Informed Parsing for Coordination with Combinatory Categorial Grammar

Jong C. Park Hyung Joon Cho park@cs.kaist.ac.kr hjcho@nlp.kaist.ac.kr Computer Science Division and Advanced Information Technology Research Center Korea Advanced Institute of Science and Technology 373-1 Gusung-dong, Yusong-gu, Daejon 305-701, KOREA

Abstract

Coordination in natural language hampers efficient parsing, especially due to the multiple and mostly unintended candidate conjuncts/disjuncts in a given sentence that shows structural ambiguity. The problem gets more serious in a combinatory categorial grammar framework, which is well known for its competent treatment of coordination, as the flexibility of syntactic analysis often strikes back as spurious ambiguity. We propose to address these ambiguities with predicate argument structures and semantic co-occurrence similarity information, and present encouraging results.

1 Introduction

Sentences with coordination contain multiple phrases of like syntactic type. When the given sentence shows structural ambiguity, there may be multiple pairs of candidates for possible conjuncts/disjuncts, usually a single pair of which is identified as intended by human language understanders. Parsing for coordination should thus find the exact syntactic boundaries of these "intended" conjuncts/disjuncts. Previous work employed a preprocessing module for parsing to work on a constrained range of the candidate conjuncts/disjuncts (Kurohashi & Nagao, 1994; Yang, 1995; Okumura & Muraki, 1994).

Combinatory categorial grammars (Steedman, 1990; 2000) are known to explain a wide range of syntactic phenomena, such as coordination, extraction and long distance dependency, without employing the notions of movement and empty categories. While CCGs offer explanations for various natural language phenomena with limited combinatory rules such as type raising and function composition, these are also well known to increase the complexity of parsing, giving rise to quite a few irrelevant syntactic analyses as well as relevant ones, a phenomenon often called as *spurious ambiguity*.

In this paper, we propose to address the two types of ambiguity with predicate argument structures and semantic co-occurrence similarity information.¹ For a more concrete discussion, we focus in this paper on coordination in Korean. First, related work is reviewed in §2. The two types of ambiguity are then discussed in §3. We also examine the characteristics of coordination in a corpus in §4. In §5, we show our proposal to enhance parsing efficiency, with encouraging experimental results.

2 Related Work

2.1 Conjunct Identification

First, we review three of the techniques that attempt to narrow down the candidate conjuncts.

2.1.1 Complex Information

Approaches that use complex information are based on the assumption that there are various clues for morphological and semantic similarity between the pair of matching conjuncts/disjuncts. The usual measure for such similarity includes the part-of-speech (POS) feature information. The proposal by Agarwal and Boggess (1992) for English coordination utilizes a semi-parser for the assignment of pertinent semantic and morphological information to each lexical item in a given sentence. The procedure for the conjunct/disjunct identification with this information is summarized below.

1. Keep pushing the lexical items in a given sentence into the stack until a coordination item such as *and*, *or*, *but*, etc, is encountered.

¹This work was supported by the Korea Science and Engineering Foundation (KOSEF) through AITrc. The second author is now with SK Teletech in Korea.

- 2. With the coordination item, treat the immediately following phrase as the *post*-conjunct/disjunct.
- 3. Pop the lexical items from the stack one by one, comparing their morphological and/or semantic features with those of the *post*-conjunct/disjunct.

The reported precision is 81%, but the method identifies only the starting positions of the conjuncts/disjuncts.

The method proposed by Okumura and Muraki (1994) for English coordination looks into the symmetric patterns of conjuncts/disjuncts. These symmetric patterns are classified into four categories: phrasal/clausal patterns, lexical patterns, morphological patterns, and complex patterns. The first three patterns are defined in terms of the respective features in lexical items. The best match among all the possible word sequences with these features before and after the coordination item is passed over to the parser. The reported precision is 75%. This method assigns various weights to the features for the measure of the symmetric patterns. It is not clear if the weight assignment method is principled and if it can also address conjuncts/disjuncts with ellipsis.

2.1.2 Co-Occurrence Information

Yang (1995) utilized co-occurrence information for the resolution of structural ambiguity in Korean noun phrase coordination. The method looks up the related pair of nouns and verbs from a large corpus and uses the statistics on the case information of the nouns with respect to the given verb for the similarity information among nouns. The co-occurrence similarity $DSim(n_1, n_2)$ between the nouns n_1 and n_2 is defined as follows, which incorporates the prediction that the similarity of the two nouns is higher when they are more frequently used with the same verb of the same syntactic category.

$$DSim(n_1, n_2) \equiv \frac{2 \cdot \sum_{g \in G} |CV_g(n_1, n_2)|}{\sum_{g \in G} |V_g(n_1)| + \sum_{g \in G} |V_g(n_2)|}$$

where²

- $G \equiv \{subj, obj, loca, inst, modi\}$
- $V_g(n) \equiv \{v | v \text{ is a verb such that } f_g(n, v) \ge 1\}$
- $|V_g(n)| \equiv \sum_{v \in V_g(n)} f_g(n, v)$

- $CV_g(n_1, n_2) \equiv V_g(n_1) \cap V_g(n_2)$
- $|CV_g(n_1, n_2)| \equiv \sum_{v \in CV_g(n_1, n_2)} min\{f_g(n_1, v), f_g(n_2, v)\}$

This method makes it possible to resolve the syntactic ambiguity due to coordination in Korean as illustrated in (1) and (2).³ DSim is used to predict correctly that 'suthayk' (stack) is in coordination with 'khyu' (queue) and not with 'yey' (example). Likewise, 'kyesan' (computation) and 'thamsayk' (search) are correctly identified to be in coordination with each other.

- (1) suthayk-kwa khyu-uy yey stack-co queue-poss example example(s) of stack(s) and queue(s) DSim(suthayk, khyu) = 0.319DSim(suthayk, yey) = 0.053
- (2) haysing hamswu-uy kyeysan kwa pekheys-uy thamsayk-ey soyotoynun sikan hashing function-poss computation-co bucket-poss search-loc spending time the time spent for the computation of the hashing function and the search for the bucket DSim(kyeysan, pekheys) = 0.024DSim(kyeysan, thamsayk) = 0.303DSim(kyeysan, sikan) = 0.067

In addition to the fact that the method may suffer from data sparseness, it is also not directly applicable when nouns are polysemous or when the conjunct/disjunct nouns have weak semantic similarity (cf. (3)).

(3) i notu-nun teyitha yoso-wa lisutu-uy taum wenso-lul cisiha-nun phointhe-lul phohamha-n-ta this node-nom data element-co list-poss next element-acc indicate-un pointer-acc contain-presdecl
'this node contains the data element and the pointer that indicates the next element in the list'

Its reported precision is 85.8%, and the recall 95.9%. We note that this co-occurrence similarity information is useful when large-scale linguistic knowledge bases such as WordNet are not available for the language in question.

2.1.3 Part-Of-Speech Patterns

The method with POS patterns (Park, 1998) extracts sentences with coordination from a POS-tagged corpus, trains the system with the POS patterns of the left- and rightconjuncts/disjuncts, and resolves the ambiguity with trained POS patterns (cf. (4)).

 $^{{}^{2}}f_{g}(n,v)$ is the number of times that noun n of type g occurs with verb v in the same sentence in a corpus.

 $^{^{3}}$ Coordination items are shown in a box. We use the Yale notation for transcribing Korean alphabets.

(4) na-nun [kemjeng ppang-kwa huyn ppang]-ul mekun salam-ul poass-ta
I-nom [black bread-co white bread]-acc eat-un person-acc see-past-decl
'I saw the person who ate black bread and white bread'

The longest pattern is selected when there are multiple candidate POS-patterns. The reported precision is 71.6%. When there is structural ambiguity as shown in (5) below, however, it is difficult to identify the right conjunct if the system considers only POS information, where the longest match is 'mangko-lul swuipha-nun nala' (the nation that imports mangos), not the intended 'mangko' (mango).

(5) <u>phainayphul-</u> <u>ina</u> <u>mangko</u>-lul swuipha-nun <u>nala</u> pineapple-co mango-acc import-un nation 'the nation that imports pineapples or mangos'

2.2 Combinatory Categorial Grammar for Korean

Examples of coordination in Korean are shown below.

- (6) [chelswu-nun uysa-ka], [yenghi-nun sensayngnim-i] toy-ess-ta
 [chelswu-top doctor-comp], [yenghi-top teach-comp] become-past-decl
 'Chelswu became a doctor, and Yenghi a teacher'
- (7) [kochwukapsi olu-myen kochwu-lul], [tway-cikokikapsi olu-myen twaycikoki-lul] swuipha-n-ta.
 [pepper-price-nom rise-if pepper-acc], [pork-price-nom rise-if pork-acc] import-pres-decl

'(The government) imports peppers if the price for peppers rises, and pork if the price for pork rises'

In CCGs, type raising and function composition rules are typically utilized to come up with single categories for those fragments above in square brackets, often called 'non-standard' constituents. Table 1 shows the reduction rules proposed for Korean (Cho, 2000; Cho and Park, 2000). Figure 1 shows a sample syntactic and semantic derivation of part of (6). Space precludes further explanation of the formalism.

3 Two Types of Ambiguity

3.1 Spurious Ambiguity

CCGs provide two syntactic derivations for the semantically unambiguous sentence (8), as shown in (9) and (10).

(8) chelswu-ka sakwa-lul mek-ess-ta Chelswu-nom apple-acc eat-past-decl

- 1. [[chelswu-nun sakwa-lul] mek]-ko [[yenghi-nun ttalki-lul] mek]-nunta
- 2. [[chelswu-nun sakwa-lul] mek]-ko [yenghi-nun [ttalki-lul mek]]-nunta
- 3. [chelswu-nun [sakwa-lul mek]]-ko [[yenghi-nun ttalki-lul] mek]-nunta
- 4. [chelswu-nun [sakwa-lul mek]]-ko [yenghi-nun [ttalki-lul mek]]-nunta

 Table 2: Example Spurious Ambiguity

'Chelswu ate an apple/apples'

- (9) [[chelswu-ka sakwa-lul] mek-ess-ta]
- (10) [chelswu-ka [sakwa-lul mek-ess-ta]]

These distinct syntactic derivations make it possible for a CCG to correctly analyze sentences with coordination, as shown in (11) and (12).

- (11) [[chelswu-ka sakwa-lul] [yenghi-ka ttalki-lul]] mek-ess-ta] 'Chelswu ate apples and Yenghi strawberries'
- (12) [chelswu-ka [[son-ul ssis]-ko [sakwa-lul mek-ess]ta]] 'Chelswu washed his hands and ate apples'

Spurious ambiguity refers to the phenomenon of this kind in which there are multiple syntactic derivations, as in (9) and (10), for a fragment that is semantically unambiguous.⁴ While descriptively justifiable, it nevertheless results in a repeated computation of the same fragments that are semantically indistinct, adversely affecting parsing efficiency. This problem gets more serious with coordination (cf. (13)).

(13) chelswu-nun sakwa-lul mek-<u>ko</u>yenghi-nun ttalkilul mek-nun-ta 'Chelswu eats apples and Yenghi strawberries'

Table 2 shows four syntactic derivations for (13), all with identical semantics.

3.2 Structural Ambiguity

The spurious ambiguity as discussed above does not give rise to wrong syntactic analyses, but the structural ambiguity may, as shown in (14).

(14) cengchi-uy canglay-na nongmintul-uy saynghwalsang-uy mwuncay-wa-nun keli-ka mel-ta

⁴Note that there are arguments that some of these syntactic derivations are associated with distinct pragmatic functions, so that the ambiguity might not be entirely 'spurious' (Prevost, 1995).

Reduction Rule	Rule Name	Rule Symbol
$X/Y \ Y \to X$	Forward Application	>
$Y \ X \backslash Y \to X$	Backward Application	<
$X \ conj \ X \to X$	Coordination	$<\phi^n>$
$X/Y \ Y/Z \to X/Z$	Forward Composition	> B
$Y \setminus Z X \setminus Y \to X \setminus Z$	Backward Composition	< B
$X/Y \ Y \setminus Z \to X \setminus Z$	Forward Crossed Composition	$> B_x$
$X \to T/(T \backslash X)$	Forward Type Raising	>T
$X \to T \backslash (T/X)$	Backward Type Raising	$<\overline{T}$

Table 1: Reduction Rules in a CCG for Korean



 $: \lambda f. and' (f \ doctor' chelswu', f \ teacher' yenghi')$

Figure 1: Sample CCG Derivation

politics-poss future-co farmers-poss life-poss problem-wa-top distance-nom far-decl

(It is) far from the problems of the future of politics or the lives of farmers'

Table 3 shows six of the syntactic derivations. While the semantics makes it clear that only the derivation 2 is the intended one, it is impossible for a parser with only syntactic information to tell the difference. Notice that this problem is not unique to a CCG-based parser. While the general solution would obviously require not only semantic information but also pragmatic and discourse information, we examine approaches that take into account only semantic information in this paper.

4 Coordination in Corpus

In an attempt to assess the coordination phenomenon in a realistic manner, we examine a POS-tagged corpus available at KAIST. The corpus contains newspaper articles (40,428 eojeol), essays (41,666 eojeol), textbooks (50,208 eojeol), technical documents (2,729 eojeol), novels (40,498 eojeol), with the total of 175,524 eojeol and 17,123 sentences.⁵ Table 4 shows the types

- 1. [cengchi-uy canglay]-na [nongmintul]-uy saynghwalsang-uy mwuncey
- 2. [cengchi-uy canglay]-na [nongmintul-uy saynghwalsang]-uy mwuncey
- 3. [cengchi-uy canglay]-na [nongmintul-uy saynghwalsang-uy mwuncey]
- 4. cengchi-uy [canglay]-na [nongmintul]-uy saynghwalsang-uy mwuncey
- 5. cengchi-uy [canglay]-na [nongmintul-uy saynghwalsang]-uy mwuncey
- 6. cengchi-uy [canglay]-na [nongmintul-uy saynghwalsang-uy mwuncey]

Table 3: Example Structural Ambiguity

and frequencies of sentences containing coordination in this corpus. We used PERL scripts to narrow down the sentences meeting certain basic conditions for coordination and manually identified sentences with coordination among those chosen.⁶ Table 4 indicates that coordination is used quite often in Korean.

5 Parsing for Coordination

In this section, we present techniques of dealing with coordination for efficient parsing.

⁵Eojeol is a unit in Korean that roughly corresponds to space-delimited words in English. Each eojeol contains both the stem and its morphological endings.

⁶A POS-tagged corpus does not give sufficient information for the fully automatic identification of sentences with coordination, since we also need to take the sentential semantics into account.

Coordination Item	Articles	Essays	Text books	Documents	Novels	Total
ending	728~(21.4%)	1092~(33.7%)	1101~(20.1%)	46~(40%)	2174~(44.5%)	5142~(30.0%)
postposition	671 (19.7%)	352~(10.7%)	812~(14.8%)	23~(20%)	320~(6.6%)	2178~(12.7%)
adverb	91 (2.7%)	22~(0.68%)	58~(1.1%)	6~(5.2%)	17~(0.35%)	194 (1%)
comma	154 (4.5%)	126~(3.9%)	216~(3.9%)	21~(18.3%)	43~(0.88%)	605~(3.5%)
Total Sentences	3403	3239	5485	115	4881	17123~(100%)

 Table 4: Characteristics of Coordination in Korean

5.1 Predicate Argument Structure

We used the CKY algorithm to implement a CCG parser. As discussed, the presence of spurious ambiguity in a given sentence forces repeated syntactic analyses for fragments with identical semantics. This can be avoided if we use the same cell to record the syntactic analyses with the same semantics.⁷ CCG makes this possible, as both the syntactic and semantic derivations are constructed in tandem. Table 5 shows part of the parsing table for (13). The following shows the relevant syntactic categories.

- chelswu-nun, yenghi-nun : $s/(s \setminus np_n)$
- sakwa-lul, ttalki-lul : $(s/np_n)/(s np_n np_a)$
- mek : $s \setminus np_n \setminus np_a$
- ko : conj

The cells (C1,R3) and (C5,R3) each contain two syntactic analyses with the same semantics, resulting in four syntactic analyses with the same semantics in the cell (C1,R7). We can prevent such multiple analyses by not writing into the cell a syntactic analysis with the same recorded semantics. We thus have one syntactic analysis for each of the cells (C1,R3), (C5,R3)and (C1,R7). For longer sentences, we expect a significant reduction in the number of derived syntactic analyses, as also partly verified by our experiments.

5.2 Co-Occurrence Similarity

5.2.1 The Algorithm

Among the pairs of head nouns of candidate conjuncts/disjuncts, about 88.1% pairs are identified as semantically related in our corpus. It is thus reasonable to consider only those candidates with some semantic relation. We use the following modification of Yang's (1995) original proposal.

- 1 Whenever the coordination reduction rule is invoked, check the syntactic categories of the candidate conjuncts.
- 2 If they are nouns or noun phrases, skip to the next step. Otherwise write them to a cell.
- 3 Locate the head nouns in the candidate conjuncts.
- 4 Compute the co-occurrence similarity of the head nouns.
- 5 If the coordination reduction rule has already been applied to the head noun of the left candidate conjunct, compare the co-occurrence similarity of the recorded and the new. If the co-occurrence similarity of the newly identified candidate conjuncts is stronger, write them to a cell, and delete the existing candidates in the cell. Otherwise, discard the new and retain the old.
- 6 Update the list of conjuncts whenever there is a newly recorded candidate conjunct.

For the co-occurrence similarity information between nouns, we used another KAIST corpus that is manually tree-tagged (Lee, 1998). It contains about 31,000 sentences with 352,730 eojeol. The average sentence length is 11.35 words. The domains include newspaper editorials, economy, religion, science fiction, expedition, novels, and history. We have considered only nominative, accusative, adverbial, complement, and adnominal cases in relation to the verbs for the extraction of co-occurrence similarity information, which is then recorded into a dictionary and is looked up by the parser when it deals with coordination.

5.2.2 Thesaurus

The use of co-occurrence similarity information as in (Yang, 1995) suffers from data sparseness, especially since we have a relatively small corpus with fairly unrestricted domains.⁸ For instance, (15) and (16) below show examples of wrong co-occurrence similarity information.

(15) <u>kimchi-wa pap(0)-man cwu-nun kes(0.002252)-ita</u> kimchee-co steamed rice-only give-top thing-decl.
'(They) served only kimchee and steamed rice' kimchi: 3 verbs, pap: 9 verbs, kes: 2083 verbs

 $^{^7\}mathrm{Cf.}$ Karttunen, 1989; Eisner, 1996; Komagata, 1999

⁸In contrast, Yang used a corpus with one million eojeol and restricted to a computer science domain.

R7	s and'mekta'tsakwa'chelswu' mekta'ttalki'yenghi' [[chelswu-ka sakwa-lul] mek]ko [[yenghi-ka ttalki-lul] mek] [[chelswu-ka sakwa-lul] mek]ko [yenghi-ka [ttalki-lul] mek] [chelswu-ka [sakwa-lul] mek] [chelswu-ka [sakwa-lul] mek] [chelswu-ka [sakwa-lul] mek] [chelswu-ka [sakwa-lul] mek] [chelswu-ka [sakwa-lul] mek] [chelswu-ka [sakwa-lul] mek]]ko [yenghi-ka [ttalki-lul] mek]]						
:							
R3	s mekta'sakwa'chelswu' [[chelswu-ka sakwa-lul]mek], [chelswu-ka [sakwa-lul mek]] (C1,R2)+(C3,R1), (C1,R1)+(C2,R2)				s mekta'ttalki'yenghi' [[yenghi-ka ttalki-lul]mek], [yenghi-ka [ttalki-lul mek]] (C5,R2)+(C7,R1), (C5,R1)+(C6,R2)		
R2	$s/(s \setminus np_n \setminus np_a)$ $\lambda f. sak wa'chelswu'[chelswu-ka sakwa-lul](C1,R1)+(C2,R1)$	$s \setminus np_n$ λ y.mekta'sakwa'y [sakwa-lul mek] (C2,R1)+(C3,R1)			$s/(s \setminus np_n \setminus np_a)$ λ f.ttalki'yenghi' [yenghi-ka ttalki-lul] (C5,R1)+(C6,R1)	$s \setminus np_n$ $\lambda y.mekta'ttalki'y$ [ttalki-lul mek] (C6,R1)+(C7,R1)	
R1	chelswu-ka	sakwa-lul	mek	ko	yenghi-ka	ttalki-lul	mek
	C1	C2	C3	C4	C5	C6	C7

Table 5: Sample CKY Parsing Table

 $\begin{array}{c} \textbf{(16)} \ \underline{\text{os-kwa}} \ \ \underline{\text{cangsingkwu}}(0)\text{-lul} \ \ \ \underline{\text{yenkwuha-nun}} \\ \underline{\text{kes}}(0.008647)\text{-iess-supnita}. \end{array}$

os: 46 verbs, cangsinkwu: 2 verbs, kes: 2083 verbs

In (15), there are three verbs that occur with 'kimchi' (kimchee), and nine verbs that occur with 'pap' (steamed rice), significantly fewer than those verbs that occur with 'kes' (thing). And in (16), the number of verbs that occur with 'cangsinkwu' (accessory) is smaller than that of the verbs that occur with other nouns. Both result in a wrong analysis. We can use a thesaurus to address this problem.

In a thesaurus, words in the same class are assumed to have related meanings. We can use these class-mate words to compensate for data sparseness. In constructing a lexicon, we consult the thesaurus when the number of verbs that occur with a given noun falls below a threshold, and let the noun share the data with those in the same class. The thesaurus has the 'word-meaning code' format. The present thesaurus contains slightly more than 1000 nouns that are manually constructed. The classification follows the NTT hierarchy. We have assigned meaning codes to only the most frequently used meanings for polysemous entries. The depth of the hierarchy is 6. The following shows adjusted results with our thesarus.

(17) $\underline{\text{kimchi}}$ -wa $\underline{\text{pap}}(0.768942)$ -man cwu-nun $\underline{\text{kes}}(0.008380)$ -ita.

kimchi: 62 verbs, pap: 64 verbs, kes: 2083 verbs

(18) <u>os</u> kwa cangsinkwu(0.648276)-lul yenkwuha-nun $\underline{\text{kes}}(0.008647)$ iess-supnita.

os: 46 verbs, cangsinkwu: 69 verbs, kes: 2083 verbs

Table 6 shows the comparison of the methods with various co-occurrence similarity dictionaries using 84 sentences containing noun phrase coordination and structural ambiguity.⁹ It shows that a thesaurus is indeed useful in dealing with data sparseness. In this experi-

	M1	M2
Precision	84.1%	88.5%
Recall	95.3%	92.7%

 Table 6: Comparison of Different Methods

ment, we have shared the nouns that are associated with fewer than 20 verbs. We have also set the maximum shared examples to 70 and tuned the figures for maximum precision.

5.3 Results

For the performance evaluation, we have compared three kinds of parsers. A employs only the CKY algorithm. B has the additional module for spurious ambiguity. C utilizes the aforementioned dictionary, in addition to the module for spurious ambiguity. We found out that the

 $^{^{9}}$ M1 utilizes only the similarly dictionary as defined by Yang (1995). M2 has the extra information from KAIST tree-tagged corpus and the thesaurus.

primary factors for the extra parsing complexity include the number of right conjunct candidates, the presence of modifiers in the left conjunct, and the type of sentences (simple or complex) with noun phrase coordination.

In order to check for the influence of structural ambiguity on parsing efficiency, we have classified 53 sentences into 4 types, considering modifiers in the left conjunct (cf. Table 7).¹⁰ Table 8 shows the number of semantic struc-

Sentences	Modifiers (range)	# of candidates
Type 1	no	≤ 3
Type 2	yes (unambiguous)	≥ 4
Type 3	yes (ambiguous)	≤ 3
Type 4	yes (ambiguous)	≥ 4

Table 7: Types of Sentences

tures as derived by each parser.¹¹ The average numbers of semantic structures derived by B and C are 26.2 and 7.3, respectively, resulting in the reduction of 72.1%. Table 9 shows

	A	В	С
Type 1	189.8	4.3	2.0
Type 2	963	32.3	12.3
Type 3	245.1	13.4	5.4
Type 4	-	83.3	16.4

 Table 8: Derived Semantic Structures

the parsing time for each method.¹² The reason that B appears generally faster than C (except for type 4) is that C spends extra time on consulting the dictionary database for the similarity. But the average analysis time by B is 298.79 ms, whereas C takes 228.92 ms, making parsing more efficient in time reduction of 23.39%. Albeit premature, we believe that these results are encouraging.

6 Concluding Remarks

Through experiments, we have confirmed that we can address spurious ambiguity in a CCG

¹²We used the statistics package of SICStus Prolog, under Sun Enterprise 250 with 512MB RAM. There were 23, 10, 10, and 10 sentences of types 1 through 4, resp.

	А	В	С
Type 1	1168	228	277
Type 2	5789	1001	1377
Type 3	1284	305	335
Type 4	-	4504	2504

Table 9: Average Parsing Times (in ms)

framework by incorporating predicate argument structures into the parsing module, and shown that co-occurrence similarity information augmented with a thesaurus helps the parser to deal with structural ambiguity. We leave open the problem of addressing those conjuncts/disjuncts that are neither semantically related nor anticipated by a corpus.

References

- R. Agarwal and L. Boggess. 1992. A simple but useful approach to conjunct identification. ACL, pp. 15-21.
- H. J. Cho. 2000. Coordinate Constructions in Korean and Parsing Issues in Combinatory Categorial Grammar. KAIST MS thesis.
- H. J. Cho and J. C. Park. 2000. Combinatory Categorial Grammar for the Syntactic, Semantic, and Discourse Analyses of Coordination in Korean. *Journal of KISS: Software and Applications*, 27(4), pp. 448-462.
- J. Eisner. 1996. Efficient normal-form parsing for Combinatory Categorial Grammar. ACL. pp. 79-86.
- L. Karttunnen. 1989. Radical Lexicalism. In Alternative Conceptions of Phrase Structure. U of Chicago Press.
- N. Komagata. 1999. Information Structure in Texts: A Computational Analysis of Contextual Appropriateness in English and Japanese. U of Pennsylvania PhD thesis.
- S. Kurohashi and M. Nagao. 1994. A syntactic analysis method of long Japanese sentences based on detection of conjunctive structures. *CL* 20(4), pp. 507-534.
- K. J. Lee. 1998. Stochastic Syntactic Analysis of Korean with Linguistic Properties. KAIST PhD thesis.
- A. Okumura and K. Muraki. 1994. Symmetric pattern matching analysis for English coordinate structures. *ANLP*, pp. 41-46.
- J. S. Park. 1998. Analysis of Coordinate Constructions in Korean with Part-of-Speech Patterns. KAIST MS thesis.
- S. Prevost. 1995. A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation. U of Pennsylvania PhD thesis.
- M. Steedman. 1990. Gapping as Constituent Coordination. Linguistics and Philosophy 13, pp. 207-263.
- M. Steedman. 2000. The Syntactic Process. MIT Press.
- J. H. Yang. 1995. Structural Ambiguity Resolution in Korean Noun Phrase Conjunctions Using Co-occurrence Similarity. *Journal of KISS (B)*, 23(3), pp. 311-321.

¹⁰The average sentence lengths of types 1 through 4 are 12.4, 16.7, 13.3, and 20.5 morphemes, respectively.

 $^{^{11}}$ A was not able to produce results at all for any of the sentences of type 4, and failed on half of the sentences of type 2 due to insufficient memory. The figures in the table reflect only the successful ones.