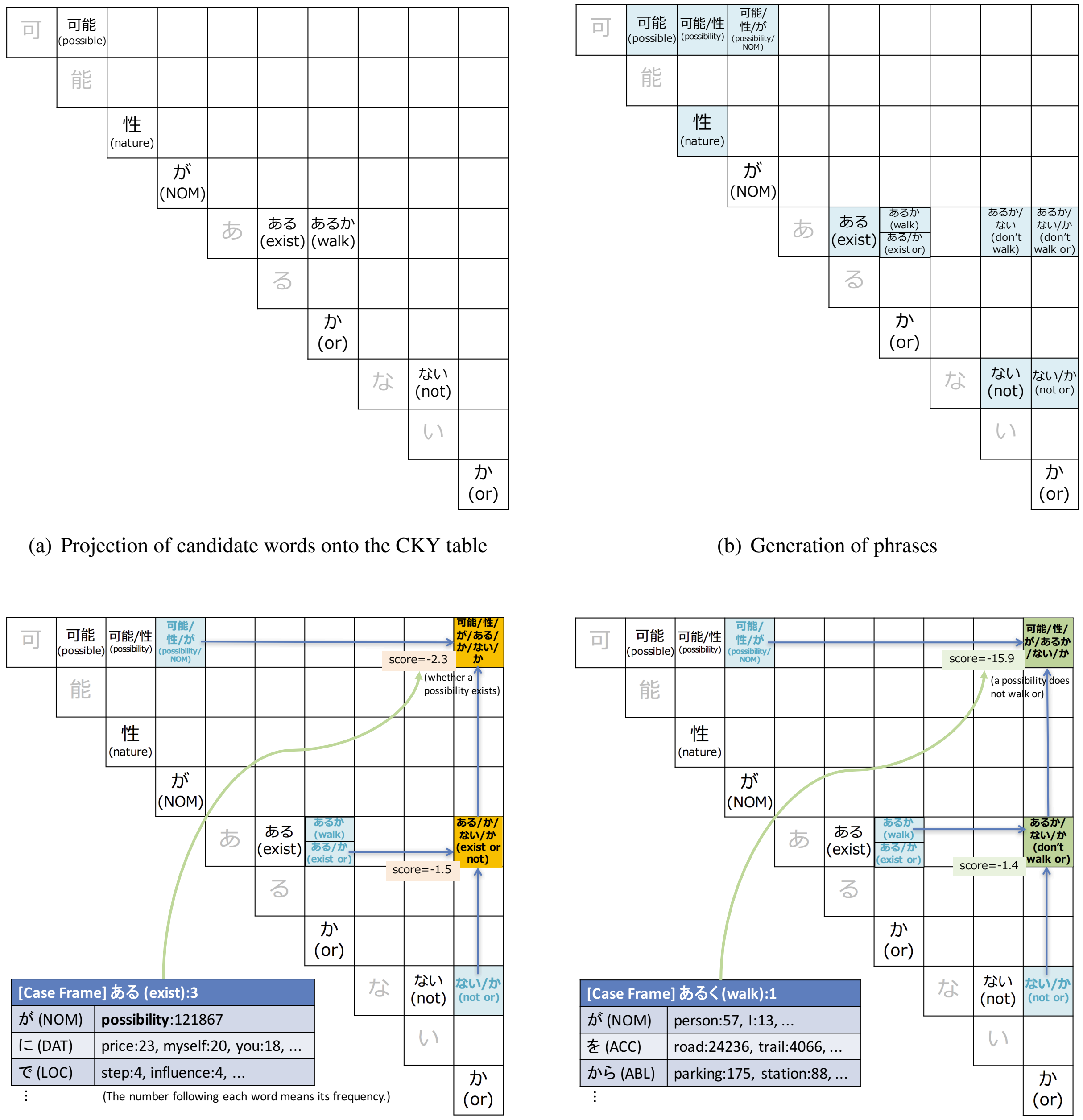
Neighboring sub-tree pairs are merged to generate a new possible sub-tree. This process is iterated in a bottom-up manner, and finally possible trees for the whole input sentence are generated.

In general, there can be multiple sub-trees that correspond to the same span. When merging two spans with multiple sub-trees, it is necessary to consider all the possible combinations of these sub-trees. New sub-trees generated by merging are ranked by the scoring function described in Section 4.2 and only top-b sub-trees are kept for the subsequent process, where b is the beam width.

Different from the usual CKY algorithm for dependency parsing, we perform PAS analysis for each cell whose head is a predicate. This analysis is done using the method of Kawahara and Kurohashi (2006), which is the process of matching between the arguments in the span and each of the case frames of the predicate. The best-matching

[8] https://code.google.com/p/word2vec/

[9] A *bunsetsu* consists of one or more content words and zero or more function words. A compound noun containing multiple content words constitutes a *bunsetsu*.



(a) Projection of candidate words onto the CKY table

(b) Generation of phrases

(c) Generation process of the tree for "可能/性/が/ある/か/ない/か" (correct analysis)

(d) Generation process of the tree for "可能/性/が/あるか/ない/か" (incorrect analysis)

Figure 1: Illustration of our joint analysis model.

case frame with the highest score is selected according to the scoring function.

For example, Figure 1(c) shows the merging process for the interpretation "可能/性/が/ある/か/ない/か" (whether a possibility exists), and Figure 1(d) shows the merging process for "可能/性/が/あるか/ない/か" (possibility does not walk). The best-matching case frame "ある:3" (exist:3) was selected for the interpretation in Figure 1(c), and the case frame "あるく:1" (walk:1) was selected in Figure 1(d), respectively.

4. Selection of the tree with the highest score

The tree with the highest score is selected as an output from the candidate trees for the whole input sentence using the following equation:

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}}\, score(y), \tag{1}$$

where $\hat{y}$ is the output tree, and Y is the candidate trees for the input sentence. The scoring function $score(y)$ is defined in Section 4.2.

Word feature	
· Marginal score of morphological analysis	
Phrase features	
· Word 2,3-grams in a phrase	
· # of phrases in a sentence	
· Words at a phrase boundary	
· # of predicates	
· A predicate representation	
Dependency features	
· A dependency label	
· Content/function words and punctuations of a modifier	
· Content/function words and punctuations of a head	
· Distance between a modifier and its head	
Features derived from lexical knowledge	
· # of predicates that do not have case frames	
· Probabilities calculated based on case frames (case frame/slot generating probability, etc.)	
· A cooccurrence probability between nouns	
· A cooccurrence probability between predicates	
· Content word similarity between a modifier and its head	
· Similarity of word sequences for coordination	

Table 2: Features.

In Figure 1, the upper-right corner cell of the CKY table, which keeps the interpretations of the whole input sentence, contains two possible trees illustrated in Figures 1(c) and 1(d). Our algorithm selects the tree of 1(c), which has the highest score. Here, selectional preferences from case frames tell that "ある" (exist) is more likely to take the nominative filler "可能性" (possibility) than "あるく" (walk).

4.2 Scoring Function and Training

The score of a tree for the input sentence x is defined as the weighted sum of features. This score is calculated using the following scoring function:

$$score(y) = \sum_i \left(w_i \cdot \phi_i \left(x, y \right) \right), \qquad (2)$$

where ϕ_i is a feature function corresponding to feature i, and w_i is a weight of feature i. This scoring function is also used for calculating the score of a sub-tree, which is constructed at an intermediate step of parsing, i.e., an intermediate cell built at Step 3.

Table 2 lists the features used, which include words constituting a phrase, dependencies between phrases, and the plausibility of a PAS measured by case frames. The basic features that are not derived from lexical knowledge (the upper part of Table 2) are based on the features used in the CaboCha parser.[10]

[10]Although the features for dependency parsing in CaboCha were designed for *bunsetsu* dependency parsing, we found out that these features are also compatible with base-phrase dependency parsing.

Corpus	Training	Test
NEWS	2,727 articles (36,623 sentences)	200 articles (1,783 sentences)
WEB	4,427 articles (13,853 sentences)	700 articles (2,195 sentences)

Table 3: Statistics of the treebanks.

We use the following learning procedure. First, the feature weights are initialized, and the word lattice for each sentence in a training corpus is input. A sentence is analyzed using the method described in Section 4.1 with a beam width of b, and candidate trees for this sentence are obtained. The tree with the highest dependency score (UAS) against the gold tree is regarded as a positive instance, and the remaining candidate trees are regarded as negative instances. The feature weights are optimized using the training instances generated from all the sentences in the training corpus. We adopt candidate selection learning and optimize the feature weights using L-BFGS. The above procedure is iterated several times to obtain the final feature weights.

5 Experiments

5.1 Experimental Settings

In our experiments, we used the Kyoto University Text Corpus[11] (Kawahara et al., 2002) (NEWS) and the Kyoto University Web Document Leads Corpus[12] (Hangyo et al., 2012) (WEB) as Japanese treebanks. NEWS consists of news articles and WEB consists of web pages in various domains. We split these into training and test sets as shown in Table 3. We merged the training sets of NEWS and WEB to generate a training set in our experiment and conducted evaluations on each test set of NEWS and WEB.

For the parser input, we used the Japanese morphological analyzer JUMAN++ (Morita et al., 2015) to generate a word lattice. We did not use all possible words in the lexicon of JUMAN++, but converted the N-best output of JUMAN++ to a word lattice to speed up parsing. This is reasonable because the segmentation accuracy of JUMAN++ is between 98%–99% and its N-best output contains only plausible words. N-best outputs were obtained using the option –autoN of JU-

[11]`http://nlp.ist.i.kyoto-u.ac.jp/EN/?KyotoUniversity\%20Text\%20Corpus`

[12]`http://nlp.ist.i.kyoto-u.ac.jp/EN/?KWDLC`

Input		Model	Word			Phrase	Dependency	
Data	Morph output		Seg	POS	All	pSeg	UAS	LAS
NEWS	1-best	KNP	99.38	98.97	97.50	98.35	89.68	87.98
		CaboCha	99.38	98.97	97.50	96.17	89.06	-
		KNP+CaboCha	99.38	98.97	97.50	98.35	91.00	-
		Our model wo/LK	99.38	98.97	97.50	98.38	89.89	88.20
	N-best		99.37	98.98	97.51	98.39	90.40	88.73
	1-best	Our model	99.38	98.97	97.50	98.39	91.26	89.54
	N-best		99.38	**99.00**	**97.54**	**98.44**	**91.61**	**89.91**
WEB	1-best	KNP	98.45	97.91	96.34	96.30	87.87	85.61
		CaboCha	98.45	97.91	96.34	92.65	86.14	-
		KNP+CaboCha	98.45	97.91	96.34	96.30	89.05	-
		Our model wo/LK	98.45	97.91	96.34	96.13	88.36	86.12
	N-best		98.48	97.93	96.39	96.26	88.79	86.52
	1-best	Our model	98.45	97.91	96.34	96.11	89.54	87.27
	N-best		**98.53**	**97.99**	**96.45**	96.31	**89.82**	**87.53**

Table 4: Evaluation results. "wo/LK" means "without lexical knowledge."

MAN++, which increases N proportionally to the length of the input sentence. We applied 10-way jackknifing to the training set and analyzed the test set using a model trained on the whole training set.

To train our joint model, we used Classias[13] with L1 regularization. We set the beam width b of our model to 10 for both training and testing.

For comparison, we adopted the latest versions of KNP and CaboCha, both of which are widely used Japanese dependency parsers. KNP is an implementation of Kawahara and Kurohashi (2006), which accepts the 1-best output of morphological analysis, applies rule-based phrase chunking and performs probabilistic labeled dependency parsing based on case frames. In KNP, we used the same case frames compiled in this paper. CaboCha is an implementation of Sassano (2004), which accepts the 1-best output of morphological analysis, applies CRF-based phrase chunking and performs transition-based unlabeled dependency parsing using SVM. The training of CRF and SVM was conducted using the training data in this paper. Because phrase chunking in CaboCha was designed to identify *bunsetsu*, we also tested KNP+CaboCha for fair comparison, which identifies phrases using KNP and parses using CaboCha. Since KNP and CaboCha are not a joint model and accept only the 1-best output of morphological analysis, we fed the 1-best morphological analysis into KNP and CaboCha. We fed both 1-best and N-best morphological analysis outputs into our model for comparison. We also tested our model without the automatically acquired lexical knowledge.

We measured the performance of each system using the F1-scores of the following aspects: word segmentation (Seg), "segmentation + POS" (POS), "segmentation+ POS + fine-grained POS + base form" (All), phrase segmentation (pSeg), and unlabeled/labeled dependency attachment score (UAS/LAS). For the dependency labels, the following four labels are defined in the treebanks: D (dependency), P (parallel), I (incomplete parallel), and A (apposition).

5.2 Results and Discussion

Table 4 lists the evaluation results. In this table, the accuracies in bold of "our model with N-best input" are significantly higher than the other models. Statistical testing was conducted using the bootstrap method (Efron and Tibshirani, 1986, 1993) at $p < 0.01$.

The following is typical examples improved by our joint model.

(3) a. あの 店 は で もの × が よく 見つかる
that shop TOP LOC thing NOM often found

b. あの 店 は でもの ○ が よく 見つかる
that shop TOP bargain NOM often found

In this example, (3a) is the incorrect output of the

[13] http://www.chokkan.org/software/classias/

	Seg	POS	All	pSeg	UAS	LAS
1-best	97.01	95.26	94.51	94.21	86.87	85.52
N-best	**97.38**	**95.63**	**95.13**	**94.44**	**87.84**	**86.49**

Table 5: Results on a corpus with ambiguities.

baseline systems, and (3b) is the correct output of our joint model. Here, case frames tell that the verb "見つかる" (be found) is likely to take "でもの" (bargain) as its nominative.

(4) a. お いや × めい と わかれる
(prefix) nasty niece ABL part

b. おい や ○ めい と わかれる
nephew and niece ABL part

In this example, (4a) is the incorrect output of the baseline systems, and (4b) is the correct output of our joint model. In this case, "わかれる" (part) is likely to take people including "おい" (nephew) and "めい" (niece) as the ablative fillers. Also, because "おい" (nephew) and "めい" (niece) are judged to be similar by the word embeddings, these are recognized as a coordinate structure.

In Table 4, while the dependency accuracies were improved well, the improvements in morphological analysis (Seg, POS, and All) and phrase segmentation (pSeg) were moderate, even though most of them were significant. In Japanese, the same word segments can have multiple possible words with the same POS and base form, which do not influence the segmentation and POS accuracy. For example, consider the following sentence.

(5) 皮 を むく
rind ACC peel/turn

The verb "むく" is represented as two possible words with different meanings "peel" and "turn," both of which appear in the N-best output of morphological analysis. Although "peel" is correct, this kind of meaning difference is not distinguished in the evaluation of segmentation and POS tagging.[14] However, such ambiguities are resolved based on lexical knowledge in our joint analysis model, and this disambiguation leads to the improvement of case frame selection and dependency parsing.

To further verify the improvements in morphological analysis, we manually annotated 50 sentences with various morphological ambiguities using the same annotation criteria as NEWS and WEB. We tested our model given 1-best and N-best morphological analysis with lexical knowledge. Table 5 shows the results. The joint model (N-best) outperformed the pipeline model (1-best) in terms of all the measures by a large margin.

6 Conclusion

This paper proposed a joint model for morphological and dependency analysis based on automatically acquired lexical knowledge. This model takes advantage of rich lexical knowledge to jointly resolve word segmentation, POS, and dependency ambiguities. In our Japanese experiments, we succeeded in showing the effectiveness of our joint model over conventional pipeline models.

In the future, we will try to incorporate lexical knowledge into a neural network-based model for joint morphological and dependency analysis. By doing this, we can automatically consider feature combinations as the polynomial kernel used in CaboCha. We also plan to integrate PAS analysis including zero anaphora resolution into our joint model.

References

David Aldous. 1985. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII —1983* pages 1–198.

Mohit Bansal and Dan Klein. 2011. Web-scale features for full-scale parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 693–702. http://www.aclweb.org/anthology/P11-1070.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Lan-*

[14] If this verb "むく" is written using a Chinese character, such as "向く" (turn) and "剥く" (peel), this kind of ambiguity does not occur. However, there are many uses of words without using Chinese characters, especially on Web texts.

guage Learning. Association for Computational Linguistics, Jeju Island, Korea, pages 1455–1465. http://www.aclweb.org/anthology/D12-1133.

Bernd Bohnet, Joakim Nivre, Igor Boguslavsky, Richárd Farkas, Filip Ginter, and Jan Hajič. 2013. Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics* 1:415–428. http://www.aclweb.org/anthology/Q13-1034.

Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Improving dependency parsing with subtrees from auto-parsed data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, pages 570–579. http://www.aclweb.org/anthology/D/D09/D09-1060.

Bradley Efron and Robert Tibshirani. 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science* 1(1):54–75.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall/CRC.

Yoav Goldberg and Michael Elhadad. 2011. Joint Hebrew segmentation and parsing using a PCFGLA lattice parser. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 704–709. http://www.aclweb.org/anthology/P11-2124.

Yoav Goldberg and Reut Tsarfaty. 2008. A single generative model for joint morphological segmentation and syntactic parsing. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics, Columbus, Ohio, pages 371–379. http://www.aclweb.org/anthology/P/P08/P08-1043.

Yoav Goldberg, Reut Tsarfaty, Meni Adler, and Michael Elhadad. 2009. Enhancing unlexicalized parsing performance using a wide coverage lexicon, fuzzy tag-set mapping, and EM-HMM-based lexical probabilities. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Association for Computational Linguistics, Athens, Greece, pages 327–335. http://www.aclweb.org/anthology/E09-1038.

Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Coling 2010 Organizing Committee, Beijing, China, pages 394–402. http://www.aclweb.org/anthology/C10-1045.

Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. 2012. Building a diverse document leads corpus annotated with semantic relations. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*. Faculty of Computer Science, Universitas Indonesia, Bali,Indonesia, pages 535–544. http://www.aclweb.org/anthology/Y12-1058.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental joint approach to word segmentation, POS tagging, and dependency parsing in Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Jeju Island, Korea, pages 1045–1053. http://www.aclweb.org/anthology/P12-1110.

Masakazu Iwatate. 2012. Development of pairwise comparison-based Japanese dependency parsers and application to corpus annotation. Ph.D dissertation at Nara Institute of Science and Technology.

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 NAACL, Main Conference*. Association for Computational Linguistics, New York City, USA, pages 176–183. http://www.aclweb.org/anthology/N/N06/N06-1023.

Daisuke Kawahara, Sadao Kurohashi, and Koiti Hasida. 2002. Construction of a Japanese relevance-tagged corpus. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*. pages 2008–2013. http://www.lrec-conf.org/proceedings/lrec2002/pdf/302.pdf.

Daisuke Kawahara, Daniel Peterson, Octavian Popescu, and Martha Palmer. 2014. Inducing example-based semantic frames from a massive amount of verb uses. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Gothenburg, Sweden, pages 58–67. http://www.aclweb.org/anthology/E14-1007.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics, Columbus, Ohio, pages 595–603. http://www.aclweb.org/anthology/P/P08/P08-1068.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning*. pages 63–69. http://www.aclweb.org/anthology/W/W02/W02-2016.pdf.

Tomas Mikolov, Chen Kai, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at International Conference on Learning Representations*.

Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. 2015. Morphological analysis for unsegmented languages using recurrent neural network language model. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 2292–2297. http://aclweb.org/anthology/D15-1276.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics* 31(1):71–106.

Xian Qian and Yang Liu. 2012. Joint Chinese word segmentation, POS tagging and parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Jeju Island, Korea, pages 501–511. http://www.aclweb.org/anthology/D12-1046.

Manabu Sassano. 2004. Linear-time dependency analysis for Japanese. In *Proceedings of Coling 2004*. COLING, Geneva, Switzerland, pages 8–14. http://www.aclweb.org/anthology/C04-1002.

Yuki Tawara, Ai Azuma, and Yuji Matsumoto. 2015. Japanese morphological analysis using dependency information (*in Japanese*). In *IPSJ 2015-NL-220*. pages 1–7.

Gertjan van Noord. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the Tenth International Conference on Parsing Technologies*. Association for Computational Linguistics, Prague, Czech Republic, pages 1–10. http://www.aclweb.org/anthology/W/W07/W07-2201.

Zhiguo Wang and Nianwen Xue. 2014. Joint POS tagging and transition-based constituent parsing in Chinese with non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 733–742. http://www.aclweb.org/anthology/P14-1069.

Zhiguo Wang, Chengqing Zong, and Nianwen Xue. 2013. A lattice-based framework for joint Chinese word segmentation, POS tagging and parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 623–627. http://www.aclweb.org/anthology/P13-2110.

Naoki Yoshinaga and Masaru Kitsuregawa. 2014. A self-adaptive classifier for efficient text-stream processing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 1091–1102. http://www.aclweb.org/anthology/C14-1103.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Character-level Chinese dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 1326–1336. http://www.aclweb.org/anthology/P14-1125.

Yuan Zhang, Chengtao Li, Regina Barzilay, and Kareem Darwish. 2015. Randomized greedy inference for joint segmentation, POS tagging and dependency parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 42–52. http://www.aclweb.org/anthology/N15-1005.

Guangyou Zhou, Jun Zhao, Kang Liu, and Li Cai. 2011. Exploiting web-derived selectional preference to improve statistical dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 1556–1565. http://www.aclweb.org/anthology/P11-1156.