NAIST-IS-DD1161009

Doctoral Dissertation

Japanese Predicate Argument Structure Analysis Based on Positional Relations between Predicates and Arguments

Yuta Hayashibe

March 13, 2014

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Yuta Hayashibe

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Satoshi Nakamura	(Co-supervisor)
Associate Professor Masashi Shimbo	(Co-supervisor)
Associate Professor Mamoru Komachi	(Co-supervisor)

Japanese Predicate Argument Structure Analysis Based on Positional Relations between Predicates and Arguments*

Yuta Hayashibe

Abstract

The goal of predicate argument structure analysis is to extract semantic relations such as "who did what to whom" that hold between a predicate and its arguments, constituting a semantic unit of a sentence. It is an important step in semantic oriented Natural Language Processing applications.

In Japanese, an argument is often omitted when we are able to guess what it is from the context. Therefore we should search not only the sentence where the predicate exists but also other sentences for the arguments. Yet, arguments are located nearby the predicate in general. Most previous work has exploited this characteristic to group candidates by positional relations between a predicate and its candidate arguments and then searched for the final candidate using a predetermined priority list of the groups (deterministic model). However, in such analysis, candidates in different groups cannot be compared.

In this dissertation, we propose a Japanese predicate argument structure analysis model which collects the most likely candidates from all the groups and then selects the final candidate from among them. Candidates from low priority groups are also taken into account and we can perform global optimization for the final decision.

Experimental results show that our model outperforms deterministic models. We also discuss future work to enhance performance of predicate argument structure analysis through an analysis of errors by grouping them depending on predicate types.

Keywords:

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1161009, March 13, 2014.

predicate argument structure analysis, positional relations between predicates and arguments, selection-then-classification model, tournament model

述語と項の位置関係に基づく日本語述語項構造解析*

林部 祐太

内容梗概

述語項構造解析の目的は，述語とそれらの項を文の意味的な構成単位として，文章から「誰が何をどうした」という意味的な関係を抽出することである．これは，機械翻訳や自動要約などの自然言語処理の応用において重要なタスクの 1 つである．

日本語では文脈から推測が可能であるとき，項はしばしば省略される．そのため，述語が存在する文以外も項を探索する必要がある．一般に，項は述語に近いところにあるという特性がある．したがって，従来の述語項構造解析の研究の多くは，候補を述語との位置関係でグループ分けし，あらかじめ求めておいたグループ間の優先順序に従って正解項を探索してきた．しかしながら，その方法には異なるグループに属する候補同士の比較ができないという問題がある．

そこで我々は，異なるグループごとに最尤候補を選出し，それらの中から最終的な出力を決めるモデルを提案する．このモデルは優先度の高いグループに属する候補以外も参照することによって最終的な決定を行うことができ，全体的な最適化が可能である．

実験では，提案手法は優先順序に従う解析よりも精度が向上することを確認した．そして，述語項構造解析の精度を向上させるために必要な今後の課題について，述語の種類に応じて分析し議論する．

キーワード

述語項構造解析，項と述語の位置関係，探索先行分類型モデル，トーナメントモデル

*奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文, NAIST-IS-DD1161009, 2014 年 3 月 13 日.

Acknowledgments

主指導教員の松本裕治教授は，なかなか博士論文の執筆を始めなかった私を根気強く最後まで見守ってくださいました．心から感謝いたします．博士論文には直接結びつかなかったテーマもありましたが，在学中に色々な研究テーマに自由に挑戦させて頂けたおかげで“失敗”もすることもできました．それはこれからの研究者・技術者人生にとって大切な経験になったと思っています．修士課程から引き続いて5年間ご指導してくださり，どうもありがとうございました．

中村哲教授には副指導教員を引き受けていただきありがとうございました．公聴会や最終審査で頂いたコメントを励みに，博士論文の最終提出まで，今の自分の精一杯の力で頑張りました．良い研究をできるように精進していきたいと思っております．

新保仁准教授から頂くコメントはいつもハッとさせられるものばかりでした．感謝いたします．新保さんのような深い洞察をできるように，これからも頑張りたいと思います．

小町守准教授には，修士課程から引き続いて研究や研究生活の指導をしていただけたことを心からお礼申し上げます．記憶に残っているお話は色々とありますが，特に，NAIST に入学したあたりにお話くださった「修士課程は問題解決能力，博士課程は問題発見能力を鍛える所」ということの真髄が，ようやく今になって分かってきた気がします．良い問題を世界に提起して，その分野の第一人者になることが今後の目標です．

北川祐子秘書には事務手続きでお世話になりました．事務作業に煩わされずに研究ができたのは，北川さんのおかげです．どうもありがとうございました．

在学中には楽天技術研究所 New York でリサーチインターンの機会を頂きました．自分の視野を広められた，とても楽しい思い出です．特に，インターンをお引き受けくださった関根聡所長と，メンターとしてお世話して頂いた萩原正人博士にお礼申し上げます．わずか2ヶ月強で実装と論文の投稿ができたのは，お二人のおかげです．どうもありがとうございました．

この論文は多くの方のおかげで書くことができました．ウェブから収集した日本語文データを使用させてくださった河原大輔准教授，(Taira, Fujita, and Nagata 2008) の詳細なアルゴリズムを教えてくださった平博順博士にお礼申し上げます．ジャーナル論文の査読で多数の有益なコメントをくださった匿名の査読者の方々に深く感謝いたします．そして，英語表現をチェックしていただいた Kevin Duh 助教に感謝いたします．

公益財団法人伊藤謝恩育英財団には大学学部・修士課程のときに奨学金をご支援頂きました．おかげさまで博士課程までの学生生活を全うすることができました．ありがとうございました．また，奈良先端科学技術大学院大学情報科学研究科には，CICP で研究費をご支援を頂きました．感謝いたします．

松本研の仲間たちには大変お世話になり，いつも多くの刺激を与えてくださいました．特に，去年の 3 月に 7 階から 5 階の学生室に移ってから一緒に 1 年間を過ごしたメンバー（岡照晃くん，西村駿人くん，久本空海くん，光瀬智哉くん，三澤賢祐くん，水本智也くん，吉本一平くん，白井尊昭くん，椿真史くん，Frances Yung さん）に感謝します．この 1 年間は非常に濃密で，今後忘れることができないものです．学生室はみんなの集まる場所で，寮と学生室とでシェアハウスのような生活だったなと思います．研究の合間に，みんなとご飯を食べに行ったり，温泉に行ったり，雑談したりして，とても楽しかったです．実験や論文で本当に辛い時期もありましたが，孤独感や絶望感に潰されずにこうやって乗り越えられたのは，みんなのおかげです．本当にありがとう．

岡照晃くん：岡くんの研究分野に対する豊富な知識と熱い情熱には尊敬の念を禁じ得ません．話すたびに，もっと色々と学ばなければという思いが湧いてきます．これからも，色々と勉強会やディスカッションしましょう．

西村駿人くん：よく岡くんと三人で一緒にご飯や風呂に行きましたね．あれは，一日の楽しみの時間でした．

久本空海くん：一緒に色々な話をしましたね．君が活き活きと輝ける場に巡り会えると信じてます．また面白い話を聞かせてね．

光瀬智哉くん，三澤賢祐くん，水本智也くん，吉本一平くん：一緒に Shared Task に取り組みましたね．その後学位論文に取り組んだ M2 のみんなは，一緒に苦楽を味わった戦友のように思っています

白井尊昭くん，椿真史くん，Frances Yung さん：料理して，お酒を呑んだりしましたね．楽しい思い出です．

その他にも，博士課程在学中に多くの方のお世話になりました．

八重島充弘くんとは，よく一緒に呑んだり遊んだりしました．僕が辛い時，色々と励ましの言葉をくれて嬉しかったのを覚えています．本当にありがとう．情に厚いところ，フットワークの軽いところ，尊敬しています．これからも，よろしくね．

金田朋子先生のお話は，研究生活の良いリフレッシュになりました．どうもありがとうございました．先生には，楽しく生きる生き方，というものも教えて頂きました．笑顔を絶やさないところ，ちょっとしたことでも喜ぶ無邪気なところ，自分の気持ちに対する素直なところ，ポジティブなところ．私はそういったところを尊敬しており，見習っていきたいと思っています．

最後に，堺でいつも祈り支えてくれた私の愛する家族に感謝します．

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Predicate argument structure constitutes a semantic unit of a sentence. *Predicates* are main parts of sentences which construct sentences with other elements (Japanese descriptive grammar research group 2010). Predicates require compliments (including subjects) to make sense which are called *arguments*. We call the semantic relation between a predicate and its argument as a *case*. For example, “書く” (write) in Example 1 is a predicate and “私” (I) and “手紙” (letter) are arguments of the predicate. There are various sets of the cases. “Ga”-case (nominative) and “agent” can be assigned to “私” (I) and “o”-case (accusative) and “theme” can be assigned to “手紙” (letter).

Example 1:

I wa-particle the letter o-particle wrote
私 は その手紙 を 書いた。
(I wrote a letter.)

The goal of *predicate argument structure analysis* is to extract semantic relations from natural language sentences such as “who did what to whom” that hold between a predicate and its arguments. This representation has an advantage to be represented in the same structure regardless of its syntactic realization.

Example 2:

the letter wa-particle I ga-particle wrote and post box ni-particle drop
その手紙 は 私 が 書い て ポスト に 投函した。

(I wrote a letter and drop it into a post box.)

In Example 2, the predicate “書く” (write) and its arguments are in a different syntactic relation to the one in Example 1, while the predicate argument structure does not change. The arguments are given the same labels in the predicate argument structure of “書く” (write) in Example 1. Therefore predicate argument structure analysis is an important step in semantic oriented Natural Language Processing applications, for example information extraction (Surdeanu, Harabagiu, Williams, and Aarseth 2003), question answering (Shen and Lapata 2007), statistical machine translation (Wu and Fung 2009), and recognizing textual entailment (Wang and Zhang 2009).

1.2 Research Purpose

In Japanese, an argument is often omitted[1] when we are able to guess what it is from the context. The omitted argument is called a *zero-pronoun* and its referring element an *antecedent*. Such linguistic phenomenon is called *zero anaphora*. Some previous work regarded predicate argument structure analysis as zero anaphora resolution (Kawahara and Kurohashi 2004; Sasano and Kurohashi 2011).

Yet, arguments are located “nearby” the predicate in general. Arguments in the same sentence of the predicate have not so large distance on the syntactic tree. In other words, most arguments directly depend on the predicates. Even when they are in other sentences, Imamura, Saito, and Izumi (2009) reported the coverage of arguments which appeared in less than one sentence before is 62.5% of the referents of zero pronouns.

Previous work has exploited this characteristic. Kawahara and Kurohashi (2004) and Taira et al. (2008) grouped candidates by positional relations between a predicate and its candidate arguments and then searched for the final candidate using a predetermined priority list of the groups. Iida, Inui, and Matsumoto (2007) searched the sentence where the predicate appears in the argument first. In these previous work, candidates in different groups are not compared directly.

We propose a Japanese predicate argument structure analysis model which collects the most likely candidates from all the groups and then selects the final candidate from among them. We can take candidates with less priority into account before making the final decision in order to perform global optimization.

[1] In this dissertation, we define *omission* as absence of an argument that has direct syntactic relation with a predicate.

1.3 Organization of This Dissertation

The organization of this dissertation is as follows. In Chapter 2, we show Japanese corpora tagged with predicate argument structures. In Chapter 3, we overview previous work on Japanese predicate argument structure analysis, especially from the standpoint of the treatment of the positional relations between arguments and predicates. In Chapter 4, we propose a method which collects the most likely candidates from all the groups and then selects the final candidate from among them. We then evaluate the model in Chapter 5. In Chapter 6, we analyze errors and explore problems to be considered. Chapter 7 summarizes our research and future directions.

Chapter 2

Japanese Predicate Argument Structure Corpus

For annotation of predicate argument structures, there are two types of abstraction level of labels of relations between predicates and arguments, *surface case* and *deep case*. While surface case uses syntactic markers like particles, deep case utilizes more labels. For example, both “Taro” in Example 3 and Example 4 are nominative in surface case. In a deep case, the former is “agent” and the latter is “experiencer.” Deep cases are useful for deeper semantic understanding. Labels in deep case are called *semantic roles*, and analysis in deep case is also called *semantic role labeling*.

Example 3:

Taro	wa-particle	curry	o-particle	ate
太郎	は	カレー	を	食べた.

(Taro ate curry.)

Example 4:

Taro	wa-particle	moved
太郎	は	感動した.

(Taro moved.)

In English, major corpora such as NomBank (Meyers, Reeves, Macleod, Szekely, Zielinska, Young, and Grishman 2004), PropBank (Palmer, Gildea, and Kingsbury 2005), FrameNet (Ruppenhofer, Ellsworth, Petruck, Johnson, and Scheffczyk 2006), and OntoNotes (Hovy, Marcus, Palmer, Ramshaw, and Weischedel 2006) utilize semantic roles. Many studies on semantic role labeling have been performed by using them.

On the other hand, in Japanese, most studies on predicate argument structure analysis principally target surface cases. This is mainly because of insufficient quality of training in machine learning approach. For example, inter-sentential arguments in both EDR Japanese corpus and GDA corpus are not annotated as we describe later. Therefore, it is necessary to annotate deep cases to such arguments too for semantic role labeling in Japanese.

One promising direction is additional annotation of deep cases to the existing corpora annotated with surface cases. Iida, Komachi, Inui, and Matsumoto (2007) claim that in Japanese, the mapping from surface cases to deep cases tends to be reasonably straightforward if a semantically rich lexicon of verbs like the VerbNet (Kipper, Dang, and Palmer 2000) is available. In Japanese, "Predicate-Argument Structure Thesaurus"[1] (Takeuchi, Inui, Takeuchi, and Fujita 2010) has the mapping between surface cases and deep cases.

In the rest of this chapter, we introduce Japanese predicate argument structure corpora annotated with surface cases in Section 2.1 and corpora annotated with deep cases in Section 2.2

2.1 Corpora Annotated with Surface Case

2.1.1 Kyoto University Text Corpus

Kyoto University Text Corpus[2](Kurohashi and Nagao 2003) is a text corpus which has manual morphological and syntactic annotations. It consists of approximately 40,000 sentences from Mainichi newspaper in 1995. Half of them are articles from January first to 17th and another half are editorials from January to December. Actually, the annotation has carried out to correct automatically analyzed results by the morphological analyzer JUMAN[3] and the dependency analyzer KNP[4]. They have simultaneously improved the analyzers during the manual correction.

They have released their annotations in a gradual manner. Annotations to about 10,000 sentences were released in September 1997 as Version 1.0, and about 20,000

[1]http://cl.it.okayama-u.ac.jp/rsc/data
[2]http://nlp.ist.i.kyoto-u.ac.jp/index.php?京都大学テキストコーパス
[3]http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[4]http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

sentences in June 1998 as Version 2.0. The whole annotations were released in July 2000 as Version 3.0.

They also annotated predicate argument structures and co-references to approximately 5,000 sentences and released them as Version 4.0 in April 2005 (Kawahara, Kurohashi, and Hasida 2002). It adopted syntactic case for the label of predicate argument relations such as case particles (ガ[ga], ヲ[o], カラ[kara] and so on) and collocation words (ニツイテ[nitsuite] and so on) for bare forms of predicates.

2.1.2 KNB Corpus

Kyoto University and NTT Blog Corpus (KNB Corpus)[5] contains 4,186 sentences in 249 blog articles (Hashimoto, Kurohashi, Kawahara, Shinzato, and Masaaki 2011). It is released in September 2009. The articles contain four topics; sightseeing in Kyoto, mobile phone, sports, and eating. It is annotated with sentiment information and grammatical information about morphology, dependency, case, and co-reference. The annotation schema is almost the same as one of Kyoto University Text Corpus.

2.1.3 NAIST Text Corpus

NAIST Text Corpus[6] (Iida et al. 2007) is based on Kyoto University Text Corpus. While only 5,000 sentences are targeted for annotation of predicate argument structures and co-references in the original corpus, all sentences are targeted in NAIST Text Corpus. Additionally, they also annotated event-noun argument relations.

The main differences are following two things. First, they tagged only three major cases: ガ[ga] (nominative), ヲ[o] (dative), and ニ[ni] (accusative). They experimentally annotated other cases (カラ[kara], ヘ[e], ト[to], ヨリ[yori], マデ[made], デ[de]) to intra-sentential arguments in 136 sentences (Iida, Komachi, Inoue, Inui, and Matsumoto 2010). By the result of the annotation, most arguments depend on predicates and appear the particles to be annotated. Therefore they came to the conclusion that such cases can be tagged automatically.

---

[5] http://nlp.ist.i.kyoto-u.ac.jp/kuntt/#ga739fe2

[6] http://cl.naist.jp/nldata/corpus/

Table 2.1: Comparison of predicate argument structure analysis of nominative case

Sub Corpus	Media	abbreviated name	Non-“core” samples	“Core” samples
Publication	Newspapers	PN	1,133	340
	Magazines	PM	1,910	86
	Books	PB	10,034	83
Library	books	LB	10,551	-
Special-purpose	Whitepaper	OW	1,438	62
	Best selling books	OB	1,390	-
	Internet Q&A	OC	90,507	938
	Blogs	OY	52,209	471
	Law	OL	346	-
	Diet minutes	OM	159	-
	PR paper	OP	354	-
	Textbook	OT	412	-
	Verse	OV	252	-

Second, they tagged predicate argument relations not for bare forms but for base forms of the predicates. They claim that it is more useful because cases of a predicate do not change even in case alternation such as passivization and causativization.

NAIST Text Corpus has multiple versions. Initial version 1.0β was released in October 2006. Though the latest version 1.5 was released in August 2010, version 1.4β released in August 2007 is used widely for research. It is based on Kyoto Text Corpus 3.0[7].

2.1.4 BCCWJ-PAS

Balanced Corpus of Contemporary Written Japanese (BCCWJ)[8] is a balanced corpus of one hundred million words. It consists of a lot of media as shown in Table 2.1. All sentences in BCCWJ are tagged with morphological information by automatic analyzer. About one percent of the whole data, called the “core” data, is manually annotated with morphological and syntactic information.

Komachi and Iida (2011) annotated[9] predicate argument structures, event-noun argument structures, and co-reference relations to “core” sentences in PN, PB, OW and OC. As of March 2014, only annotation for OC is made public.

The annotation schema is almost the same as that of NAIST Text Corpus. In contrast to the work of annotation to NAIST Text Corpus, they exploited Lexical Conceptual

[7] http://nlp.ist.i.kyoto-u.ac.jp/nl-resource/corpus/KyotoCorpus3.0.tar.gz
[8] http://www.ninjal.ac.jp/corpus_center/bccwj/
[9] http://cl.naist.jp/nldata/bccwj/pas/

```
[
    [main 14:走:103ac8]
    [attribute past]
    [agent @1:c#nil:]
    [manner 9:直ちに:3ced5f]
    [goal [
        [main 12:結成:0ef56b]
        [object 10:温知会:``=Z 温知会という組織'']
    ]]
    [sequence [
        [main 5:追放:3cf185]
        [source 3:ムラ:0f2459]
    ]]
    [and *PREVIOUS-SENTENCE]
]
```

Figure 2.1: A semantic frame in EDR Japanese Corpus

Structure dictionary by Takeuchi[10] when they need obligatory judgment in their annotation. Komachi and Iida (2011) reported this improves agreement between annotators for ambiguous cases.

2.2 Corpora Annotated with Deep Case

2.2.1 EDR Japanese Corpus

EDR Japanese Corpus, released in 1995, is a part of the EDR Electronic Dictionary[11]. It consists of approximately 20,000 sentences from dictionaries, newspapers, and magazines. The corpus has morphological, syntactic and semantic information such as word sense defined in "EDR Thesaurus" and semantic frames.

Example 5:

and		village	o-particle	be expelled		immediately	Onwakai	no-paticle	formation	ni-paticle
そして	、	ムラ	を	追放され	、	直ちに	温知会	の	結成	に

[10]http://cl.it.okayama-u.ac.jp/rsc/lcs

[11]http://www2.nict.go.jp/out-promotion/techtransfer/EDR

```
<su>
  <adp><persnamep>太郎</persnamep><ad>は</ad></adp>
  <adp>
    <np>
      <vp><adp><vp>慌て</vp><ad>て</ad></adp><v>逃げる</v></vp>
      <n>男</n>
    </np>
    <ad>を</ad>
  </adp>
  <v>追いかける</v>
</su>
```

Figure 2.2: An annotation in the GDA format

do
走る 。
(He was expelled to the village. Immediately, he formed Onwakai.)

Figure 2.1 shows a semantic frame representation of the sentence in Example 5. The leaves are tuples of conceptual connection (so-called semantic roles such as object and source), the number of word, surface form, and meaning id. Most elements (not every element) in a sentence constitute a frame.

“走,” “結成,” and “追放” in Figure 2.1 are predicates and the relations between their arguments are annotated. Therefore, we can regard this as a kind of predicate argument structure tagged corpus with deep case. However, note that inter-sentential arguments are not part of the case frames.

2.2.2 GDA Corpus

GDA (Global Document Annotation) Corpus[12] is a text corpus which has manual morphological, syntactic, semantic annotations. It comprises of approximately 37,000 sentences from Mainichi newspaper in 1994.

They are annotated in the GDA format developed by Hashida (2005) which is a kind of XML. Morphologies are tagged by many tags like noun tag <n> and adjective tag

[12] http://www.gsk.or.jp/catalog/gsk2009-b/

`<aj>`. A dependency relation is also represented by trees of XML tags as shown in Figure 2.2.

Semantic annotations include word sense, anaphora, co-reference and semantic relationship between morphologies. They define approximately 100 semantic relations such as “agent,” “experiencer,” and “argument.”

While they define annotation schema for inter-sentential arguments, they are not annotated in the corpus as noted in (Iida et al. 2007).

Chapter 3

Previous Work on Japanese Predicate Argument Structure Analysis

In this chapter, we overview previous work on Japanese predicate argument structure analysis especially from the standpoint of the treatment of positional relations between an argument and its predicate. We summarize them in Table 3.1.

3.1 Use of Positional Relations in Deterministic Analysis

3.1.1 Order by Statistical Score Approach

Kawahara and Kurohashi (2004) divided the Japanese predicate argument structure analysis into two stages; zero-pronoun detection stage and antecedent identification stage. In the antecedent identification stage, they used a search order statistically computed from a corpus in advance.

First, the approach detects zero-pronouns using case analysis based on a case frame dictionary. Then, they look for the antecedent of a zero-pronoun by following a predefined search order. For every candidate, they compute similarity between a candidate of a case frame and judge whether it is feasible or not with a binary classifier. They select the first candidate in the order which satisfies the following two conditions; the similarity score exceeds a threshold; and the classifier judges it is feasible for the antecedent of the zero-pronoun.

[1] This does not search for inter-sentential candidates.

Table 3.1: Summary of previous work and proposed method

	The number of positional relation types	Use different argument identification models by positional relations	Deterministic analysis
Kawahara and Kurohashi (2004)	20		√
Iida et al. (2007)	2	√	√
Taira et al. (2008)	7	√	√
Sasano and Kurohashi (2011)	-		√
Imamura et al. (2009)	-		
Yoshikawa, Asahara, and Matsumoto (2011)[1]	-		
Proposed method	3	√	

There is only one common classifier regardless of the positional group. Features for the classifier are the similarity, POS and so on.

They categorized arguments into 20 types (they called these "location classes") according to the sentence and document structure such as sub-clause, main-clause as shown in Table 3.2.

Using Kyoto Text Corpus (Kawahara et al. 2002), they calculate score of location class L as follows:

$$\frac{\text{\# of antecedents in L}}{\text{\# of possible antecedents in L}} \tag{3.1}$$

The scores mean how likely the class tends to have an antecedent. They sorted classes by using the scores in descending order, and then defined search order for each case.

> Nominative case: L6, L1, L2, L3, L4, L7, L10, L5, L8, L14, L12, L9, L11, L15, L17, L16, L13, L18, L20, L19
>
> Accusative case: L5, L14, L6, L4, L17, L3, L7, L2, L9, L15, L10, L8, L20, L1, L12, L18, L13, L11, L19, L16
>
> Dative case: L6, L10, L4, L14, L5, L2, L17, L9, L15, L3, L16, L12, L20, L1, L8, L11, L7, L18, L13, L19

Table 3.2: Location classes of antecedents defined by Kawahara and Kurohashi (2004). Vz means a predicate that has a zero pronoun. Va is quoted predicate whose case component is an antecedent.

the sentence under consideration	Vz and Va are conjunctive	Vz constitutes the main clause.
L1 case components of “parent predicate of Vz”		√
L2 case components of “parent predicate of Vz”		
L3 case components of “parent predicate of Vz”	√	√
L4 case components of “parent predicate of Vz”	√	
L5 case components of “child predicate of Vz”		
L6 case components of “child predicate of Vz”	√	
L7 case components of “parent predicate of parent noun phrase of Vz”		√
L8 case components of “parent predicate of parent noun phrase of Vz”		
L9 case components of “parent predicate of parent predicate of Vz”		√
L10 case components of “parent predicate of parent predicate of Vz”		
L11 case components of “predicate of main clause”		√
L12 case components of “predicate of subordinate clause depending on main clause”		
L13 other noun phrases following Vz		
L14 other noun phrases preceding Vz		
1 sentence before		
L15 case components of “predicate of main clause”		√
L16 case components of “predicate of subordinate clause depending on main clause”		
L17 other noun phrases		
2 sentence before		
L18 case components of “predicate of main clause”		√
L19 case components of “predicate of subordinate clause depending on main clause”		
L20 other noun phrases		

3.1.2 “Inter-sentential Candidate First” Approach

Iida et al. (2007) exploited syntactic patterns of zero-pronouns and their antecedents by using the boosting-based algorithm BACT (Kudo and Matsumoto 2004), which is designed to learn subtrees useful for classification.

BACT accepts tree structure data as its input and acquire the subtrees which are useful for classification from all subtrees. Iida et al. (2007) generated a tree for the input of BACT by adding the syntactic dependency tree of the sentence and relational features between antecedent candidates and zero-pronouns to a root node.

Because it is impossible to use dependency relation for inter-sentential antecedent search, they adopted the following procedure with two antecedent identification models (M_{10} and M_{20}) and two anaphoricity determination models (M_{11} and M_{21}).

1. Identification of the most likely intra-sentential antecedent C_1^* by the model M_{10}

2. Calculation of the anaphoricity score p_1 of C_1^* by the intra-sentential anaphoricity determination model M_{11}. If p_1 exceeds the predefined threshold θ_{intra}, the system returns C_1^* as the antecedent. Otherwise, go to step 3.

3. Identification of the most likely inter-sentential antecedent C_2^* by the model M_{20}.

4. Calculation of the anaphoricity score p_2 of C_2^* by the inter-sentential anaphoricity determination model M_{21}. If p_2 exceeds the predefined threshold θ_{inter}, the system returns C_2^* as the antecedent. Otherwise, the system answers there is no antecedent.

The models $M_{10}, \cdots, M_{21}$ are trained with BACT and parameters θ_{intra} and θ_{inter} are estimated with development data. This approach does not refer inter-sentential candidates when it identifies the most likely intra-sentential antecedent and judges anaphoricity.

3.1.3 Empirical Order Approach

Taira et al. (2008) proposed an approach which simultaneously analyzes all cases by using decision lists. Decision list is a set of rules with application order. Its readability of learned lists is high and its interpretation by human is also easy.

They combined four characteristics for a rule of the decision lists.

- Positional relation of the target predicate and a candidate
- Functional word of a candidate
- Voice of the predicate
- One of the following characteristics of a candidate
  - Head word
  - Generalization levels (words, semantic categories, parts of speech)
  - POS

Then they treated one combination as one feature for Support Vector Machine and for each predicate they learned the weights of the features by SVMs. Finally, the features sorted by the weight become the decision list for a predicate. In other words, one feature represents one rule of a decision list.

They made a word a unit of argument. They defined following seven types of positional relations between arguments and predicates based on their syntactic dependency relations. Note that fw and bw are optional types.

- Incoming Connection Type (ic) The bunsetsu[2] phrase which includes an argument depends on one which includes a predicate
  Japan and America negotiation ga made progress
  日 米 交 渉 nominative が 進展した
  (A progress of negotiation between Japan and America in the negotiations has made)

- Outgoing Connection Type (oc) The bunsetsu phrase which includes a predicate depends on one which includes an argument
  impulse purchase did newly-published book
  衝 動 買 い nominative した 新 刊 本
  (A newly-published book which I bought from impulse)

- Within the Same Phrase Type (sc) An argument and a predicate appear in the same bunsetsu phrase
  Japan America negotiation
  日 nominative 米 交 渉
  (negotiation between Japan and America)

- Connection into Other Case role Types (ga_c, wo_c, ni_c) The bunsetsu phrase which includes an argument depends indirectly on another bunsetsu which includes a predicate via another case argument
  Tom to friend by persuasion
  トム dative, ga_c への 友人 nominative による 説 得
  (persuasion of Tom by his friend)

- Non-connection Type (nc) An argument appears in the different sentence where a predicate does

- Forward Type (fw) An argument appears before a predicate in the document.

[2]Bunsetsu is a basic grammatical unit of Japanese consisted of one content word and zero or more functional words.

- Backward Type (bw) An argument appears after a predicate in the document.

The analysis is performed in the following procedure for each predicate.

1. Search ic, oc, ga_c, wo_c, ni_c for arguments by using each decision list.
2. For case(s) whose arguments are determined in the step 1, search sc with its decision list.
3. For case(s) whose arguments are determined in the step 2, count ratio of argument existence in the target case of the target predicate in a corpus. If the ratio exceeds 50%, search nc, fw, and bw with eash decision list.

We can say this method empirically defines search order as follows:
ic, oc, ga_c, wo_c, ni_c > sc >> nc, fw, bw
There is no priority between ic, oc, ga_c, wo_c, and ni_c.

This model has an advantage to exploit connections into arguments in other cases. Therefore it has a chance to optimize argument assignments in the whole predicate argument structures. However, this model judges how likely the target candidate is an argument only using the candidate and the predicate. Therefore, it does not refer to all candidates for the final output and does not perform relative comparison between candidates.

3.1.4 "Direct Dependency First" Approach

Sasano and Kurohashi (2011) proposed a method which first generates case frame candidates by the following procedure.

- The system collects case frames related to the target predicate.
- It combines each case frame candidate and the elements which depend on the predicate. When a case can have several candidate arguments, it generates all possible combinations.
- It fills all unfilled cases in generated combinations with candidates which do not have direct dependencies with the predicate.

The method computes probabilities of all generated candidates with a log-linear model, and then outputs the candidate which has the highest probability. They use semantic class features, named entity features, and positional binary features shown in Table 3.3.

This method jointly analyzes all cases of a predicate. However, it is impossible to compare a candidate which has a direct dependency with the target predicate and a candidate which does not have a direct dependency with the target predicate.

3.2 Use of Positional Relations as Features

Some works have utilized tendency of argument in the view of the positional relations as features not as deterministic analysis.

3.2.1 Maximum Entropy Model Approach

Imamura et al. (2009) proposed a discriminative approach based on the maximum entropy model. Their approach utilizes features related to positional relations and syntactic relations between a candidate and a predicate, but does not divide analysis based on the positional relations. The method adds a special noun phrase NULL which means the predicate does not have an argument, and then identifies the most likely candidate. In order to reduce the number of candidates for the search, they eliminated all the candidates which appear three or more sentences before the one where the predicate appears, except for the candidates which are identified as arguments of other predicates. This method only needs one model for every case. However, it cannot use features of relations between candidates.

3.2.2 Markov Logic Approach

Yoshikawa et al. (2011) proposed an approach using Markov Logic, which jointly analyzes predicate argument structures of all predicates in the same sentence. Markov Logic is based on first-order predicate logic and Markov Networks. It is a framework of statistical relational learning which accepts an inconsistent set of first-order predicate logic formulas with a certain penalty. It is a big advantage for Markov Logic to be able to decide several logical formulas simultaneously. A decision what argument is proper can affect other argument strictures, and the opposite can be equally true. In such situation, it needs to identify the best combination of argument assignments.

However, the number of possible combination is expected to be very large. They jointly reduced candidates while argument identification. Yet it costs considerable time to execute the system. Therefore they excluded all inter-sentential candidates because of complexity of computation.

They used features (observed predicates) like lexical category of candidates, syntactic relations and so on. The syntactic relations between candidates are expressed as whether they are in direct dependency, sibling or ancestor.

Table 3.3: Positional binary features exploited by Sasano and Kurohashi (2011)

For intra-sentential candidates (64)	
Itopic	It appears with a particle “は”
IP-self	The predicate depends on it. (Parent)
IC-self	It depends on the predicate. (Child)
IGP-self	The predicate depends on the phrase which depends on it. (Grand-Parent)
IGC-self	It depends on the phrase which depends on the predicate. (Grand-Child)
⋮	⋮
B-self	It precedes the predicate without the relations listed above (Before)
IA-self	It follows the predicate without the relations listed above (After)
IP-*ga*-ov	It is in the ga-case of a predicate which is depended by the predicate without omission
IP-*ga*-om	It is in the ga-case of a predicate which is depended by the predicate with omission
IP-*o*-ov	It is in the o-case of a predicate which is depended by the predicate without omission
⋮	⋮
IP-*ga*-ov	It is in the ga-case of a predicate which is depended by the phrase depended by the predicate without omission
⋮	⋮
For inter-sentential candidates (21)	
B1	It appears in the sentence one sentence before the predicate.
B1-*ga*-ov	B1 and It is in the ga-case of a predicate without omission.
B1-*ga*-om	B1 and It is in the ga-case of a predicate with omission.
B1-*o*-ov	B1 and It is in the o-case of a predicate without omission.
B1-*o*-om	B1 and It is in the o-case of a predicate with omission.
⋮	⋮
B2	It appears in the sentence two sentence before the predicate.
B2-*ga*-ov	B2 and It is in the ga-case of a predicate without omission.
B2-*ga*-om	B2 and It is in the ga-case of a predicate with omission.
B2-*o*-ov	B2 and It is in the o-case of a predicate without omission.
B2-*o*-om	B2 and It is in the o-case of a predicate with omission.
⋮	⋮
B3	It appears in the sentence three sentence before the predicate.
⋮	⋮

Chapter 4

Japanese Predicate Argument Structure Analysis Comparing Candidates in Different Positional Relation

Deterministic approaches discussed in Chapter 3 do not check candidates in lower priority positional groups, because they check candidates in higher priority positional groups first and finish search when the most likely candidate is found. However, these methods have difficulties in identifying arguments in the lower priority positional groups in return for high performance for arguments in higher priority positional groups. Additionally, the final judgment should be done after search of all candidates to enhance overall performance.

In this chapter, we propose a *Selection-and-Knockout approach* to predicate argument structure analysis. This approach comprises two phases; the *selection* phase and the *knockout* phase. Namely, this approach gathers the most likely candidates from all the groups in the first selection phase, and then selects the final candidate amongst them in the second knockout phase. This approach trains and uses different models for different cases.

4.1 Classification of Arguments According to Their *Positional Relation* between Predicates

In this dissertation, we classify arguments into four categories according to their *positional relation* between predicates: intra-sentential arguments (those that have direct syntactic dependency with the predicates[1]), zero intra-sentential arguments (those appearing as zero-pronouns but having their antecedents in the same sentence), inter-sentential arguments (those appearing as zero-pronouns and their antecedents are not in the same sentence) and exophora arguments (those not appearing explicitly in the document). In this dissertation, we use *INTRA_D, INTRA_Z*, *INTER*, and *EXO* respectively as a shorthand for these argument categories.

Furthermore, we consider that a predicate has ARG_{NULL} as an argument in a case when the predicate does not have an argument in the case. We also consider that its positional relation between the predicate is *NULL*. In this dissertation, we do not discriminate EXO and NULL, and call them NO-ARG.

For instance, take “コロッケ” (croquettes) in the first sentence in Example 6. It is the argument of the “o”-case (accusative) for predicate “受け取った” (received), and hence falls into the INTRA_D category. It can be classified as the INTRA_Z category in terms of predicate “食べた” (ate), for which it is referred to by a zero pronoun for the “o”-case. “飲んだ” (drank) has an argument “彼女” (she) with “ga”-case (nominative) as INTER and ARG_{NULL} with “ni”-case (dative).

Example 6:

croquette	o-particle	received	she	wa-particle	in hurry	ate
コロッケ	を	受け取った	彼女	は	急いで	食べた.

(She received a croquette and ate it in hurry.)

orange juice	mo-particle	drank
オレンジジュース	も	飲んだ.

(She drank orange juice too.)

[1]Dependency relations here do not have directions.

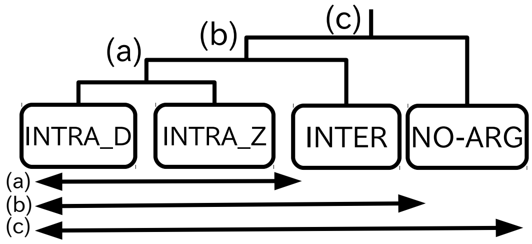

Figure 4.1: An argument identification in the "knockout" phase from the three most likely argument candidates.

4.2 Selection-and-Knockout Approach in Predicate Argument Structure Analysis

We propose a *Selection-and-Knockout approach*. This approach aims that explicitly-trained classification models between candidates in different positional relations are able to distinguish the marginal cases where two candidates are in different positional relations. In order to train and use such models, we split the process of predicate argument structure analysis into two phases; the *selection* phase and the *knockout* phase.

4.2.1 Selection Phase

In the first phase, we select the most likely argument for each of the INTRA_D, INTRA_Z and INTER types for each predicate using an argument identification model. We may use different features for models of different argument types. At the analysis, each model accepts a predicate and candidates as its input and outputs one of them.

4.2.2 Knockout Phase

In the second phase, we determine which of the three candidates (INTRA_D, INTRA_Z and INTER), is the most appropriate argument, or if there is no explicit argument appearing in the context (NO-ARG).

This phase is composed of three binary classification models (a), (b), and (c) illustrated in Figure 4.1.

(a) Judge which of INTRA_D and INTRA_Z is more likely to be an argument of the predicate.

(b) Judge which of INTER and the candidate selected by (a) is more likely to be an argument of the predicate.
(c) Judge whether the candidate selected by (b) qualifies as an argument of the predicate or not.

We tried different orders for the "knockout" phase in the preliminary experiment, the order shown in Figure 4.1. This is because two types of inter-sentential arguments INTRA_D and INTRA_Z have similar characteristics and training models between the two types. The classifier (a) in Figure 4.1 is trained to distinguish marginal cases between INTRA_D and INTRA_Z. Moreover, characteristics of clues to identify inter-sentential arguments and intra-sentential arguments are significantly different. For identification of intra-sentential arguments, syntactic structures and function words are very important clues. On the other hand, for identification of intra-sentential arguments, sentence distance and discourse play as strong clues. Therefore training models for INTRA_D and INTER or for INTRA_Z and INTER are not very important. This is why the our proposed knockout phase first selects more likely candidates from two intra-sentential candidates.

Training Method of the Knockout Phase

As shown in Algorithm 1, each of the classifiers (a), (b), and (c) uses the most likely candidates in the selection phase for its training examples. First, it obtains argument positional relation type of the gold argument of a given one predicate (line 2). Then, it collects the most likely candidates from INTRA_D, INTRA_Z, and INTER (line 5 to 7). This collection is the same procedure in the selection phase in the actual analysis.

If the gold argument positional relation type is NO_ARG, it makes three examples labeled "NO_ARG" for classifier (c) by using the three most likely candidates (line 10 to 12).

If the gold argument positional relation type is not NO_ARG, it first makes one example for classifier (c) by using the gold argument (line 14). Then, it makes two examples depending on the positional relation type of the gold argument (line 15 to 24).

If the gold argument is INTRA_D or INTRA_Z, it makes one example for classifier (a) with the gold argument and the most likely argument in the opposite positional relation type of intra-sentential argument type. In particular, it uses the most likely candidates of INTRA_Z if the gold argument is INTRA_D (line 16) and the one of INTRA_D if the gold argument is INTRA_Z (line 19). Of course, the label of the

Algorithm 1 Training data generation of (a) classifier_a, (b) classifier_b, (c) classifier_c

```
procedure TRAIN(predicate, gold_argument, candidates)
    gold_argument_type ← getArgumentType(predicate, gold_argument)

                        ▷ Collection of the most likely candidates in three positional relations
    most_likely_candidate_INTRA_D ← getMostLikelyCandidate(predicate, candidates, INTRA_D)
    most_likely_candidate_INTRA_Z ← getMostLikelyCandidate(predicate, candidates, INTRA_Z)
    most_likely_candidate_INTER ← getMostLikelyCandidate(predicate, candidates, INTER)

    if gold_argument_type = NO_ARG then
        MakeExample(classifier_c, NO_ARG, predicate, most_likely_candidate_INTRA_D)
        MakeExample(classifier_c, NO_ARG, predicate, most_likely_candidate_INTRA_Z)
        MakeExample(classifier_c, NO_ARG, predicate, most_likely_candidate_INTER)
    else
        MakeExample(classifier_c, HAVE_ARG, predicate, gold_argument)
        if gold_argument_type = INTRA_D then
            MakeExample(classifier_a, INTRA_D, predicate, gold_argument, most_likely_candidate_INTRA_Z)
            MakeExample(classifier_b, INTRA, predicate, gold_argument, most_likely_candidate_INTER)
        else if gold_argument_type = INTRA_Z then
            MakeExample(classifier_a, INTRA_Z, predicate, gold_argument, most_likely_candidate_INTRA_D)
            MakeExample(classifier_b, INTRA, predicate, gold_argument, most_likely_candidate_INTER)
        else if gold_argument_type = INTER then
            MakeExample(classifier_b, INTER, predicate, gold_argument, most_likely_candidate_INTRA_D)
            MakeExample(classifier_b, INTER, predicate, gold_argument, most_likely_candidate_INTRA_Z)
        end if
        return
    end if
end procedure


procedure MAKEEXAMPLE(classifier, label, predicate, candidate1, candidate2)    ▷ candidate2 is optional
    if candidate1 and candidate2 is not in co-reference relation then
        Collect features F for predicate and candidate1, candidate2
        Give a training example using F with label to classifier
    end if
end procedure
```

Table 4.1: Examples made for training. **bold texts** refer to the most likely arguments. Non-italic texts refer to the correct gold argument.

Gold positional relation type	Examples made for training.		
	classifier	label	used arguments
NO-ARG	(c)	NO-ARG	**INTRA_D**
	(c)	NO-ARG	**INTRA_Z**
	(c)	NO-ARG	**INTER**
INTRA_D	(a)	INTRA_D	INTRA_D, **INTRA_Z**
	(b)	INTRA_D	INTRA_D, **INTER**
	(c)	HAVE-ARG	INTRA_D
INTRA_Z	(a)	INTRA_Z	**INTRA_D**, INTRA_Z
	(b)	INTRA_Z	INTRA_Z, **INTER**
	(c)	HAVE-ARG	INTRA_Z
INTER	(b)	INTER	INTER, **INTRA_D**
	(b)	INTER	INTER, **INTRA_Z**
	(c)	HAVE-ARG	INTER

example is the gold positional relation type. It also makes one example for classifier (b) with the gold argument and the most likely argument in INTER (line 17 and 20). The label of the example is INTRA.

If the gold argument is INTER, it makes two examples for classifier (b) with the two most likely intra-sentential candidate (line 22 to 23). The label of the example is INTER.

We summarized the generated examples in Table 4.1. The procedure which makes an example with given candidates and a predicate (MAKEEXAMPLE in line 30) does nothing when given two candidates are annotated that are in the co-reference relation. It is the same training method as the tournament model (Iida, Inui, Takamura, and Matsumoto 2003) as described below.

4.3 Related Work

Our proposed model is based on two existing models; selection-and-classification model and tournament model, which we explain in this subsection.

4.3.1 Selection-and-Classification Model

The selection-and-classification model is proposed by Iida, Inui, and Matsumoto (2005) for noun phrase anaphora resolution. The model first selects the most likely

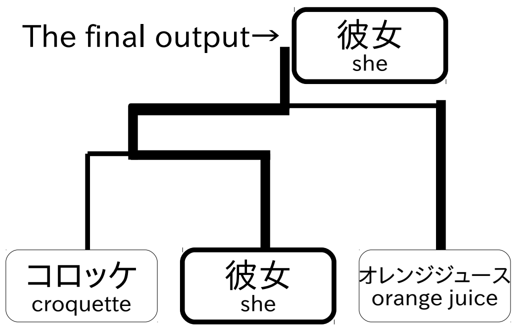

Figure 4.2: Identification of an argument with the tournament model

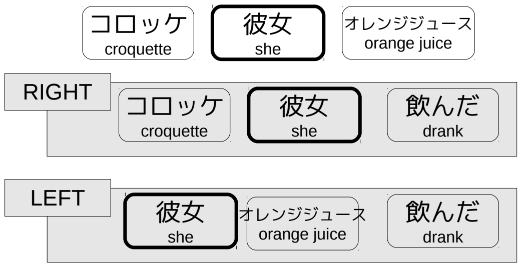

Figure 4.3: Training of the tournament model

antecedent for the target (possibly) anaphoric expression. Second, the model classifies the target anaphoric expression into either true anaphoric or not with the most likely antecedent. They employ this approach since there are almost no clues for a Japanese noun phrase to determine anaphoric or not by looking only at the noun phrase. Similarly, in our approach, after selecting the most likely candidate of the argument for each type, we determine which candidate is the actual argument.

For the negative example, our approach uses the three most likely candidates, one from each group, while the selection-and-classification model uses only one. We think that this is effective for judging whether an argument exists.

4.3.2 Tournament Model

The *tournament model* is proposed by Iida et al. (2003) for zero-anaphora resolution. For all the candidate antecedents (virtually all noun phrases appearing in preceding context), the model performs a two-class classification; which candidate in the pair of candidates is likely to be the antecedent of the zero-anaphora.

We give an example by taking Example 6 on page 24. Take nominative argument 彼女 (she) of the predicate 飲んだ (drank) and consider there are two other candidates コロッケ (croquette) and オレンジジュース (orange juice). For identification, the method lines up candidates in order of appearance in the document, and then performs “knockout tournament”[2] as shown in Figure 4.2.

The advantage of the tournament model is that the model can use pairwise features of candidates. Additionally, the ordering of candidates utilizes the characteristics that nearby arguments tend to be an argument, because backward candidates have an advantage to output.

For training of the model, it generates examples labeled “LEFT” or “RIGHT” with both the gold argument and another candidate. If the gold argument is forward, the label is “LEFT” and if it is the backward, the label is “RIGHT”, as shown in Figure 4.3. However, no example is generated if the gold argument and another candidate are in co-reference relation. This is because using such pairs adds noise for training.

Similarly, we select an argument comparing the most likely candidates of two argument types in the knockout phase of our approach. While the original tournament model is trained and compares candidates regardless of positional relations, our model explicitly compares the most likely candidates in all positional groups. We think that it is effective for explicit training and comparison for marginal situation.

---

[2]They called this “tournament”, but we use a more proper term “knockout.”

Chapter 5

Evaluation

In this chapter, we evaluate our proposed Selection-and-Knockout approach for predicate argument structure analysis. The purpose of this evaluation is investigation of the following matters:

- Implicit comparison of the most likely candidates in different positional groups selected by unique models.
- Discrimination of the inter-sentential candidate groups into INTRA_D and INTRA_Z
- Comparison of the most likely candidates of INTRA_D and INTRA_Z

We discuss the overall performance by comparing several models and our proposed model.

5.1 Evaluation Dataset

We use the NAIST Text Corpus 1.4β (Iida et al. 2007) for evaluation, because most previous work performs experiments on this version. We excluded 6 articles due to

Table 5.1: Statistics of NAIST Text Corpus

	Articles	Editorials	# of articles	# of sentences	# of predicates
Training	January 1st to 11th	January to August	1,673	23,150	64,173
Development	January 12th to 13th	September	458	4,634	13,105
Evaluation	January 14th to 17th	October to December	662	8,795	24,296

Table 5.2: Statistics of arguments in NAIST Text Corpus

	Case	SAME_BS	INTRA_D	INTRA_Z	INTER	EXO	NULL
Training	ga	128	33,775	12,057	7,436	10,529	248
	o	58	22,869	2,051	806	60	38,329
	ni	527	10,427	555	278	25	52,361
Development	ga	39	6,954	2,700	1,819	1,501	92
	o	10	4,724	445	147	18	7,761
	ni	51	2,673	212	80	5	10,084
Evaluation	ga	40	12,805	4,829	3,241	3,140	241
	o	18	8,665	829	291	14	14,479
	ni	132	5,023	358	155	12	18,616

annotation errors.[1]

We split the data in the way figured in Table 5.1. The way to split the data is the same as Taira et al. (2008) and Yoshikawa et al. (2011). We show the argument distribution statistics in Table 5.2.[2]

5.2 Evaluation Settings

All the features were automatically acquired with the result of the following systems.

- Part-of-speech tags: Japanese part-of-speech and morphological analyzer MeCab 0.996[3]
- Bunsetsu segments: Kyoto Text Corpus 3.0
- Dependency relation: Japanese dependency structure analyzer CaboCha 0.66[4]

Both analyzers use the IPADIC-2.7.0-20070801. The analysis is performed from the beginning of the sentence to the end. Candidates are extracted from the sentences appearing before the sentence that has the target predicate. One bunsetsu segment is regarded as one candidate.

Generally speaking, predicate argument structure analysis includes predicate identification. However, all previous work assumed predicate positions to be known be-

[1]Excluded ID of documents ID: 951230038, 951225057, 950106156, 950106034, 951221047, 950106211

[2]SAME_BS means that an argument exists in the same bunsetsu segment where a predicate does.

[3]https://code.google.com/p/mecab/

[4]http://sourceforge.jp/projects/naist-jdic/

forehand. Accordingly, we also follow their setting and give the target positions to our system in advance. Predicates include light verb like "する" (do) and compound verbs.

In order to identify the most likely argument candidates, we used the tournament model (Iida et al. 2003). We trained different models and used them for search of the most likely candidates. For instance, our proposed method identifies the three most likely candidates, INTRA_D, INTRA_Z, and INTER, in the selection phase by using three different models.

5.3 Classifier and Features

We used Support Vector Machine (Cortes and Vapnik 1995) for each classification model with a linear kernel. We used the implementation of LIBLINEAR 1.93[5] and tuned its parameters using a development data.

We employed almost the same features adopted in (Imamura et al. 2009).

- Head word, functional word, and other words and their parts-of-speech for the bunsetsu segments in the predicate and the candidate
- When the predicate includes passive auxiliaries, its base form
- Relation between the predicate and the candidate in the dependency tree[6]
  Notation:
  - N_a: The candidate bunsetsu segment in the dependency tree
  - N_p: The predicate bunsetsu segment in the dependency tree
  - N_c: The bunsetsu segment where two paths N_a to ROOT and N_p to ROOT cross
  - $A_{a \cdots c}$: Bunsetsu segments on the path between N_a and N_c
  - $A_{p \cdots c}$: Bunsetsu segments on the path between N_p and N_c
  - $A_{c_1, c_2, \cdots c_r}$: Bunsetsu segments on the path between N_c and ROOT

  We converted each bunsetsu segment N_x to a strings in the following five styles and their concatenations.

[5] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

[6] (Imamura et al. 2009) does not describe implementation.

- Base form of the head word
- Part-of-speech tag of the head word
- Base form of the functional word
- Part-of-speech tag of the functional word
- Base form and part-of-speech of the functional word

We assume $S_{a\cdots c}$ is a string representation of $A_{a\cdots c}$, $A_{p\cdots c}$ is a string representation of $S_{p\cdots c}$, and their concatenation is $S_{a\cdots c} + S_{p\cdots c}$.

We utilized $r+1$ types of strings for the one style. $S_{a\cdots c} + S_{p\cdots c}$, $S_{a\cdots c} + S_{p\cdots c} + S_{c_1}$, $S_{a\cdots c} + S_{p\cdots c} + S_{c_1,c_2}$, $\cdots$ $S_{a\cdots c} + S_{p\cdots c} + S_{c_1,c_2,\cdots,c_r}$ Therefore, we expressed the relation between the predicate and the candidate in $5(r+1)$ strings.

- Relations between two candidates in the dependency tree
- Distances measured by the number of bunsetsu segments or sentences between the predicate and the candidate
- Distances measured by the number of bunsetsu segments or sentences between the candidates
- PMI (pointwise mutual information) (Hindle 1990) scores of [⟨Head word of the candidate, Case⟩,⟨Predicate⟩] calculated from corpora.[7]

  We followed the model of co-occurrence of a predicate and an argument by Fujita, Inui, and Matsumoto (2004). We regarded $\langle v,c,n\rangle$ as co-occurrence of $\langle v,c\rangle$ and n in order to estimate the co-occurrence probability $P(\langle v,c,n\rangle)$ where a noun n depends a verb v via a case-particle c.

$$PMI(\langle v,c\rangle,n) = \log \frac{P(\langle v,c,n\rangle)}{P(\langle v,c\rangle)P(n)}$$

  We did not perform any smoothing. We computed PMI scores and converted them into binary features[8] by using the following two corpora: NEWS and WEB.

[7](Imamura et al. 2009) used co-occurrence score with Good Turing discounting and back off smoothing. It is computed by the Japanese Mainichi newswire corpus between 1991 to 2002 excluding corresponding articles in NAIST text corpus.

[8]Feature fires only when the value is less than x. In the experiment, x changes from -4 to 4 by 0.1.

NEWS: We used about 18 million sentences in Mainichi newspapers published from 1991 to 2003 (excluded 1995) (The Mainichi Newspapers 1991 2003). Part-of-speech tagging was performed on the data with MeCab 0.98 and dependency structure parsing was done by CaboCha 0.60pre4. Both analyzers use the NAIST Japanese Dictionary 0.6.3. We extracted about 27 million pairs of a predicate and an argument with a case marker particle "が" (ga), "を" (o) or "に" (ni).[9]

WEB: We used about 500 million sentences collected from the Web by Kawahara and Kurohashi (2006). Part-of-speech tagging was performed by JUMAN and dependency parsing was done by KNP. We extracted 5.3 billion pairs of a predicate and an argument using KNP.[10]

- Binary information whether the candidate phrase is used as an argument in any of previous predicates
- A rank of the candidate in Salient Reference List (Nariyama 2002)

5.4 Models for Comparison

Because previous works use different features and machine learning methods in different settings from ours, we compare our model SK (Selection and Knockout) with a baseline model IIDA2005, and other models IIDA2007, IIDA2007^{+}, SK^{-} in order to analyze how it is effective to divide a model considering positional relations.

5.4.1 IIDA2005

This model selects a most likely candidate among all candidates regardless of positional relations in the selection phase. Then, this model judges whether it is feasible for an argument or not (eligibility judgment). It is identical to the selection-and-classification model (Iida et al. 2005).

Furthermore, it is almost the same to (Imamura et al. 2009) in terms of the selection of one among all candidates. Main differences between them are two-fold: (1) IIDA2005 uses different models for identification of the most likely candidate and

[9] Unique total are; Verb: about 31 hundred, Noun: 327 hundred, Pair: 7 million

[10] Unique total are; Verb:about 801 million, Noun: about 288 million, Pair: 160 million

eligibility judgment. (2) IIDA2005 additionally uses relational features between candidates in the selection phase.

The purposes of this model are: (1) To investigate whether the priority order for positional groups is effective or not. (2) To investigate the effect of implicit comparison of the most likely candidates in different positional groups selected by unique models.

5.4.2 IIDA2007

This model first selects the most likely candidate after search of inter-sentential candidates. If it is judged feasible for the argument, it finishes the search. Otherwise, it does the same procedure for the inter-sentential candidates. It is a model of (Iida et al. 2007), which preferentially searches for inter-sentential candidates as described in Section 3.1.2.

Our implementation does not use BACT but SVM for identification of the most likely candidates and deciding wheter the candidate is feasible to be an argument. By comparing this and IIDA2005, we investigate the effect of inter-sentential candidates first approach.

5.4.3 IIDA2007$^+$

This model first selects the most likely candidate after search of INTER_D candidates. If it is judged feasible for the argument, it finishes the search. Otherwise, it does the same procedure for INTRA_Z and INTER.

This model is an extension of IIDA2007 which searches syntactically nearby groups.

By comparing it and IIDA2007, we investigate the effect of division of the inter-sentential candidate group into INTRA_D and INTRA_Z.

5.4.4 SK$^-$

This model has two steps in the "knockout" phase to assume the argument type is classified to two groups INTRA and INTER. INTRA includes INTRA_D and INTRA_Z.

The knockout phase is composed of two binary classification models (b) and (c) illustrated in Figure 4.1. Classifier (c) selects whether the most likely argument is in INTRA or INTER unlike our proposed model.

By comparing it and SK, we investigate the effect of comparison of the most likely candidates of INTRA_D and INTRA_Z.

5.4.5 Comparison to Previous Work

We compare our proposal with previous work by comparing SK, (Taira et al. 2008) and (Imamura et al. 2009). They experiment for all positional relations with NAIST text corpus.

Strictly speaking, these systems are not directly comparable since they do not have the same experimental setting. For example, the experiment of (Taira et al. 2008) is performed with 19,501 predicates for test, 49,527 for training, and 11,023 for development. It uses gold syntactic dependency and gold POS annotation in Kyoto Text Corpus 4.0 for training and in-house POS-Tagger for test.

The experiment of (Imamura et al. 2009) is performed with 25,500 predicates for test, 67,145 for training, and 13,594 for development. They use gold syntactic dependency and POS in Kyoto Text Corpus 4.0, whereas we take those annotations from Kyoto Text Corpus 3.0.

There are other previous works which also propose Japanese predicate argument structure analysis. However, we do not make comparisons because of the following reasons.

(Sasano and Kurohashi 2011) exclud predicates in passive form or causative form for the evaluation, because their system analyze surface cases.

(Yoshikawa et al. 2011) do not analyze inter-sentential arguments.

(Watanabe, Asahara, and Matsumoto 2010) propose a structured prediction model that learns predicate word senses and argument roles simultaneously. However, they used different dataset for their evaluation.

5.5 Evaluation Metrics

We evaluate precision, recall, and F-value with the following formula for INTRA_D, INTRA_Z, and INTER.

$tp_{(T)}$ is the number of arguments which the system successfully identifies and whose positional relation is T. $tp_{(T)}$ is the number of arguments which the system incorrectly identifies and whose positional relation is T.

Table 5.3: Comparison of predicate argument structure analysis of nominative case

	INTRA_D			INTRA_Z			INTER			ALL			
	P	R	F	P	R	F	P	R	F	P	R	F	A_M
IIDA2005	80.32	85.49	82.82	45.86	48.33	47.07	27.05	17.37	21.16	66.56	88.54	75.99	50.35
IIDA2007	82.93	87.40	85.11	51.12	50.11	50.61	40.49	2.56	4.82	74.29	79.59	76.85	46.85
IIDA2007+	85.22	85.85	85.53	59.63	33.53	42.92	31.74	2.25	4.21	80.06	68.72	73.95	44.22
SK−	82.11	87.88	84.90	50.22	51.31	50.76	26.65	14.22	18.55	69.68	87.73	77.67*	51.40
SK	84.25	86.97	85.59	51.81	50.67	51.24	25.78	15.86	19.64	70.72	85.95	77.59*	52.15
Taira et al. (2008)	-	-	75.53	-	-	30.15	-	-	23.45	-	-	57.4	43.04
Imamura et al. (2009)	85.2	88.8	87.0	58.8	43.4	50.0	47.5	7.6	13.1	79.4	68.0	73.2	50.03

Table 5.4: Comparison of predicate argument structure analysis of accusative case

	INTRA_D			INTRA_Z			INTER			ALL			
	P	R	F	P	R	F	P	R	F	P	R	F	A_M
IIDA2005	92.30	92.42	92.36	42.41	32.33	36.69	13.13	8.93	10.63	87.33	89.29	88.30	46.56
IIDA2007	92.48	92.57	92.53	42.51	31.48	36.17	22.22	0.69	1.33	89.11	88.18	88.65	43.34
IIDA2007+	92.87	92.21	92.54	50.29	10.62	17.53	22.22	0.69	1.33	91.95	84.46	88.05	37.13
SK−	92.25	92.72	92.48	41.60	31.97	36.15	14.94	4.47	6.88	88.12	89.21	88.66	45.17
SK	92.94	92.72	92.83	46.29	29.31	35.89	15.96	5.15	7.79	89.52	88.40	88.96*	45.51
Taira et al. (2008)	-	-	88.20	-	-	11.41	-	-	9.32	-	-	79.5	36.31
Imamura et al. (2009)	95.60	92.20	93.90	53.70	21.60	30.80	25.00	0.40	0.70	94.3	84.7	89.2	41.80

$$Precision = \frac{tp_{(T)}}{tp_{(T)} + fp_{(T)}}, \quad Recall = \frac{tp_{(T)}}{tp_{(T)} + fn_{(T)}}, \quad F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

We also define tp, fp, and fn of the whole the system (ALL), and Precision, Recall, F-value of the system.

5.6 Results

Tables 5.3–5.5 present the results of the experiments of nominative, accusative, and dative cases respectively. P, R, F, A_M means Precision, Recall, F-value, and macro-average of F-value (arithmetic average of F-values of INTRA_D, INTRA_Z, and INTER).

We performed Approximate Randomization Test (Chinchor, Hirschman, and Lewis 1993) for F-value of ALL of SK^-, SK and IIDA2007 using the script composed by Takamura[11]. [12] The mark * means $p < 0.05$ compared to IIDA2007.

[11] http://www.lr.pi.titech.ac.jp/~takamura/pubs/randtest_fm.pl

[12] We regarded the occasions when the system outputs incorrect arguments as not fp but fn.

Table 5.5: Comparison of predicate argument structure analysis of dative case

	INTRA_D			INTRA_Z			INTER			ALL			
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	A_M
IIDA2005	90.18	71.49	79.76	40.85	8.10	13.52	8.57	1.94	3.16	88.63	64.75	74.83	32.14
IIDA2007	90.27	71.27	79.65	41.33	8.66	14.32	0.00	0.00	nan	89.25	64.47	74.86	nan
IIDA2007+	89.95	71.45	79.64	72.00	5.03	9.40	0.00	0.00	nan	89.73	64.22	74.86	nan
SK−	90.18	71.49	79.76	39.39	7.26	12.26	10.00	0.65	1.21	89.16	64.71	74.99	31.08
SK	90.15	71.57	79.79	51.11	6.42	11.41	7.14	0.65	1.18	89.42	64.59	75.01	30.80
Taira et al. (2008)	-	-	89.51	-	-	3.66	-	-	11.76	-	-	83.15	34.98
Imamura et al. (2009)	91.10	72.60	80.80	0.00	0.00	nan	0.00	0.00	nan	91.1	66.1	76.6	nan

5.6.1 Comparison to Deterministic Models

We discuss the overall performance by comparing ALL's F-value of IIDA2005, IIDA2007, and IIDA2007$^+$.

Nominative Cases

In the nominative case, the performance results are IIDA2007 $>$ IIDA2005 $>$ IIDA2007$^+$.

Precision of IIDA2007 is higher than that of IIDA2005. On the other hand, recall of IIDA2007 is lower than that of IIDA2005. Precision can be enhanced by limiting the search candidates to inter-sentential candidates. Though recall decreases due to its limitation, the overall performance improves, because inter-sentential arguments are about three times as many as intra-sentential arguments.

IIDA2007$^+$ improves precision and F-value of INTRA_D owing to prior search of INTRA_D. It also enhances precision of INTRA_Z, but it deteriorates the recall of INTRA_Z considerably. As a result, such prior search makes the overall performance worse.

Accusative Cases

It has similar tendency to the nominative case. Prior search of INTRA_D does not make the overall performance much worse, not like the nominative case, because the number of INTRA_D is not so large.

Dative Cases

Unlike nominative and accusative cases, in the dative case the performance results are IIDA2007$^+$ $\simeq$ IIDA2007 $>$ IIDA2005.

Table 5.6: Confusion Matrix of nominative errors in IIDA2007 (left in each cell), SK$^-$ (center in each cell), and SK (right in each cell)

Gold \ System	INTRA_D	INTRA_Z	INTER	NO-ARG
INTRA_D	468 / 478 / 434	464 / 457 / 485	16 / 194 / 231	665 / 423 / 518
INTRA_Z	686 / 717 / 642	733 / 748 / 699	23 / 266 / 319	967 / 620 / 722
INTER	632 / 679 / 527	615 / 677 / 579	43 / 489 / 564	1,868 / 935 / 1057
NO-ARG	506 / 563 / 465	486 / 558 / 499	40 / 317 / 361	0 / 0 / 0

One possible cause of this inconsistency is the difference in argument distribution. According to Table 5.2, over 90% of arguments in the dative case are INTRA_D. Thus, prior search of INTRA_D improves recall of INTRA_D, and therefore increases overall performance.

5.6.2 Proposed Methods

Deterministic models lower recall and F-value of arguments in low priority positional groups. This makes decrease in macro average of F-value. On the other hand, proposed method enhances micro average (F-value of ALL) without sacrificing macro average due to explicit comparison of the most likely candidates in all positional groups. In fact, both SK and SK$^-$ are superior to IIDA2005, IIDA2007, and IIDA2007$^+$.

Table 5.6 shows breakdown of errors in the nominative case by IIDA2007, SK$^-$, and SK. Though errors in INTER increase with SK$^-$ and SK (according to the third column), false negatives decrease (fourth column). It may be because IIDA2007 classifies argument eligibility of inter-sentential candidates without referring to inter-sentential candidates, while SK$^-$ and SK can refer to the inter-sentential most likely candidates, when classifying argument eligibility.

SK is comparable with SK$^-$ in the nominative and dative cases. In the accusative case, improvement of precision of INTRA_D enhances the overall performance. Thus, we can say that an implicit comparison of two inter-sentential candidates INTRA_D and INTRA_Z is effective.

5.6.3 Comparison to Previous Work

As for the nominative case, the overall performance of SK is higher than those of (Taira et al. 2008) and (Imamura et al. 2009). This is because (Imamura et al. 2009) does not exploit information between candidates, and (Taira et al. 2008) adopts deterministic analysis. In the accusative case, the performance of SK is also higher than that of (Taira et al. 2008), and comparable with (Imamura et al. 2009).

On the other hand, the performance of (Taira et al. 2008) is the best in the dative case. (Imamura et al. 2009) does not work better than (Taira et al. 2008) in the dative case in contrasted in the nominative and accusative cases. This is because it depends on arguments in other cases. Proposed method and (Imamura et al. 2009) perform argument structure analysis without referring to the analysis results in other cases, while (Taira et al. 2008) jointly analyzes argument structures by using the relationship like "the bunsetsu phrase which includes an argument depends indirectly on one which includes a predicate via another case argument" (ga_c, wo_c, ni_c).

Chapter 6

Error Analysis

In this chapter, we discuss what kind of errors have still remained and what approach is hopeful. We analyze them from not only a quantitative perspective but also a qualitative one.

6.1 Predicate Type Classification

We classified all predicates in NAIST Text Corpus into six groups.

- Verb predicate (VERB)
- Nominal verb predicate (N_VERB)
- Noun predicate (NOUN)
- "I"-adjective predicate (ADJECTIVE_I)
- "Na"-adjective predicate (nominal adjective predicate) (ADJECTIVE_NA)
- Adnominal phrase (ADNOMINAL)

For the classification, we used the automatic POS tagger MeCab with IPADIC-2.7.0-20070801 and handcrafted rules shown in Algorithm 2. In this algorithm, OTHER means it was hard to automatic predicate type classification because of POS tagging error. We show the statistics of argument types in Table 6.1.

Algorithm 2 Predicate type classification

```
procedure PREDICATETYPECLASSIFICATION(words)
    last_word ← words[-1]

    if last_word.pos1 = “形容動詞語幹” or last_word.pos2 = “形容動詞語幹” then
        return ADJECTIVE_NA

    else if last_word.pos0 = “形容詞” then
        return ADJECTIVE_I

    else if last_word.pos0 = “連体詞” then
        return ADNOMINAL

    else if last_word.pos0 = “動詞” then
        if last_word.base_form = “する” then
            return N_VERB
        else
            return VERB
        end if

    else if last_word.pos0 = “名詞” then
        if last_word.pos1 = “接尾” or length(words) ≥ 2 then
            last_word ← words[-2]
        end if
        following_word ← get_following_word(last_word)
        if last_word.pos1 = “サ変接続” or last_word.pos2 = “サ変接続” or following_word.base_form = “する” then
            return N_VERB
        end if
        return NOUN

    else if last_word.pos0 = “助動詞” or last_word_pos1 = “接続詞” then
        return VERB

    else if last_word.pos0 = “副詞” then
        return NOUN

    end if
    return OTHER
end procedure
```

Table 6.1: Statistics of arguments in Test Data in NAIST Text Corpus by Predicate Type

Case	Predicate-type	SAME_BS	INTRA_D	INTRA_Z	INTER	EXO	NULL	Total
Nominative	VERB	12	6,576	2,888	1,974	2,142	152	13,744
	N_VERB	6	2,717	1,247	919	890	32	5,811
	NOUN	13	1,072	373	253	58	28	1,797
	ADJECTIVE_I	7	1,171	123	44	34	19	1,398
	ADJECTIVE_NA	2	1,196	163	41	5	2	1,409
	ADNOMINAL	0	43	2	1	2	0	48
	OTHER	0	30	33	9	9	8	89
Accusative	VERB	8	6,119	501	174	12	6,930	13,744
	N_VERB	5	2,531	307	117	2	2,849	5,811
	NOUN	0	5	11	0	0	1,781	1,797
	ADJECTIVE_I	4	6	1	0	0	1,387	1,398
	ADJECTIVE_NA	0	1	0	0	0	1,408	1,409
	ADNOMINAL	0	0	3	0	0	45	48
	OTHER	1	3	6	0	0	79	89
Dative	VERB	98	3,818	218	99	12	9,499	13,744
	N_VERB	0	1,033	104	52	0	4,622	5,811
	NOUN	1	11	8	0	0	1,777	1,797
	ADJECTIVE_I	1	133	22	3	0	1,239	1,398
	ADJECTIVE_NA	0	26	5	1	0	1,377	1,409
	ADNOMINAL	0	0	0	0	0	48	48
	OTHER	32	2	1	0	0	54	89

6.1.1 Adjectives

Note that there are two kinds of Japanese adjectives; "I"-adjective and "Na"-adjective. "Na"-adjectives are also referred to as adjectival nouns ("形容動詞" in Japanese) or nominal adjectives ("名詞的形容詞" in Japanese). This is because their behaviors are also close to nouns like following particles in Example 7.

Example 7:

uplifted	ga-particle	feel
元気	が	出る

(I feel uplifted.)

IPADIC with which NAIST Text Corpus was annotated treats such "Na"-adjectives nouns. Thus, we regard predicates which end with such nouns as "Na"-adjective predicates.

Additionally, in NAIST Text Corpus adnominal words like "大きい" (big) are also tagged as predicates, though they are not "I"-adjective nor "Na"-adjective.

Table 6.2: Comparison of predicate argument structure analysis of VERB

	INTRA_D			INTRA_Z			INTER			ALL		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Nominative	83.12	85.31	84.20	50.95	49.45	50.18	25.83	15.81	19.61	68.31	82.32	74.66
Accusative	93.88	93.33	93.61	46.00	27.54	34.46	7.55	2.30	3.52	90.94	89.40	90.16
Dative	91.51	74.25	81.98	47.37	4.13	7.59	14.29	1.01	1.89	91.07	67.84	77.75

6.1.2 Nominal Verbs

Nominal verbs ("サ変動詞" in Japanese) behave differently from ordinary verbs.

1. They usually follow "する" (do) to behave as predicates.
   理解する (understand) 損する (lose)
2. They sometimes do not follow "する"
   (a) If employers and employees agree / even now / family-care leave system / o-particle / start / can
   労使間が合意すれば いまからでも 介護休業制度 を 実施 できる。
   (If employers and employees agree, even now the family-care leave system can be started.)
   (b) to the citizens who supported the disease overcome / thanks / , and / in his home country / International Foundation leukemia / founded
   病気克服を支えた市民に 感謝 、 母国に 国際白血病財団を 設立した。
   (To thank the citizens who supported the disease overcome, he founded the International Foundation leukemia in his home country.)
3. They can be used as nouns
   (a) number of births / rapid increase / of / background
   出生数 急増 の 背景
   (The background of rapid increase of the number of births)

Thus, we distinguish them from other verbs for analysis by making a new predicate type (nominal verb predicate).

6.2 Verb Predicate

We show the performance of our system SK for VERB in Table 6.2. The result of analysis in the accusative case is relatively better than those of the other cases. (The

F-value in the accusative case exceeds 90.) We consider that analysis results in the accusative case should be useful clues for analysis in the other cases. Thus, in this section, we investigate errors where accusative arguments are successfully identified though other arguments are not.

6.2.1 Light Verb

Many errors are made by verbs which behave differently from other verbs such as “行う” (do), “する” (do), “なる” (become), “ある” (be). Such verbs themselves do not play the central role in comprising the meaning of a sentence. Their arguments carry the main meaning of the argument structures. They are known as light verbs and this phenomenon is called “light verb construction” (Miyamoto 1999).

“行う” (do) in Example 8 gives central meaning to the accusative argument “技術移転” (technical transfer).

Example 8:

Japan	no-particle	desulphurization technology	o-particle	like
日本$_{\text{nominative}}$	の	脱 硫 技 術	を	はじめとした

energy utilization technology	wa-particle	the top of the world		to China and so on
エネルギー利用技術	は	世界の先端をいっており	、	中国などへの

technical transfer	o-particle	actively	do	should
技 術 移 転$_{\text{accusative}}$	を	積極的に	行う	べきだ。

(Because Japanese energy utilization technologies like desulphurization technology are the top of the world, they should be engaged in technology transfers to China and so on.)

“し” (do; the base form is “する”) in Example 9 also gives central meaning to an accusative argument “損” (disbenefit).

Example 9:

disbenefit	o-particle	do	did	people
損$_{\text{accusative}}$	を	し	た	層$_{\text{nominative}}$

(people who lose benefit)

For such verbs, we should use information of the accusative argument for analysis. Simply, we may regard the argument as the predicate instead of the light verb.

On the other hand, “なる” (be, become) in the following sentences gives the central meaning to the accusative arguments, and the accusative arguments behave as if they are noun predicates. In other words, nominative arguments and dative arguments behave as the same.

Example 10:

admit withdrawal / whether / ga-particle / first / step / to-particle / become

離脱を認めるか どうか$_{\text{nominative}}$ が 、最初の 関門$_{\text{dative}}$ と なる 。

(It is the first step whether they admit withdrawal.)

Example 11:

length / 40 meters / ni-particle / mo-particle / be / with 3 cars / semitrailer

長さ$_{\text{nominative}}$ 40 メートル$_{\text{dative}}$ に も なる 3 両編成の 大型トラック

(semitrailer with 3 cars whose length is 40 meters)

For the analysis, we should exploit the model for noun predicate because of the analogy.

6.2.2 Dative Obligatory Judgment

In NAIST Text Corpus, obligatory grammatical cases (nominative, accusative and dative cases) are annotated. The annotation guideline requires annotators not to tag optional arguments as dative arguments, even if they follow "に" (ni; dative case marker).

However, as Iida et al. (2007) noted, obligatory judgment of phrases following ni-particle can be very subjective compared with the other cases. In fact, the annotator agreement for dative INTRA_D (79.85) is less than those of nominative (82.65) and accusative (92.57) (Matsubayashi, Iida, Sasano, Yokono, Matsuyoshi, Fujita, Miyao, and Inui 2013). This lowers recall and F-value of the system for the dative case. In the following sentence, our system SK does not output dative argument, though the predicates have dative arguments.

Example 12:

bridge beam / no-particle / joint parts / ni-particle / about 50 centimeters

橋げた の ジョイント部分$_{\text{dative}}$ に 約五〇センチの

uneven / ga-particle / was generated / and so on

段差$_{\text{nominative}}$ が 出来る などし 、

(about 50 centimeters uneven was generated at joint parts of bridge beam)

Example 13:

January ni-particle enter after the second cold day

一月$_{dative}$ に 入って から 二番目の冷え込み 。

(It is the second cold day in January)

For practical purposes, whether the argument is obligatory or not may not be very important. Thus, we may perform semantic role labeling instead of identification of the dative case.

6.2.3 Case Alternation

NAIST Text Corpus is annotated with the grammatical case relations between the base form of the predicate and its arguments. Thus, assigned cases for arguments will change depending on whether the predicates follow auxiliary verbs which cause syntactic transformations such as passivization and causativization. In fact, in some errors our system SK identified arguments in another incorrect case. For example, "子供" (child) in Example 14 is analyzed as dative argument incorrectly by our system SK. (An auxiliary verb "せ" makes causativization)

Example 14:

about three years old child ni-particle numbers o-particle write make

三歳くらいの 子供$_{nominative}$ に 数字$_{accusative}$ を 書か せ 、

parents who scolded him/her for not writing them o-particle saw

出来ないとしかっている親 を 見かけました 。

(I saw parents who make his/her about three years old child write numbers and scolded him/her for not writing them.)

Previous work tried to exploit this linguistic phenomenon by adding features which fire in such conditions. However, it is a difficult question how many training examples are needed to weight such features properly (data sparseness). We should adopt more explicit approach like rule-based transformation using a dictionary for these case alternations not by implicit feature expression.

Our system SK identified "行方不明者" (the disappeared) as the nominative argument in Example 15. "含む" (included) is an intransitive verb and its transitive coun-

Table 6.3: Comparison of predicate argument structure analysis of N_VERB

	INTRA_D			INTRA_Z			INTER			ALL		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Nominative	82.90	85.09	83.98	57.24	55.49	56.35	31.15	17.63	22.52	70.08	82.47	75.77
Accusative	91.08	91.58	91.33	46.36	33.22	38.71	26.83	9.40	13.92	86.64	87.07	86.85
Dative	86.47	66.21	75.00	45.00	8.65	14.52	0.00	0.00	nan	84.72	59.48	69.89

Table 6.4: Comparison of argument distribution of VERB and N_VERB on percentage

		INTRA_D	INTRA_Z	INTER
Nominative	VERB	47.85	21.01	14.36
	N_VERB	46.76	21.46	15.81
Accusative	VERB	44.52	3.65	1.27
	N_VERB	43.56	5.28	2.01
Dative	VERB	27.78	1.59	0.72
	N_VERB	17.78	1.79	0.89

terpart is “含める” (include). We can exploit such a correspondence relation.

Example 15:

burying alive	o-particle	included	the disappeared	over 580
生き埋め$_{\text{accusative}}$	を	含む	行方不明者$_{\text{dative}}$	五八〇人以上

(The disappeared people including burying alive are over 580.)

JUMAN dictionary contains information about relations between transitive verbs and intransitive verbs. However, the coverage is not broad, because the number of words in the dictionary is about 30,000. Recently, Sasano, Kawahara, Kurohashi, and Okumura (2013) have acquired large knowledge of relation between the passive and active voices by using Web corpus. They plan to apply their framework to acquire knowledge about case alternation between intransitive and transitive verbs.

6.3 Nominal Verb Predicate

We show the performance of our system SK for N_VERB in Table 6.3. Its results are similar to that of verb predicate; the performance in the accusative case is the best, and that in the dative case is the worst.

However, the performance in the accusative and dative cases is much worse than those in VERB (see Table 6.2). We consider this is caused by the fact that the number of INTRA_D of N_VERB is much less than that of VERB to be seen in Table 6.4. The performance of INTRA_D is worse especially in the dative case. On the other hand, the performance of INTRA_Z and INTER of N_VERB is better than those of VERB. This means systems need to search for more candidates to identify arguments. Moreover, most errors require context.

In Example 16, our system SK incorrectly identified “首相” (prime minister) as the nominative argument of “代理署名する” (sign by procuration) .

Example 16:

for this / prime minister / wa-particle / in the conversation / governor / for
これに対し 首 相 は 会 談 の 場 で 、知 事 nominative に対して

sign by procuration / not at all / did not ask / said that
代理署名する よう 一 切 、求めなかった という 。

(A person said that the prime minister did not ask the governor to sign by procuration in the conversation.)

In Example 17, our system SK incorrectly identified “社” (company) as the nominative argument of “追及し” (question).

Example 17:

in this action / complainant / wa-particle / captain and other persons’
今回の訴えでも、原 告 側 nominative は 機 長 ら の

negligence liability / and / design of the air frame has fault / point outed that / China Airlines
過 失 責 任 とともに、機体の設計に欠陥があった と 指 摘 し 、中 華 航 空

and / airbus company / for / joint tort liability / o-particle / question
と エアバス社 に対して 共同不法行為責任 accusative を 追及し

be to
ていく姿勢だ。

(In this action, complainant is to point outed that captain and other persons’ negligence liability and fault of air frame design, and question joint tort liability for China Airlines and airbus company.)

To address this problem, we need to exploit the relationship between the other argument structures. We may exploit analysis results in the accusative case because the performance is higher than that of the other cases.

Another approach is to use nominal verbs annotated as event nouns in NAIST Text Corpus. In NAIST Text Corpus, nominal verbs are annotated as not only predicates but also event nouns. (Komachi, Iida, Inui, and Matsumoto 2007) assumed argument structures of event nouns and verbs are the same in order to acquire co-occurrences of arguments and event nouns.

Table 6.5: Comparison of predicate argument structure analysis of NOUN

	INTRA_D			INTRA_Z			INTER			ALL		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Nominative	82.36	87.13	84.68	50.61	56.03	53.18	15.28	13.04	14.07	66.70	98.16	79.43
Accusative	0.00	0.00	nan	33.33	9.09	14.29	nan	nan	nan	9.09	6.25	7.41
Dative	83.33	45.45	58.82	100.00	12.50	22.22	nan	nan	nan	85.71	30.00	44.44

6.4 Noun Predicate

We show the results of NOUN in Table 6.5. According to Table 6.1, most noun predicates do not have accusative and dative arguments. Actually, "noun predicates" which have accusative or dative are incorrectly classified into NOUN, though they are other predicate types.

For example, "絡み" (relation) in Example 18 is POS-tagged as general noun (名詞-一般).

Example 18:

penalty / wa-particle / right to silence / and so on / to / relation / de-particle

罰則$_{\text{dative}}$ は 、黙 秘 権$_{\text{nominative}}$ な ど との 絡み で

has legal problem

法律上問題がある

(The penalty has legal problems related to right to silence and so on.)

"人気" (popular) in Example 19 is also POS-tagged as general noun, though it is used as "NA"-adjective.

Example 19:

speaking badly of their wives / scene / ga-particle / every time / this / ga-particle

女房の悪口を言い合う シーン$_{\text{nominative}}$ が 毎回あって、これ が

male viewers / no-particle / popular / is

男性視聴者$_{\text{dative}}$ の 人気 である 。

(Every time there is a scene where they speak badly their wives. This is popular for male viewers.)

Thus, we do not discuss the accusative and dative cases of noun predicates here. In the rest of this section, we investigate errors where nominative arguments are not identified by the system.

The nominative argument of "ビル" (building) in Example 20 is "本館" (main building). However, our system SK identified “とき” (when) as the argument.

Example 20:

in 60 main building ga-particle when it built great building was
六〇年に 本 館$_{nominative}$ が できたときは 威風堂々たる ビル だった

(When the main building was built in 60, it was great building.)

To address this predicate argument structure analysis, semantic information about nouns is needed. For example, Nihongo Goi Taikei (Ikehara, Miyazaki, Shirai, Yokoo, Nakaiwa, Ogura, Ooyama, and Hayashi 1997) is a Japanese thesaurus consisting of a hierarchy of about 3,000 semantic classes and about 300,000 nouns. Kyoto University noun case frame dictionary (Sasano and Kurohashi 2009) is an automatically constructed dictionary which describes the semantic relationship between two nouns. However, more difficult but important thing is understanding the contexts of documents. “救助” (rescue) is the nominative case of “宝探しゲーム” (a game of treasure digging) in Example 21 and “ゼロ化” (make it zeroi) is the nominative case of “夢” (dream) in Example 22.

Example 21:

relying on dog noses rapid rescue ga-particle necessary
犬の鼻を頼りにした 迅速な 救助$_{nominative}$ が 不可欠なわけだ 。

. . .

However for dogs a kind of a game of treasure digging de-particle
で も 犬にとっては 一種の 宝 探 し ゲ ー ム で 、

work hard for reward of oversized sausages
ごほうびの特大ソーセージ目指して必死になる

(It is necessary to rescue people relying on dog noses. . . . However, because it is a kind of game of treasure digging for dogs, they work hard for reward of oversized sausages.)

Example 22:

fire wa-particle thorough transformation of consciousness and attitudes
火事 は 意 識 や 態 度 の 変 革 と 徹 底 で

make it zero mo-particle dream is not
ゼ ロ 化$_{nominative}$ も 夢$_{nominative}$ ではない

(Making fire zero is not a dream thorough transformation of consciousness and

Table 6.6: Comparison of predicate argument structure analysis of ADJECTIVE_I

	INTRA_D			INTRA_Z			INTER			ALL		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Nominative	89.00	93.25	91.08	38.10	39.02	38.55	20.00	6.82	10.17	83.55	98.53	90.43
Accusative	57.14	66.67	61.54	nan	0.00	nan	nan	nan	nan	57.14	36.36	44.44
Dative	82.43	45.86	58.94	75.00	13.64	23.08	nan	0.00	nan	82.05	40.76	54.47

Table 6.7: Comparison of predicate argument structure analysis of ADJECTIVE_NA

	INTRA_D			INTRA_Z			INTER			ALL		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
Nominative	90.89	94.31	92.57	39.23	31.29	34.81	15.38	9.76	11.94	84.68	99.08	91.32
Accusative	nan	0.00	nan	nan	nan	nan	nan	nan	nan	nan	0.00	nan
Dative	55.56	38.46	45.45	nan	0.00	nan	nan	0.00	nan	55.56	31.25	40.00

attitudes.)

It is clearly impossible to identify arguments only with knowledge of semantic relations between two nouns.

6.5 Adjective Predicate

We show the performance of our system SK for ADJECTIVE_I in Table 6.6 and ADJECTIVE_NA in Table 6.7.

Accounting to the tables, especially the performance of the accusative and dative cases is quite low, whereas F-values of the nominal case exceed 90. We consider the lack of knowledge of valency causes this.

According to Table 6.1, most adjective predicates do not have accusative and dative arguments. In fact, (Japanese descriptive grammar research group 2007) describes that adjective predicates generally have one argument to represent the owner of feeling, emotions, nature and so on. On the other hand, it also describes some adjective predicates have another argument to represent the targets of the feeling, emotions, nature and so on.

It classifies two-argument structures into four sentence patterns.

1. が, が
   In this pattern, the experiencer of the feeling is marked with "が" (ga) or "は" (wa) and the theme with "が" (ga). In NAIST Text Corpus, the first argument is annotated with nominative and the second is accusative.

(a) I wa-particle you ga-particle enjoyed ga-particle happy
私 は 君 が 楽しんでくれたこと が 嬉しい
(I am happy that you enjoyed it.)

(b) little sister ga-particle apple ga-particle like remind that
妹 が りんご が 好き だということを思いだした
(I remind that my little sister likes apples.)

2. が, に
In this pattern, the experiencer of the feeling is marked with "が" (ga) or "は" (wa) and the theme with "に" (ni). In NAIST Text Corpus, the first argument is annotated with nominative and the second is dative.

(a) father wa-particle his daughter ni-particle affectionate
父 は 娘 に 優しい
(The father is affectionate for his daughter.)

(b) he wa-particle research ni-particle crazy about
彼 は 研究 に 夢中だ
(He is crazy about his research.)

3. が, から
In this pattern, the owner of the characteristic is marked with "が'' (ga) or "は" (wa) and the source with "から" (kara). A few adjectives have this pattern. In NAIST Text Corpus, the first argument is annotated with nominative but the second is ignored to annotate.

(a) university wa-particle station kara-particle far
大学 は 駅 から 遠い
(The university is far from the station.)

4. が, と
In this pattern, the owner of the characteristic is marked with "が" (ga) or "は" (wa) and the target with "と" (to). A few adjectives have this pattern. In NAIST Text Corpus, the first argument is annotated with nominative but the second is ignored to annotate.

(a) Kyoto wa-particle Nara to-particle near
京都 は 奈良 と 近い
(Kyoto is near Nara.)

Following this, adjectives never have both accusative case and its dative case, and limited adjectives have another arguments.

Therefore we should exploit this linguistic characteristic for analysis by using valency data.

IPA Lexicon (IPA 1990) contains basic 136 “I”-adjectives and their valencies. For example, it describes that “明るい” (knowing much) can be a two-argument adjective and that its argument in the nominative case is human or organization like Tanaka (personal name) or the institution and that in the accusative case is the target like the classic literature or international affairs. Example 23 is an example in the IPA Lexicon.

Example 23:

Tanaka-san (personal name)	wa-particle	international affairs	ni-particle	knowing much
田中さん	は	国際情勢	に	明るい

(Tanaka-san knows international affairs much.)

Another valency resource is Kyoto University's case frame data (Kawahara and Kurohashi 2006)[1]. It is automatically constructed from 1.6 billion Japanese sentences on the Web. Each case frame is represented by a predicate and a set of its case filling words. It has about 40,000 predicates (including adjectives) and 13 case frames on average per each predicate. While IPA Lexicon has a few argument samples for an adjective usage, Kyoto University's case frame data has a number of sample arguments and also has frequency in a corpus for each argument.

Note that we may need to disambiguate meaning of the target predicate.

Example 24:

the street	ga-particle	light	because	safe
道	が	明るい	ので	安全だ

(It is safe because the street is light.)

For example the meaning of “明るい” in Example 24 is different to one in Example 23. It is clearly a different case frame and does not need an accusative argument.

[1] http://www.gsk.or.jp/en/catalog/gsk2008-b/

Chapter 7

Conclusion

7.1 Summary

In this dissertation, we proposed a Japanese predicate argument structure analysis model which collects the most likely candidates from all the groups and then selects the final candidate from among them.

In general, arguments are located close to the predicate. Previous work has exploited this characteristic to group candidates by positional relations between a predicate and its candidate arguments and then searched for the final candidate using a predetermined priority list of groups. However, in such analysis, candidates in different groups cannot be directly compared. Our model differs from them in the way that it can also take into account candidates from low priority and can perform global optimization for the final decision.

In the experiment, we compared our model and several deterministic models, in order to analyze how it is effective for analysis to divide a model considering positional relations. In the nominative and accusative cases, our model outperformed the deterministic models.

In Chapter 6, we discussed what kind of errors still remained and what approach is hopeful, by grouping errors depending on predicate types.

7.2 Future Directions

As shown in Chapter 6, there are various kinds of problems to be addressed. They vary according to predicate types. We believe we can solve them by address two main challenges. However, we summarize them by remarking two directions; knowledge acquisition and application, and context understanding.

Knowledge acquisition and application

For verb predicate, we showed some linguistic phenomena; light verb construction and case alternation. We also discussed the need of valency for dative and accusative analysis for adjective verb predicates. We plan to acquire more knowledge about such predicates, in order to improve accuracy for these cases.

Context understanding

As shown in errors in noun predicates, the remaining errors are difficult to solve without context understanding. It means we need to capture not only relations between arguments and predicates but also relations between argument structures.

Yoshikawa et al. (2011) proposed an approach using Markov Logic, which jointly analyzes predicate argument structures of all predicates in the same sentence. However, it has a drawback that computational effort rapidly increases when it explores more candidates. In English semantic role labeling, Laparra and Rigau (2013) proposed a deterministic algorithm to capture shared arguments between predicates. We intend to adopt this algorithm in Japanese predicate argument structure analysis.

Bibliography

[Chinchor, Hirschman, and Lewis] Chinchor, N., Hirschman, L., and Lewis, D. D. (1993). "Evaluating Message Understanding Systems: An Analysis of the Third Message Understanding Conference (MUG-3)." *Computational Linguistics*, **19** (3), pp. 409–449.

[Cortes and Vapnik] Cortes, C. and Vapnik, V. (1995). "Support-Vector Networks." *Machine Learning*, **20** (3), pp. 273–297.

[Fujita, Inui, and Matsumoto] Fujita, A., Inui, K., and Matsumoto, Y. (2004). "Detection of Incorrect Case Assignments in Automatically Generated Paraphrases of Japanese Sentences.." In *Proceedings of 1st International Joint Conference on Natural Language Processing*, pp. 14–21.

[Hashida] Hashida, K. (2005). "Global document annotation (GDA) annotation manual." http://i-content.org/gda/tagman.html.

[Hashimoto, Kurohashi, Kawahara, Shinzato, and Masaaki] Hashimoto, C., Kurohashi, S., Kawahara, D., Shinzato, K., and Masaaki, N. (2011). "Construction of a Blog Corpus with Syntactic, Anaphoric, and Sentiment Annotations (in Japanese)." *Journal of natural language processing*, **18** (2), pp. 175–201.

[Hindle] Hindle, D. (1990). "Noun Classification from Predicate-Argument Structures." In *Proceedings of the 28th Annual Meeting on Association for Computational Linguistics*, pp. 268–275.

[Hovy, Marcus, Palmer, Ramshaw, and Weischedel] Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). "OntoNotes: the 90% solution." In *Proceedings of the Human Language Technology Conference of the NAACL*, pp. 57–60.

[Iida, Inui, and Matsumoto] Iida, R., Inui, K., and Matsumoto, Y. (2005). "Anaphora Resolution by Antecedent Identification Followed by Anaphoricity Determination." *ACM Transactions on Asian Language Information Processing*, **4** (4), pp. 417–434.

[Iida, Inui, and Matsumoto] Iida, R., Inui, K., and Matsumoto, Y. (2007). "Zero-anaphora Resolution by Learning Rich Syntactic Pattern Features." *ACM Transactions on Asian Language Information Processing*, **6** (4), pp. 1:1–1:22.

[Iida, Inui, Takamura, and Matsumoto] Iida, R., Inui, K., Takamura, H., and Matsumoto, Y. (2003). "Incorporating Contextual Cues in Trainable Models for Coreference Resolution." In *Proceedings of the 10th EACL Workshop on the Computational Treatment of Anaphora*, pp. 23–30.

[Iida, Komachi, Inoue, Inui, and Matsumoto] Iida, R., Komachi, M., Inoue, N., Inui, K., and Matsumoto, Y. (2010). "Annotating Predicate-Argument Relations and Anaphoric Relations : Findings from the Building of the NAIST Text Corpus (in Japanese)." *Journal of Natural Language Processing*, **17** (2), pp. 25–50.

[Iida, Komachi, Inui, and Matsumoto] Iida, R., Komachi, M., Inui, K., and Matsumoto, Y. (2007). "Annotating a Japanese Text Corpus with Predicate-Argument and Coreference Relations." In *Proceedings of the Linguistic Annotation Workshop*, pp. 132–139.

[Ikehara, Miyazaki, Shirai, Yokoo, Nakaiwa, Ogura, Ooyama, and Hayashi] Ikehara, S., Miyazaki, M., Shirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y., and Hayashi, Y. (Eds.) (1997). *Nihongo Goi Taikei, A Japanese Lexicon (in Japanese)*. Iwanami Shoten.

[Imamura, Saito, and Izumi] Imamura, K., Saito, K., and Izumi, T. (2009). "Discriminative Approach to Predicate-Argument Structure Analysis with Zero-Anaphora Resolution." In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pp. 85–88.

[IPA] IPA (1990). "IPA Lexicon of the Japanese language for computers (Basic Adjectives) (in Japanese).".

[Kawahara and Kurohashi] Kawahara, D. and Kurohashi, S. (2004). “Zero Pronoun Resolution Based on Automatically Constructed Case Frames and Structural Preference of Antecedents.” In *Proceedings of the 1st International Joint Conference on Natural Language Processing*, pp. 334–341.

[Kawahara and Kurohashi] Kawahara, D. and Kurohashi, S. (2006). “Case Frame Compilation from the Web using High-Performance Computing.” In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pp. 1344–1347.

[Kawahara, Kurohashi, and Hasida] Kawahara, D., Kurohashi, S., and Hasida, K. (2002). “Construction of a Japanese Relevance-tagged Corpus.” In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pp. 2008–2013.

[Kipper, Dang, and Palmer] Kipper, K., Dang, H. T., and Palmer, M. (2000). “Class-based Construction of a Verb Lexicon.” In *Proceedings of the 7th National Conference on Artificial Intelligence*, pp. 691–696.

[Komachi and Iida] Komachi, M. and Iida, R. (2011). “Annotating Predicate-Argument Structure and Anaphoric Relations to BCCWJ (in Japanese).” In *Japanese Corpus Workshop*, pp. 325–330.

[Komachi, Iida, Inui, and Matsumoto] Komachi, M., Iida, R., Inui, K., and Matsumoto, Y. (2007). “Learning-Based Argument Structure Analysis of Event-Nouns in Japanese.” In *Proceedings of the Conference of the Pacific Association for Computational Linguistics*, pp. 120–128.

[Kudo and Matsumoto] Kudo, T. and Matsumoto, Y. (2004). “A Boosting Algorithm for Classification of Semi-Structured Text.” In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 301–308.

[Kurohashi and Nagao] Kurohashi, S. and Nagao, M. (2003). “Building a Japanese Parsed Corpus.” In *Treebanks: Building and Using Parsed Corpora*, chap. 14, pp. 249–260.

[Laparra and Rigau] Laparra, E. and Rigau, G. (2013). “ImpAr: A Deterministic Algorithm for Implicit Semantic Role Labelling.” In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 1180–1189.

[Matsubayashi, Iida, Sasano, Yokono, Matsuyoshi, Fujita, Miyao, and Inui] Matsubayashi, Y., Iida, R., Sasano, R., Yokono, H., Matsuyoshi, S., Fujita, A., Miyao, Y., and Inui, K. (2013). "Analysis of Problems in Annotation of Japanese Japanese Predicate Argument Structure (in Japanese)." In *Proceedings of Information Processing Society of Japan 214th Meeting of SIG Natural Language Processing*, pp. 12:1–18.

[Meyers, Reeves, Macleod, Szekely, Zielinska, Young, and Grishman] Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., and Grishman, R. (2004). "The NomBank Project: An Interim Report." In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pp. 24–31.

[Miyamoto] Miyamoto, T. (1999). *The Light Verb Construction in Japanese: the role of the verbal noun*. John Benjamins Publishing Company.

[Nariyama] Nariyama, S. (2002). "Grammar for Ellipsis Resolution in Japanese." In *Proceedings of the 9th International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 135–145.

[Palmer, Gildea, and Kingsbury] Palmer, M., Gildea, D., and Kingsbury, P. (2005). "The Proposition Bank: An Annotated Corpus of Semantic Roles." *Computational Linguistics*, **31** (1), pp. 71–106.

[Ruppenhofer, Ellsworth, Petruck, Johnson, and Scheffczyk] Ruppenhofer, J., Ellsworth, M., Petruck, M. R. L., Johnson, C. R., and Scheffczyk, J. (2006). "FrameNet II: Extended Theory and Practice.".

[Sasano, Kawahara, Kurohashi, and Okumura] Sasano, R., Kawahara, D., Kurohashi, S., and Okumura, M. (2013). "Automatic Knowledge Acquisition for Case Alternation between the Passive and Active Voices in Japanese." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1213–1223.

[Sasano and Kurohashi] Sasano, R. and Kurohashi, S. (2009). "A Probabilistic Model for Associative Anaphora Resolution." In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 1455–1464.

[Sasano and Kurohashi] Sasano, R. and Kurohashi, S. (2011). "A Discriminative Approach to Japanese Zero Anaphora Resolution with Large-scale Case Frames (in Japanese)." *Journal of Information Processing*, **52** (12), pp. 3328–3337.

[Shen and Lapata] Shen, D. and Lapata, M. (2007). "Using Semantic Roles to Improve Question Answering." In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 12–21.

[Surdeanu, Harabagiu, Williams, and Aarseth] Surdeanu, M., Harabagiu, S., Williams, J., and Aarseth, P. (2003). "Using Predicate-Argument Structures for Information Extraction." In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, Vol. 1, pp. 8–15.

[Taira, Fujita, and Nagata] Taira, H., Fujita, S., and Nagata, M. (2008). "A Japanese Predicate Argument Structure Analysis Using Decision Lists." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 523–532.

[Takeuchi, Inui, Takeuchi, and Fujita] Takeuchi, K., Inui, K., Takeuchi, N., and Fujita, A. (2010). "A Thesaurus of Predicate-Argument Structure for Japanese Verbs to Deal with Granularity of Verb Meanings." In *Proceedings of the Eighth Workshop on Asian Language Resouces*, pp. 1–8.

[The Mainichi Newspapers] The Mainichi Newspapers (1991-2003). "The Mainichi Newspapers CD-ROM.".

[Wang and Zhang] Wang, R. and Zhang, Y. (2009). "Recognizing Textual Relatedness with Predicate-Argument Structures." In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 784–792.

[Watanabe, Asahara, and Matsumoto] Watanabe, Y., Asahara, M., and Matsumoto, Y. (2010). "A Structured Model for Joint Learning of Argument Roles and Predicate Senses." In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*, pp. 98–102.

[Wu and Fung] Wu, D. and Fung, P. (2009). "Can Semantic Role Labeling Improve SMT?." In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, pp. 218–225.

[Yoshikawa, Asahara, and Matsumoto] Yoshikawa, K., Asahara, M., and Matsumoto, Y. (2011). "Jointly Extracting Japanese Predicate-Argument Relation with Markov Logic." In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 1125–1133.

[Japanese descriptive grammar research group] Japanese descriptive grammar research group (2007). *Modern Japanese Grammar 2 (in Japanese)*. Kurosio Publishers.

[Japanese descriptive grammar research group] Japanese descriptive grammar research group (2010). *Modern Japanese Grammar 1 (in Japanese)*. Kurosio Publishers.

業績リスト

論文誌

1. 林部 祐太, 小町 守, 松本 裕治：述語と項の位置関係ごとの候補比較による日本語述語項構造解析, 自然言語処理, Vol. 21, No. 1, pp. 3–26 (2014)

国際会議 (査読あり)

2. Hayashibe, Y., Hagiwara, M., and Sekine, S.: phloat : Integrated Writing Environment for ESL learners, in *Proceedings of the Second Workshop on Advances in Text Input Methods*, pp. 57–72, Mumbai, India (2012)

3. Mizumoto, T., Hayashibe, Y., Komachi, M., Nagata, M., and Matsumoto, Y.: The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings, in *Proceedings of COLING 2012: Posters*, pp. 863–872, Mumbai, India (2012)

4. Hayashibe, Y., Komachi, M., and Matsumoto, Y.: Japanese Predicate Argument Structure Analysis Exploiting Argument Position and Type, in *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 201–209, Chiang Mai, Thailand (2011)

国際会議 (査読なし)

5. Yoshimoto, I., Kose, T., Mitsuzawa, K., Sakaguchi, K., Mizumoto, T., Hayashibe, Y., Komachi, M., and Matsumoto, Y.: NAIST at 2013 CoNLL Grammatical Error Correction Shared Task, in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pp. 26–33, Sofia, Bulgaria (2013)

6. Mizumoto, T., Hayashibe, Y., Sakaguchi, K., Komachi, M., and Matsumoto, Y.: NAIST at the NLI 2013 Shared Task, in *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 134–139, Atlanta, Georgia (2013)

7. Mizumoto, T., Hayashibe, Y., Komachi, M., Nagata, M., and Matsumoto, Y.: The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings, in *Proceedings of COLING 2012: Posters*, pp. 863–872, Mumbai, India (2012)

研究会・全国大会等 (査読なし)

8. 林部 祐太, 萩原 正人, 関根 聡：英作文統合支援環境 phloat, 言語処理学会第 19 回年次大会, pp. 802–805, 名古屋 (2013)

9. 水本 智也, 林部 祐太, 小町 守, 永田 昌明, 松本 裕治：大規模英語学習者コーパスを用いた英作文の文法誤り訂正の課題分析, 情報処理学会第 209 回自然言語処理研究会, pp. 5:1–8, 京都 (2012)

10. 水本 智也, 林部 祐太, 坂口 慶祐, 小町 守, 松本 裕治：英作文誤り訂正における複数の手法の利用に関する考察, 情報処理学会第 208 回自然言語処理研究会, pp. 8:1–7, 仙台 (2012)

11. 林部 祐太, 小町 守, 松本 裕治, 隅田 飛鳥：日本語テキストに対する述語語義と意味役割のアノテーション, 言語処理学会第 18 回年次大会, pp. 397–400, 広島 (2012)

12. Laura, S. E., 林部 祐太, 松本 裕治：現代日本語におけるコロケーション：検出と分析, 第 1 回コーパス日本語学ワークショップ, pp. 53–58, 東京 (2012)

13. 林部 祐太, 小町 守, 松本 裕治：文脈情報と格構造の類似度を用いた日本語文間述語項構造解析, 情報処理学会第 201 回自然言語処理研究会, pp. 10:1–8, 東京 (2011)

受賞

14. Error Detection and Correction Workshop 2012, 前置詞トラック:最優秀賞, 動詞トラック:優秀賞, オープントラック: 奨励賞 (2012.09.04), チーム NAIST (林部 祐太，坂口慶祐，水本智也) のリーダー

15. 情報処理学会第 201 回自然言語処理研究会 学生奨励賞 (2011.05.17)