Word-based Japanese typed dependency parsing with grammatical function analysis

Takaaki Tanaka   Nagata Masaaki
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan
{tanaka.takaaki,nagata.masaaki}@lab.ntt.co.jp

Abstract

We present a novel scheme for word-based Japanese typed dependency parser which integrates syntactic structure analysis and grammatical function analysis such as predicate-argument structure analysis. Compared to bunsetsu-based dependency parsing, which is predominantly used in Japanese NLP, it provides a natural way of extracting syntactic constituents, which is useful for downstream applications such as statistical machine translation. It also makes it possible to jointly decide dependency and predicate-argument structure, which is usually implemented as two separate steps. We convert an existing treebank to the new dependency scheme and report parsing results as a baseline for future research. We achieved a better accuracy for assigning function labels than a predicate-argument structure analyzer by using grammatical functions as dependency label.

1 Introduction

The goal of our research is to design a Japanese typed dependency parsing that has sufficient linguistically derived structural and relational information for NLP applications such as statistical machine translation. We focus on the Japanese-specific aspects of designing a kind of Stanford typed dependencies (de Marneffe et al., 2008).

Syntactic structures are usually represented as dependencies between chunks called *bunsetsus*. A bunsetsu is a Japanese grammatical and phonological unit that consists of one or more content words such as a noun, verb, or adverb followed by a sequence of zero or more function words such as auxiliary verbs, postpositional particles, or sentence-final particles. Most publicly available Japanese parsers, including CaboCha [1] (Kudo et al., 2002) and KNP [2] (Kawahara et al., 2006), return bunsetsu-based dependency as syntactic structure. Such parsers are generally highly accurate and have been widely used in various NLP applications.

However, bunsetsu-based representations also have two serious shortcomings: one is the discrepancy between syntactic and semantic units, and the other is insufficient syntactic information (Butler et al., 2012; Tanaka et al., 2013).

Bunsetsu chunks do not always correspond to constituents (e.g. NP, VP), which complicates the task of extracting semantic units from bunsetsu-based representations. This kind of problem often arises in handling such nesting structures as coordinating constructions. For example, there are three dependencies in a sentence (1): a coordinating dependency $b2$ – $b3$ and ordinary dependencies $b1$ – $b3$ and $b3$ – $b4$. In extracting predicate-argument structures, it is not possible to directly extract a coordinated noun phrase ワインと酒 “wine and sake” as a direct object of the verb 飲んだ “drank”. In other words, we need an implicit interpretation rule in order to extract NP in coordinating construction: head bunsetsu $b3$ should be divided into a content word 酒 and a function word の, then the content word should be merged with the dependent bunsetsu $b2$.

(1)	$_{b1}$ 飲んだ	| $_{b2}$ ワイン	と	| $_{b3}$ 酒	の	|$_{b4}$ リスト
	nonda	*wain*	*to*	*sake*	*no*	*risuto*
	drank	wine	CONJ	sake	GEN	list

‘A list of wine and sake that (someone) drank’

Therefore, predicate-argument structure analysis is usually implemented as a post-processor of bunsetsu-based syntactic parser, not just for assigning grammatical functions, but for identifying constituents, such as an analyzer SynCha [3] (Iida

[1] http://taku910.github.io/cabocha/.

[2] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP.

[3] http://www.cl.cs.titech.ac.jp/∼ryu-i/syncha/.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics
and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 237–242,
Beijing, China, July 26-31, 2015. ©2015 Association for Computational Linguistics

	魚 / フライ	を	食べ	た	か / も / しれ / ない	三毛 / 猫
	fish / fry	-ACC	eat	-PAST	may	calico / cat
	“the calico cat that may have eaten fried fish”					
SUW	NN / NN	PCS	VB	AUX	P / P / VB / AUX	NN / NN
LUW	NN	PCS	VB	AUX	AUX	NN

Figure 1: A tokenized and chunked sentence.

et al., 2011), which uses the parsing results from CaboCha. We assume that using a word as a parsing unit instead of a bunsetsu chunk helps to maintain consistency between syntactic structure analysis and predicate-argument structure analysis.

Another problem is that linguistically different constructions share the same representation. The difference of a gapped relative clause and a gapless relative clause is a typical example. In sentences (2) and (3), we cannot discriminate the two relations between bunsetsus $b2$ and $b3$ using unlabeled dependency: the former is a subject-predicate construction of the noun 猫 “cat” and the verb 食べる “eat” (subject gap relative clause) while the latter is not a predicate-argument construction (gapless relative clause).

(2) $_{b1}$ 魚 を |$_{b2}$ 食べ た | $_{b3}$ 猫
sakana o *tabe ta* *neko*
fish ACC eat PAST cat
‘the cat that ate fish’

(3) $_{b1}$ 魚 を | $_{b2}$ 食べ た | $_{b3}$ 話
sakana o *tabe ta* *hanashi*
fish ACC eat PAST story
‘the story about having eaten fish’

We aim to build a Japanese typed dependency scheme that can properly deal with syntactic constituency and grammatical functions in the same representation without implicit interpretation rules. The design of Japanese typed dependencies is described in Section 3, and we present our evaluation of the dependency parsing results for a parser trained with a dependency corpus in Section 4.

2 Related work

Mori et al. (2014) built word-based dependency corpora in Japanese. The reported parsing achieved an unlabeled attachment score of over 90%; however, there was no information on the syntactic relations between the words in this corpus. Uchimoto et al. (2008) also proposed the criteria and definitions of word-level dependency structure mainly for annotation of a spontaneous speech corpus, the Corpus of Spontaneous Japanese (CSJ) (Maekawa et al., 2000), and they do not make a distinction between detailed syntactic functions either.

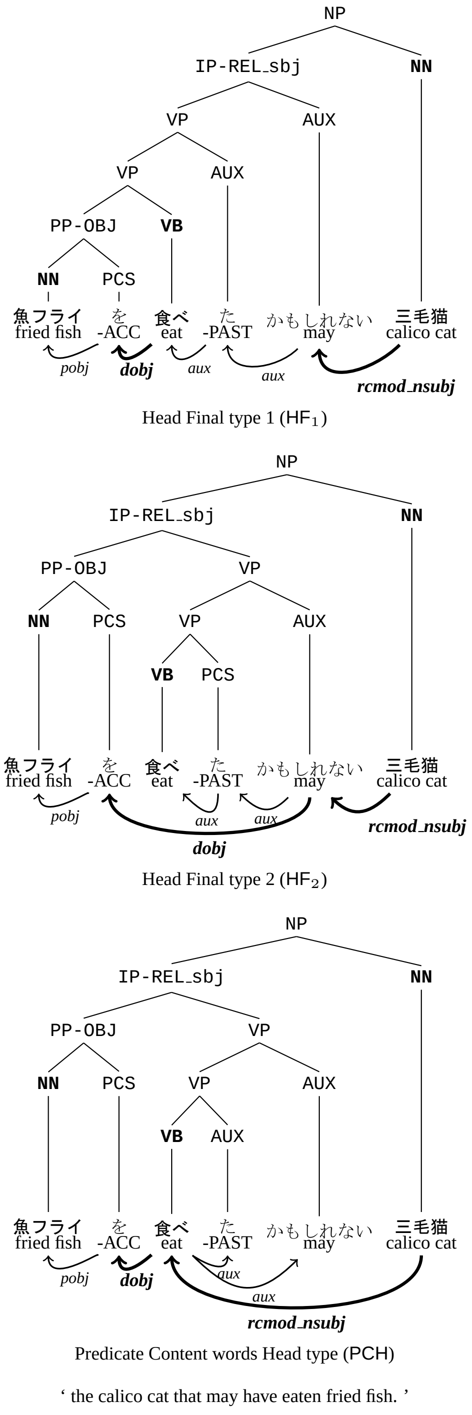

Figure 2: Example structures in three dependency schemes. Boldface words are content words that may be predicates or arguments. Thick lines denote dependencies with types related to predicate-argument structures.

Category	Dep. type	
case (argument)	*nsubj*	subject
	dobj	direct object
	iobj	indirect object
case (adjunct)	*tmod*	temporal
	lmod	locative
gapped relative clause	*rcmod_nsubj*	subject gap relative clause
	rcmod_dobj	direct object gap relative clause
	rcmod_iobj	indirect object gap relative clause
adnominal clause	*ncmod*	gapless relative clause
adverbial clause	*advcl*	
coordinating construction	*conj*	
apposition	*appos*	
function word relation	*aux*	relation between an auxiliary verb and other word
	pobj	relation between a particle and other word

Table 1: Dependency types (excerpt).

We proposed a typed dependency scheme based on the well-known and widely used Stanford typed dependencies (SD), which originated in English and has since been extended to many languages, but not to Japanese. The Universal dependencies (UD) (McDonald et al., 2013; de Marneffe et al., 2014) has been developed based on SD in order to design the cross-linguistically consistent treebank annotation [4]. The UD for Japanese has also been discussed, but no treebanks have been provided yet. We focus on the feasibility of word-based Japanese typed dependency parsing rather than on cross-linguistic consistency. We plan to examine the conversion between UD and our scheme in the future.

3 Typed dependencies in Japanese

To design a scheme of Japanese typed dependencies, there are three essential points: what should be used as parsing units, which dependency scheme is appropriate for Japanese sentence structure, and what should be defined as dependency types.

3.1 Parsing unit

Defining a word unit is indispensable for word-based dependency parsing. However, this is not a trivial question, especially in Japanese, where words are not segmented by white spaces in its orthography. We adopted two types of word units defined by NINJL [5] for building the Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa et al., 2014; Den et al., 2008): Short unit word (SUW) is the shortest token conveying morphological information, and the long unit word (LUW) is the basic unit for parsing, consisting of one or more SUWs. Figure 1 shows example results from the preprocessing of parsing. In the figure, "/" denotes a border of SUWs in an LUW, and "||" denotes a bunsetsu boundary.

3.2 Dependency scheme

Basically, Japanese dependency structure is regarded as an aggregation of pairs of a left-side dependent word and a right-side head word, i.e. right-headed dependency, since Japanese is a head-final language. However, how to analyze a predicate constituent is a matter of debate. We define two types of schemes depending on the structure related to the predicate constituent: first conjoining predicate and arguments, and first conjoining predicate and function words such as auxiliary verbs.

As shown in sentence (4), a predicate bunsetsu consists of a main verb followed by a sequence of auxiliary verbs in Japanese. We consider two ways of constructing a verb phrase (VP). One is first conjoining the main verb and its arguments to construct VP as in sentence (4a), and the other is first conjoining the main verb and auxiliary verbs as in sentence (4b). These two types correspond to sentences (5a) and (5b), respectively, in English.

(4) 猫 が 魚 を 食べ た かもしれない
cat NOM fish ACC eat PAST may
'the cat may have eaten the fish'
a. [[[$_{\text{VP}}$ 猫が 魚を 食べ] た] かもしれない]
S O V aux aux
b. [猫が [魚を [$_{\text{VP}}$ 食べ た かもしれない]]]
S O V aux aux

(5) a. [The cat [may have [$_{\text{VP}}$ eaten the fish]]] .
S aux aux V O
b. [The cat [[$_{\text{VP}}$ may have eaten] the fish]] .
S aux aux V O

The structures in sentences (4a) and (5a) are similar to a structure based on generative grammar. On the other hand, the structures in sentences (4b) and (5b) are similar to the bunsetsu structure.

We defined two dependency schemes **Head Final type 1** (HF$_1$) and **Head Final type 2** (HF$_2$) as shown in Figure 2, which correspond to structures of sentences (4a) and (4b), respectively. Additionally, we introduced **Predicate Content word Head type** (PCH), where a content word (e.g. verb) is treated as a head in a predicate phrase so as to link the predicate to its argument more directly.

3.3 Dependency type

We defined 35 dependency types for Japanese based on SD, where 4-50 types are assigned for syntactic relations in English and other languages.

[4] http://universaldependencies.github.io/docs/.
[5] National Institute for Japanese Language and Linguistics.

LUW (Long Unit Word)		source
l_FORM	form	LUW chunker
l_LEMMA	lemma	LUW chunker
l_UPOS	POS	LUW chunker
l_INFTYPE	inflection type	LUW chunker
l_INFFORM	inflection form	LUW chunker
l_CPOS	non-terminal symbol	*
l_SEMCLASS	semantic class	thesaurus**
l_PNCLASS	NE class	thesaurus**
SUW (Short Unit Word)		
s_FORM_R	form (rightmost)	tokenizer
s_FORM_L	form (leftmost)	tokenizer
s_LEMMA_R	lemma (rightmost)	tokenizer
s_LEMMA_L	lemma (leftmost)	tokenizer
s_UPOS_R	POS	tokenizer
s_CPOS_R	non-terminal symbol	*
s_SEMCLASS_R	semantic class	thesaurus**
s_PNCLASS_R	NE class	thesaurus**

Table 2: Word attributes used for parser features. * 26 non-terminal symbols (e.g. NN, VB) are employed as coarse POS tags (CPOS) from an original treebank. ** Semantic classes SEMCLASS and PNCLASS are used for general nouns and proper nouns, respectively from a Japanese thesaurus (Ikehara et al., 1997) to generalize the nouns.

Table 1 shows the major dependency types. To discriminate between a gapped relative clause and a gapless relative clause as described in Section 1, we assigned two dependency types *rcmod* and *ncmod* respectively. Moreover, we introduced gap information by subdividing *rcmod* into three types to extract predicate-argument relations, while the original SD make no distinction between them.

The labels of case and gapped relative clause enable us to extract predicate-argument structures by simply tracing dependency paths. In the case of HF$_1$ in Figure 2, we find two paths between content words: 魚フライ “fried fish”(NN)←*pobj*←***dobj***← 食べ “eat”(VB) and 食べ (VB)←*aux*←*aux*←***rcmod_nsubj***← 三毛猫 “calico cat”(NN). By marking the dependency types *dobj* and *rcmod_nsubj*, we can extract the arguments for predicate 食べる, i.e., 魚フライ as a direct object and 三毛猫 as a subject.

4 Evaluation

We demonstrated the performance of the typed dependency parsing based on our scheme by using the dependency corpus automatically converted from a constituent treebank and an off-the-self parser.

4.1 Resources

We used a dependency corpus that was converted from the Japanese constituent treebank (Tanaka et al., 2013) built by re-annotating the Kyoto University Text Corpus (Kurohashi et al., 2003) with phrase structure and function labels. The Kyoto corpus consists of approximately 40,000 sentences from newspaper articles, and from these 17,953 sentences have been re-annotated. The treebank is designed to have complete binary trees, which can be easily converted to dependency trees by adapting head rules and dependency-type rules for each partial tree. We divided this corpus into 15,953 sentences (339,573 LUWs) for the training set and 2,000 sentences (41,154 LUWs) for the test set.

4.2 Parser and features

In the analysis process, sentences are first tokenized into SUW and tagged with SUW POS by the morphological analyzer MeCab (Kudo et al., 2004). The LUW analyzer Comainu (Kozawa et al., 2014) chunks the SUW sequences into LUW sequences. We used the MaltParser (Nivre et al., 2007), which marked over 81 % in labeled attachment score (LAS), for English SD. Stack algorithm (projective) and LIBLINEAR were chosen as the parsing algorithm and the learner, respectively. We built and tested the three types of parsing models with the three dependency schemes.

Features of the parsing model are made by combining word attributes as shown in Table 2. We employed SUW-based attributes as well as LUW-based attributes because LUW contains many multiword expressions such as compound nouns, and features combining LUW-based attributes tend to be sparse. The SUW-based attributes are extracted by using the leftmost or rightmost SUW of the target LUW. For instance, for LUW 魚フライ in Figure 1, the SUW-based attributes are s_LEMMA_L (the leftmost SUW's lemma 魚 “fish”) and s_LEMMA_R (the rightmost SUW's lemma フライ “fry”).

4.3 Results

The parsing results for the three dependency schemes are shown in Table 3 (a). The dependency schemes HF$_1$ and HF$_2$ are comparable, but PCH is slightly lower than them, which is probably because PCH is a more complicated structure, having left-to-right dependencies in the predicate phrase, than the head-final types HF$_1$ and HF$_2$. The performances of the LUW-based parsings are considered to be comparable to the results of a bunsetsu-dependency parser CaboCha on the same data set, i.e. a UAS of 92.7%, although we cannot directly compare them due to the difference in parsing units. Table 3 (b) shows the results for each dependency type. The argument types (*nsubj*,

Scheme	UAS	LAS
HF$_1$	94.09	89.49
HF$_2$	**94.21**	**89.66**
PCH	93.53	89.22

(a) Overall results

	F$_1$ score		
dep. type	HF$_1$	HF$_2$	PCH
nsubj	80.47	**82.12**	81.08
dobj	92.06	90.28	**92.29**
iobj	**82.05**	80.22	81.89
tmod	55.54	**56.01**	54.09
lmod	52.10	**53.56**	48.48
rcmod_nsubj	60.38	61.10	**62.95**
rcmod_dobj	28.07	33.33	**39.46**
rcmod_iobj	32.65	33.90	**36.36**
ncmod	82.81	**83.07**	82.94
advcl	65.28	**66.70**	60.69
conj	**70.78**	70.68	69.53
appos	51.11	**57.45**	46.32

(b) Results for each dependency type

Table 3: Parsing results.

Scheme	Precision	Recall	F$_1$ score
HF$_1$	82.1	71.4	76.4
HF$_2$	81.9	67.0	73.7
PCH	**82.5**	**72.4**	**77.1**
SynCha	76.6	65.3	70.5

Table 4: Predicate-argument structure analysis.

dobj and *iobj*) resulted in relatively high scores in comparison to the temporal (*tmod*) and locative (*lmod*) cases. These types are typically labeled as belonging to the postpositional phrase consisting of a noun phrase and particles, and case particles such as が "ga", を "o" and に "ni" strongly suggest an argument by their combination with verbs, while particles に and で "de" are widely used outside the temporal and locative cases.

Predicate-argument structure We extracted predicate-argument structure information as triplets, which are pairs of predicates and arguments connected by a relation, i.e. $(pred, rel, arg)$, from the dependency parsing results by tracing the paths with the argument and gapped relative clause types. $pred$ in a triplet is a verb or an adjective, arg is a head noun of an argument, and rel is nsubj, dobj or iobj.

The gold standard data is built by converting predicate-argument structures in NAIST Text Corpus (Iida et al., 2007) into the above triples. Basically, the cases "ga", "o" and "ni" in the corpus correspond to "nsubj", "dobj“ and "iobj", respectively, however, we should apply the alternative conversion to passive or causative voice, since the annotation is based on active voice. The conversion for case alternation was manually done for each triple. We filtered out the triples including zero pronouns or arguments without the direct dependencies on their predicates from the converted triples, finally 6,435 triplets remained.

Table 4 shows the results of comparing the extracted triples with the gold data. PCH marks the highest score here in spite of getting the lowest score in the parsing results. It is assumed that the characteristics of PCH, where content words tend to be directly linked, are responsible. The table also contains the results of the predicate-argument structure analyzer SynCha. Note that we focus on only the relations between a predicate and its dependents, while SynCha is designed to deal with zero anaphora resolution in addition to predicate-argument structure analysis over syntactic dependencies. Since SynCha uses the syntactic parsing results of CaboCha in a cascaded process, the parsing error may cause conflict between syntactic structure and predicate-argument structure. A typical example is that case where a gapped relative clause modifies a noun phrase A の B "B of A", e.g., [$_{\mathrm{VP}}$ 庭 から 逃げ た] [$_{\mathrm{NP}}$ 猫 の 足跡] "footprints of the cat that escaped from a garden." If the noun A is an argument of a main predicate in a relative clause, the predicate is a dependent of the noun A; however, this is not actually reliable because two analyses are separately processed. There are 75 constructions of this type in the test set; the LUW-based dependency parsing captured 42 correct predicate-argument relations (and dependencies), while the cascaded parsing was limited to obtaining 6 relations.

5 Conclusion

We proposed a scheme of Japanese typed-dependency parsing for dealing with constituents and capturing the grammatical function as a dependency type that bypasses the traditional limitations of bunsetsu-based dependency parsing. The evaluations demonstrated that a word-based dependency parser achieves high accuracies that are comparable to those of a bunsetsu-based dependency parser, and moreover, provides detailed syntactic information such as predicate-argument structures. Recently, discussion has begun toward Universal Dependencies, including Japanese. The work presented here can be viewed as a feasibility study of UD for Japanese. We are planning to port our corpus and compare our scheme with UD to contribute to the improvement of UD for Japanese.

References

Alastair Butler, Zhen Zhou and Kei Yoshimoto. 2012. Problems for successful bunsetsu based parsing and some solutions. In *Proceedings of the Eighteenth Annual Meeting on the Association for Natural Language Processing*, pp. 951–954.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Proceedings of COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation*.

Marie-Catherine de Marneffe, Natalia Silveira, Timothy Dozat, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso and Hideki Ogura. 2008. A proper approach to Japanese morphological analysis: Dictionary, model and evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC-2008)*.

Ryu Iida, Mamoru Komachi, Kentaro Inui and Yuji Matsumoto. 2007. Annotating a Japanese Text Corpus with Predicate-argument and Coreference Relations. In *Proceedings of the the Linguistic Annotation Workshop (LAW '07)*, pp. 132–139.

Ryu Iida and 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.

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.

Shunsuke Kozawa, Kiyotaka Uchimoto and Yasuharu Den. 2014. Adaptation of long-unit-word analysis system to different part-of-speech tagset. In *Journal of Natural Language Processing*, Vol. 21, No. 2, pp. 379–401 (in Japanese).

Taku Kudo, Kaoru Yamamoto and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pp. 230–237.

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.

Kikuo Maekawa, Hanae Koiso, Sasaoki Furui, Hitoshi Isahara. 2000. Spontaneous Speech Corpus of Japanese. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*, pp. 947–952.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka and Yasuharu Den. 2014. Balanced Corpus of Contemporary Written Japanese. *Language Resources and Evaluation*, Vol. 48, pp. 345–371.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of the 51st Annual Meeting of the ACL (ACL 2013)*.

Shunsuke Mori, Hideki Ogura and Teturo Sasada. 2014. A Japanese word dependency corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pp. 753–758.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing, In Journal of Natural Language Engineering, Vol. 13, No. 2, pp. 95–135.

Takaaki Tanaka and Masaaki Nagata. 2013. Constructing a Practical Constituent Parser from a Japanese Treebank with Function Labels. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pp. 108–118.

Kiyotaka Uchimoto and Yasuharu Den . 2008. Word-level Dependency-structure Annotation to Corpus of Spontaneous Japanese and its Application. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2008)*, pp.3118–3122.