Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank

Yusuke Miyao
University of Tokyo

Takashi Ninomiya
CREST, JST
University of Tokyo

Jun'ichi Tsujii
CREST, JST
University of Tokyo

Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033
{yusuke,ninomi,tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper describes a method of semi-automatically acquiring an English HPSG grammar from the Penn Treebank. First, heuristic rules are employed to annotate the treebank with *partially-specified derivation trees*. Lexical entries are automatically extracted from the annotated corpus by inversely applying schemata to partially-specified derivation trees.

1 Methodology

To date, manual writing has been the only way to develop grammars based on linguistic theories. Linguistics explains language phenomena as a symbolic system of metaphysical linguistic entities such as syntactic categories. Hence, grammar development has had to rely on the linguistic intuition of grammar writers to explicate a system of unobservable linguistic entities. However, manual writing is inherently impractical as a means of developing and maintaining a robust grammar. A large number of grammar rules or lexical entries require complicated implementations, and grammar writers face difficulties in maintaining the consistency of detailed constraints. Although a few studies could apply a hand-crafted grammar to a real-world corpus (Riezler et al., 2002), these required considerable human effort that lasted for over a decade.

The new strategy outlined here is *corpus-oriented grammar development*, where a linguistics-based grammar is automatically acquired from an annotated corpus. Since the formulation of a grammar includes unobservable linguistic entities, we first *externalize* our linguistic intuition as *annotations* to a corpus. If unobservable linguistic entities were explicated as annotations, a system of linguistic entities, i.e., a grammar, would automatically be induced conforming to a linguistic theory that would explain the given annotations.

This idea is articulated within the context of *lexicalized grammar formalism*, including Lexicalized Tree Adjoining Grammar (LTAG) (Schabes et al., 1988), Combinatory Categorial Grammar (CCG) (Steedman, 2000), and Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994). Lexicalized grammars are formulated with a small number of grammar rules and a large lexicon. Hence, grammar rules can be manually written, while a lexicon should be automatically acquired.

To enable the acquisition of a lexicon in this situation, what must be externalized as annotations? Our previous work (Miyao et al., 2003a) suggests a solution: given grammar rules, lexical entries can be determined if each sentence is annotated with i) a history of rule applications, and ii) additional annotations to make the grammar rules be *pseudo-injective*. Lexical entries are then extracted by reverse-engineering the given annotations with the inverse application of grammar rules. In the acquisition of HPSG, these annotations are defined as *partially-specified derivation trees* of HPSG, which will be described in Section 3. Heuristics-based annotation will provide partially-specified derivation trees at low cost.

The inverse application of HPSG schemata will integrate partially-specified constraints given as annotations and induce lexical entries.

Compared to manual development, our approach has the following advantages.

Inexpensive The dominant cost in our approach is in maintaining annotation rules. Any heuristic rule and statistical method can be exploited, and a grammar writer is not hampered by having to maintain the consistency of the grammar. Development costs are therefore expected to be comparable to those for shallow analyzers, which utilize heuristic rules.

Wide-coverage The acquired grammar can support various constructions in real-world texts. Lexical entries will be obtained even for constructions beyond the grammar developers' prospect.

Available for machine learning An annotated corpus can be used as training data for the probabilistic modeling or machine learning of statistical parsing.

Organization of heuristic knowledge Various types of knowledge implicitly represented by heuristic rules are externalized as the annotations to a corpus. Through grammar acquisition, such knowledge is automatically organized into a grammar conforming to a linguistic theory.

Studies on the extraction of LTAG (Xia, 1999; Chen and Vijay-Shanker, 2000; Chiang, 2000) and CCG (Hockenmaier and Steedman, 2002) proposed the acquisition of lexicalized grammars from the Penn Treebank. They invented a LTAG/CCG-specific procedure to extract lexical entries from a treebank with heuristic annotations. Our study further pursues this approach, and the extraction procedure exploits the inverse application of HPSG schemata. Compared to LTAG and CCG, constraints used by HPSG are more complicated and fine-grained. Although this seems to be an obstacle to grammar acquisition, we will demonstrate that heuristic annotation and inverse schemata allow the acquisition of a lexicon.

Several methods have been proposed to automatically acquire Lexical Functional Grammars (LFG) (Bresnan, 1982) from treebanks annotated using heuristic rules (Cahill et al., 2002; Frank et al., 2003). Their aim was to automate the process to annotate c-structures with *functional schemata*, which are unification-based grammatical rules in LFG. Since the consistency of resulting schemata depends directly on the design of annotation rules, these must carefully be arranged to conform to LFG. In our approach, however, grammar rules (schemata) are given and the target of annotation is partially-specified derivation trees, which are partial results of parsing. Since annotation is separated from the design of schemata, annotation rules are not responsible for the consistency of the grammar, and these are not necessarily systematically arranged. Predefined schemata organize partially-specified constraints into a lexicon, and guarantee its consistency.

Subcategorization acquisition has extensively been studied to extract a dictionary of subcategorization frames from annotated/unannotated corpora (surveyed in (Korhonen, 2002)). The methods assumed that the classes of subcategorization frames were given, and words (in most cases, verbs) were classified into the given classes using heuristic patterns and/or corpus statistics. Our method does not require predefined subcategorization classes, and acquire lexical entries for all words in a corpus together with complete derivation structures. The method is intended to semi-automatically develop a grammar from scratch.

2 Head-driven Phrase Structure Grammar

HPSG (Pollard and Sag, 1994) is a linguistic theory based on lexicalized grammar formalism. A small number of schemata explain general grammatical constraints, while a large number of lexical entries express word-specific characteristics. Both schemata and lexical entries are represented by typed feature structures, and constraints represented by feature structures are checked with *unification* (for details, see (Pollard and Sag, 1994)).

Figure 1 provides the definition of an HPSG sign, which represents the syntactic/semantic behavior of words/phrases. HEAD feature expresses the characteristics of the head word of a constituent, such as syntactic categories. MODL, MODR, SUBJ, and COMPS represent selectional constraints of left-modifiee, right-modifiee, left-argument, and right-argument. REL and SLASH features are used to explain relative expressions

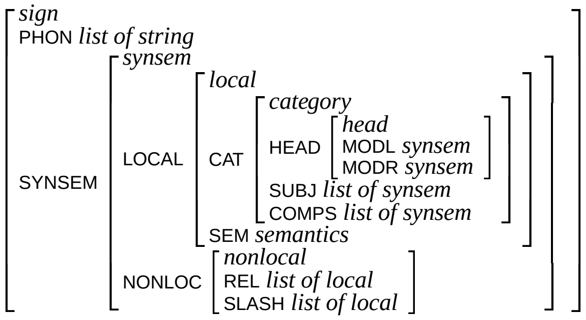

Figure 1: HPSG sign

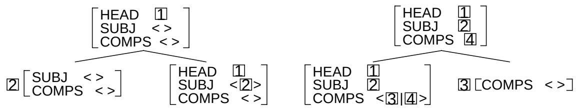

Figure 2: Subject-Head Schema (left) and Head-Complement Schema (right)

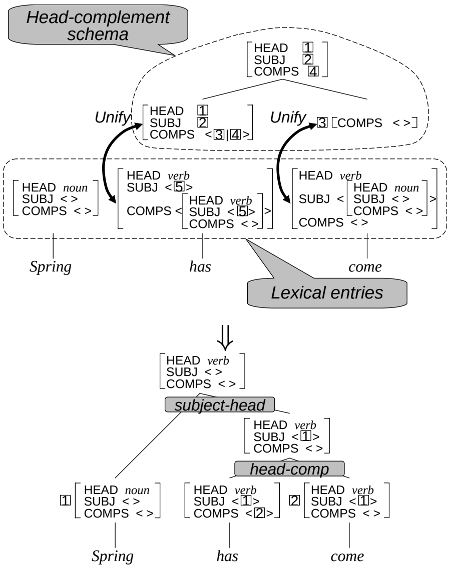

Figure 3: HPSG parsing

and unbounded dependencies. SEM feature represents the semantics of a constituent, and in this study it expresses a predicate-argument structure.

Figure 2 presents the Subject-Head Schema and the Head-Complement Schema[1] defined in (Pollard and Sag, 1994). In order to express general constraints, schemata only provide sharing of feature values, and no instantiated values.

Figure 3 has an example of HPSG parsing of the sentence "*Spring has come.*" First, each of the lexical entries for "*has*" and "*come*" are unified with a daughter feature structure of the Head-Complement Schema. Unification provides the phrasal sign of the mother. The sign of the larger constituent is obtained by repeatedly applying schemata to lexical/phrasal signs. Finally, the phrasal sign of the entire sentence is output on the top of the derivation tree.

3 Acquiring HPSG from the Penn Treebank

As discussed in Section 1, our grammar development requires each sentence to be annotated with i) a history of rule applications, and ii) additional annotations to make the grammar rules be *pseudo-injective*. In HPSG, a history of rule applications is represented by a tree annotated with schema names. Additional annotations are required because HPSG schemata are not injective, i.e., daughters' signs cannot be uniquely determined given the mother. The following annotations are at least required. First, the HEAD feature of each non-head daughter must be specified since this is not percolated to the mother sign. Second, SLASH/REL features are required as described in our previous study (Miyao et al., 2003a). Finally, the SUBJ feature of the complement daughter in the Head-Complement Schema must be specified since this schema may subcategorize an unsaturated constituent, i.e., a constituent with a nonempty SUBJ feature. When the corpus is annotated with at least these features, the lexical entries required to explain the sentence are uniquely determined. In this study, we define *partially-specified derivation trees* as tree structures annotated with schema names and HPSG signs including the specifications of the above features.

We describe the process of grammar development in terms of the four phases: *specification, externalization, extraction,* and *verification.*

3.1 Specification

General grammatical constraints are defined in this phase, and in HPSG, they are represented through the design of the sign and schemata. Figure 1 shows the definition for the typed feature structure of a sign used in this study. Some more features are defined for each syntactic category al-

[1] The value of *category* has been presented for simplicity, while the other portions of the sign have been omitted.

though they have been omitted from the figure: e.g., VFORM represents verbal forms.

Following (Pollard and Sag, 1994), this study defines the following schemata: Subject-Head, Head-Complement, Head-Modifier, Modifier-Head, and Filler-Head Schema. In addition to these, two schemata are defined as supporting constructions that often occur in the Penn Treebank. The *Head-Relative Schema* is defined for relative expressions, while HPSG explains this construction with a null relativizer. The *Filler-Insertion Schema* is defined for the construction in which an inserted clause introduces a slash, which is filled by the entire sentence. For example, in the sentence "*Mr. Kuehn, the company said, will retain the rest of the current management team,*" the complement of the inserted clause is coindexed with the entire sentence.

3.2 Externalization

This phase annotates the Penn Treebank with partially-specified derivation trees. The following annotations are sequentially added to each node in a treebank tree: head/argument/modifier marks, SUBJ features, SLASH/REL features, HPSG categories, and schema names.

First, head/argument/modifier distinctions are annotated to each node in trees using *the head percolation table* (Magerman, 1995; Collins, 1997), and trees are converted to binary trees. Since this procedure is mostly the same as in existing studies (Xia, 1999; Hockenmaier and Steedman, 2002), the details are omitted here.

After this, SUBJ features in the following constructions are specified.

Subject-control verbs Subject-control verbs such as "*try*" take VP as its complement in HPSG analysis, and its subject is shared with the unfilled subject of the VP[2]. In the Penn Treebank, complements of control verbs are represented as S with the empty subject (the top of Figure 4). Such trees are annotated with the structure-sharings as shown in the bottom of Figure 4, where the SUBJ feature of to-infinitive is coindexed with NP-1 (represented by $\boxed{1}$).

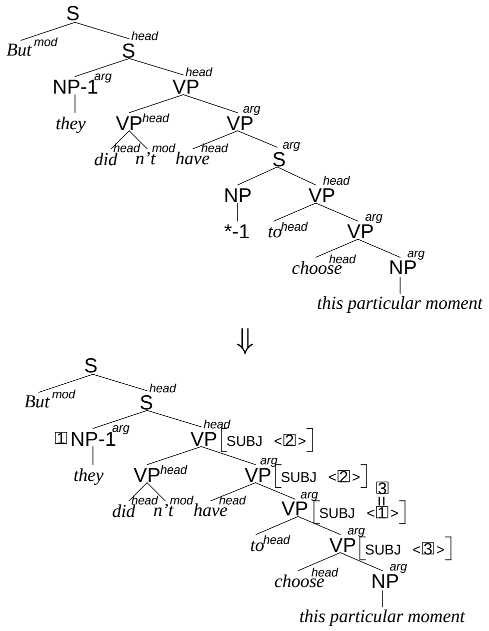

Figure 4: Annotation of subject-control and auxiliary verbs

Auxiliary verbs HPSG regards an auxiliary verb as a head that takes VP as its complement, and the subject of the auxiliary verb is shared with the unfilled subject of the complement. This relation is explicitly specified as shown in Figure 4 (represented by $\boxed{2}$). Infinitival marker "*to*" is also treated as an auxiliary verb (represented by $\boxed{3}$).

Coordinations In coordination constructions, subcategorization features are shared among conjunct phrases. In VP coordination, for example, subjects of VPs are shared and this is represented by structure-sharing of SUBJ features.

The next procedure specifies SLASH/REL features and the schema names of either the Filler-Head, Filler-Insertion, or Head-Relative Schema.

Slash & filler-head schema Since Penn Treebank-style annotation represents unbounded dependencies with trace marker '*T*', this mark is exploited to detect unbounded dependencies. The algorithm is very similar to the marking of *forward arguments* described by (Hockenmaier and Steedman, 2002). The difference is that when the filler of the slash is found, i.e., the node with the same ID number, the corresponding construction is annotated with the Filler-Head Schema (or the Filler-Insertion Schema) (Figure 5).

[2] Strictly, this analysis is for *equi* verbs such as "*seem*", although these two classes of verbs have not been distinguished in our current implementation.

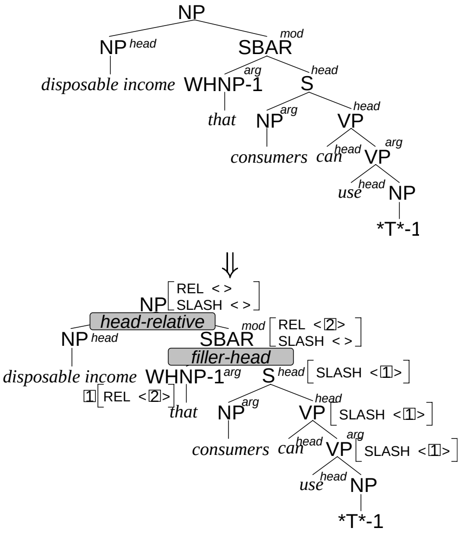

Figure 5: Annotation of slashes and relative clauses

S	HEAD *verb* SUBJ ⟨⟩ COMPS ⟨⟩	VBD	[HEAD [*verb* VFORM *finite* TENSE *past*]]
VP	HEAD *verb* SUBJ ⟨_⟩	VBG	[HEAD [*verb* VFORM *pres_part*]]
PP	HEAD *prep* SUBJ ⟨⟩	IN	[HEAD *prep*]
NP	HEAD *noun* SUBJ ⟨⟩	NNS	[HEAD [*noun* NUM *plural*]]

Figure 6: Mapping rules from Penn Treebank-style symbols into HPSG categories

Relative clauses In our implementation, a special schema is applied for relative clause constructions. When a relative clause is found, its construction is annotated with the Head-Relative Schema. In addition, a relative clause (SBAR) and its relativizer (WHNP) are annotated as having non-empty REL features (Figure 5).

Finally, each node is annotated with an HPSG category by mapping non-/pre-terminal symbols to HPSG categories. Figure 6 shows some of the mapping rules. Schema names are also assigned to all internal nodes that have not yet been assigned schema names. This is done by referring to head/argument/modifier annotations.

The above procedure annotates the treebank with partially-specified derivation trees. For example, Figure 7 shows a partially-specified derivation tree corresponding to the treebank tree in Figure 4. While the above procedure is the main part of the externalization phase, more heuristic rules were implemented to preprocess the Penn Treebank. The aim of preprocessing was to fix as many errors in the treebank as possible, and to provide a fine-grained structure to Penn Treebank-style trees with flat structures, e.g., insertion and apposition.

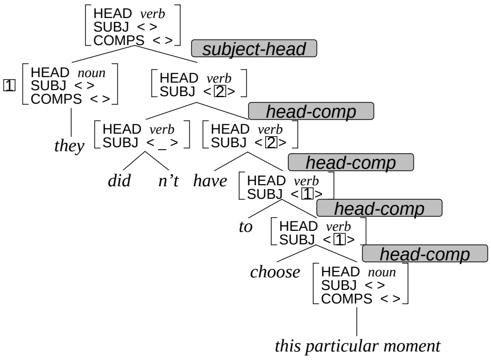

Figure 7: Partially-specified derivation tree corresponding to Figure 4

$$\begin{bmatrix} \text{HEAD } \textit{verb} \\ \text{SUBJ } \left\langle \begin{bmatrix} \text{HEAD } \textit{noun} \\ \text{SUBJ } \langle\rangle \\ \text{COMPS } \langle\rangle \end{bmatrix} \right\rangle \\ \text{COMPS } \langle\rangle \end{bmatrix}$$

Figure 8: Phrasal sign obtained by applying the Subject-Head Schema to the root of the partially-specified derivation tree in Figure 7

3.3 Extraction

In this phase, lexical entries are automatically extracted from partially-specified derivation trees given as the annotations to the treebank. *Inverse schemata* are applied to each phrasal sign in a partially-specified derivation tree. That is, given a mother as an input to a schema, daughters are computed. This procedure is considered to be the inverse of parsing described in Section 2.

For example, given the partially-specified derivation tree in Figure 7, the Subject-Head Schema is applied to the root of the tree. Then, a right daughter will be a feature structure in Figure 8. Subsequently, by applying the Head-Complement Schema to this feature structure, a left daughter is obtained as in the left of Figure 9. This will be a lexical entry for the auxiliary verb "*did*". Similarly, inverse applications of schemata will output lexical entries for all words in this sen-

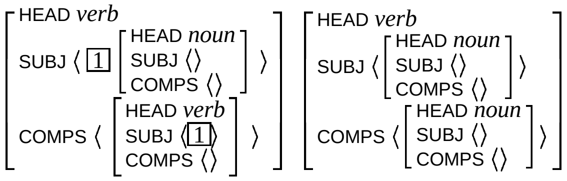

Figure 9: Lexical entries for "*did*" (left) and "*choose*" (right) extracted from the partially-specified derivation tree in Figure 7

Head annotation	58
SUBJ annotation	10
SLASH/REL annotation	16
HPSG category mapping	71
Error-fix	94
Other preprocessing rules	18

Table 1: Number of heuristic rules in annotating the Penn Treebank

tence. For example, Figure 9 has the lexical entry for "*choose*" extracted from the same sentence.

Extracted lexical entries are then generalized to *lexical entry templates*. We manually listed features to be ignored for eliminating word-specific and context-specific constraints. For example, the TENSE features in the subcategorization list (SUBJ and COMPS) can be ignored because they are irrelevant to the syntactic constraints in English. Additionally, some lexical specifications can be added to lexical entry templates. Most important is the specification of lexical semantics. In the current implementation, predicate-argument structures are constructed using heuristic pattern rules on the structure of a lexical sign.

3.4 Verification

By investigating the obtained lexicon, we can find i) defects in the grammar theory, ii) shortcomings in heuristic rules, and iii) errors in the treebank. If any shortcomings are found in heuristic rules, we return to the externalization phase to adjust them.

4 Evaluation

The algorithm described in Section 3 was implemented to acquire an HPSG grammar from the Penn Treebank Section 02-21 (39,598 sentences). Table 1 lists the number of annotation rules used in the current implementation.

	words	templates	templates per word
noun	24,947	99	1.33
verb	10,634	1,596	2.05
adjective	8,126	44	1.31
adverb	1,300	69	2.90
preposition	184	213	9.89
particle	60	12	1.48
determiner	44	30	4.84
conjunction	36	103	11.72
punctuation	15	179	27.27
total	42,669	2,345	1.70

Table 2: Number of words/lexical entry templates in the grammar acquired from Section 02-21

Lexical entries were successfully extracted from 38,263 sentences. Table 2 lists the number of words/lexical entry templates in the obtained grammar[3]. Compared to the automatic extraction of LTAG (Xia, 1999), the number of lexical entry templates was significantly reduced. This implies that the HPSG grammar achieved a higher degree of abstraction. Compared to the automatic extraction of CCG (Hockenmaier and Steedman, 2002), the number of templates increased. We assume that this was because CCG exploits grammar rules to explain syntactic variations (e.g. wh-extraction and relative clauses), while HPSG uses lexical entries. Hence, an HPSG grammar should have more lexical entries corresponding to various syntactic variations. This is substantiated by the results in Table 2, where the number of lexical entry templates for verbs is significantly higher than for the other parts of speech.

Table 3 shows lexical/sentential coverage against Section 23. Coverage was measured by comparing the acquired lexicon to lexical entries extracted from Section 23. In the table, G denotes the original grammar, and $\bar{G}$ a grammar modified to treat unknown words with a method similar to (Hockenmaier and Steedman, 2002); words occurring less than 10 times in Section 02-21 were treated equally as unknown words. A suffix denotes the threshold of the frequency of lexical entry templates; a grammar includes a lex-

[3] The summation of the number of words is not equal to the total number because a word might be assigned more than one part of speech and be double-counted.

	seen	unseen				sentential
	⟨sw,sc⟩	⟨sw,sc⟩	⟨sw,uc⟩	⟨uw,sc⟩	⟨uw,uc⟩	
G_0	94.99%	2.21%	0.10%	2.70%	0.00%	43.0%
$\bar{G}_0$	98.48%	1.41%	0.10%	0.01%	0.00%	75.9%
$\bar{G}_1$	98.46%	1.44%	0.10%	0.01%	0.00%	75.6%
$\bar{G}_5$	98.38%	1.52%	0.10%	0.01%	0.00%	74.7%
$\bar{G}_{10}$	98.25%	1.64%	0.10%	0.01%	0.00%	73.3%

Table 3: Lexical/sentential coverage against Section 23

	success	failure	error	time (sec.)
G_0	51.9%	39.2%	8.9%	4.21
$\bar{G}_0$	85.4%	1.2%	13.4%	5.47
$\bar{G}_1$	88.9%	1.2%	9.9%	4.42
$\bar{G}_5$	93.1%	1.3%	5.6%	6.03
$\bar{G}_{10}$	96.1%	1.6%	2.3%	5.25

Table 4: Results of parsing experiments

ical entry template only if it occurred more than the threshold. The "seen" and "unseen" columns represent the lexical coverage, which is the same measure as (Xia, 1999; Hockenmaier and Steedman, 2002). The "seen" column has the ratio of the word/template pairs covered by the grammar. The results are comparable to the existing studies, despite the fine-grained constraints of HPSG. The "unseen" columns have the ratio of pairs not covered by the grammar, where "sw"/"uw" mean seen/unseen words, and "sc"/"uc" mean seen/unseen templates. In most cases, both word and template were in the grammar, but they were not related. This could have been improved by a more sophisticated method of treating unknown words. The "sentential" column indicates the sentential coverage, where a sentence was judged to be covered when the grammar included correct lexical entries for all words in the sentence. This measure can be considered to be the "ideal" accuracy attained by the grammar, i.e., sentential accuracy when a parser and a disambiguation model worked perfectly.

To evaluate the robustness of our grammar in a real parsing task, we conducted parsing experiments with an HPSG parser with CFG filtering (Torisawa et al., 2000). The parser did an exhaustive search, and did not apply any heuristic techniques, such as beam-thresholding, to reduce the search space (Riezler et al., 2002) because the effectiveness of such techniques greatly depends on the characteristics of a disambiguation model. Without such techniques, however, predicate-argument structures (SEM features) cause an exponential explosion in the search space. The SEM feature was thus ignored in the parsing experiments. Another literature (Miyao et al., 2003b) described a technique to reduce the search space by beam-thresholding and reported the accuracy of predicate-argument relations attained with an automatically acquired HPSG grammar.

Shortcomings of annotation rules	
Constructions currently unsupported	16
Preprocessing failures	3
Annotation failures	1
Errors in the Penn Treebank	
Tree structure errors	6
Nonterminal errors	4
Preterminal errors	1
Constructions unsupported by HPSG	
Argument clusters	13
Head extraction	1

Table 5: Reasons for the failures of grammar acquisition

Table 4 lists the results of parsing POS-tagged sentences in Section 23 containing less than or equal to 40 words (2,287 sentences). The "success" column lists the ratio of successful parsing, i.e., at least one parse was output (not necessarily including the correct answer). The "failure" column represents the ratio of failures i.e., no parses were output. The "error" indicates the ratio of sentences that exceeded the space limit (40,000 edges). The "time" shows the average parsing time for success/failure sentences. The results attest to the significant robustness of the grammar against real-world texts.

Grammar acquisition failed for 1,335 sentences, and the reasons for these failures were investigated for the sentences in Section 02 (45 failures). The results listed in Table 5 reveal that dominant reasons were the shortcomings in annotation rules and errors in the treebank. We intend to reduce both of these by enhancing annotation rules, which should lead to further improvements in the grammar. There were relatively fewer defects in the grammar theory than expected. The results indicate that the fragility of deep processing was not inherent to linguistic theory.

5 Concluding Remarks

The principal idea proposed here was to externalize linguistic intuition as annotations to a corpus, and a large lexicon was automatically extracted from the annotations by inversely applying grammar rules to the given annotations. This approach was applied to the acquisition of a robust HPSG grammar from the Penn Treebank, which was successfully obtained at low cost.

Our claim is that the fragility of deep linguistic analysis is the result of difficulties with the development of a robust grammar based on linguistics, as opposed to most researchers who believe in the inherent impossibility of deep analysis of real-world texts. This study enabled us to develop and maintain a robust grammar based on linguistics at low cost, and opened up the possibility of robust deep analysis of real-world texts.

References

Joan Bresnan, editor. 1982. *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA.

Aoife Cahill, Mairead McCarthy, Josef van Genabith, and Andy Way. 2002. Parsing with PCFGs and automatic f-structure annotation. In *Proceedings of 7th International Lexical-Functional Grammar Conference*.

John Chen and K. Vijay-Shanker. 2000. Automated extraction of TAGs from the Penn Treebank. In *Proceedings of 6th IWPT*.

David Chiang. 2000. Statistical parsing with an automatically-extracted tree adjoining grammar. In *Proceedings of 38th ACL*, pages 456–463.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of 35th ACL*, pages 16–23.

Anette Frank, Louisa Sadler, Josef van Genabith, and Andy Way. 2003. From treebank resources to LFG f-structures: Automatic f-structure annotation of treebank trees and CFGs extracted from treebanks. In Anne Abeille, editor, *Building and Using Syntactically Annotated Corpora*, pages 367–389. Kluwer Academic Publishers.

Julia Hockenmaier and Mark Steedman. 2002. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of 3rd LREC*.

Anna Korhonen. 2002. *Subcategorization Acquisition*. Ph.D. thesis, Computer Laboratory, University of Cambridge. Published as Techical Report UCAM-CL-TR-530.

David M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proceedings of 33rd ACL*, pages 276–283.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. 2003a. Lexicalized grammar acquisition. In *Proceedings of 10th EACL Companion Volume*, pages 127–130.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. 2003b. Probabilistic modeling of argument structures including non-local dependencies. In *Proceedings of RANLP 2003*, pages 285–291.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of 40th ACL*.

Yves Schabes, Anne Abeillé, and Aravind K. Joshi. 1988. Parsing strategies with 'lexicalized' grammars: Application to tree adjoining grammars. In *Proceedings of 12th COLING*, pages 578–583.

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press.

Kentaro Torisawa, Kenji Nishida, Yusuke Miyao, and Jun'ichi Tsujii. 2000. An HPSG parser with CFG filtering. *Natural Language Engineering Special Issue – Efficient Processing with HPSG: Methods, Systems, Evaluation*, 6(1):63–80.

Fei Xia. 1999. Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of 5th NLPRS*.