PROCESSING COMPLEX NOUN PHRASES IN A NATURAL LANGUAGE INTERFACE TO A STATISTICAL DATABASE

Fred POPOWICH, Paul MCFETRIDGE, Dan FASS, Gary HALL

School of Computing Science / Centre for Systems Science
Simon Fraser University, Burnaby, B.C., Canada V5A 1S6

Abstract

Analysis of a corpus of queries to a statistical database has shown considerable variation in the location and order of modifiers in complex noun phrases. Nevertheless, restrictions can be defined on nominal modification because of certain correspondences between nominal modifiers and the role they fulfill in a statistical database, notably that the names of database tables and columns, and values of columns, are all determined by the modifiers. These restrictions are described. Incorporating these restrictions into Head-Driven Phrase Structure Grammar (HPSG) has caused us to examine the treatment of nominal modification in HPSG. A new treatment is proposed and an implementation within an HPSG based natural language front-end to a statistical database is described.

1 Introduction

A prototype natural language front-end to statistical databases is being developed as part of an Executive Information System for Rogers Cablesystems, a Canadian cable television company. The initial target database is the Rogers Technical Operations Database, a relational database containing statistical data describing aspects of the company's business related to customer service.

The front-end employs an HPSG chart parser. There are numerous variations of HPSG; we have chosen [PS87] since it is the most familiar and widely published. Our results can be extended to other variations. In the spirit of HPSG, we have avoided a proliferation of grammar rules and kept them highly schematic.

In developing the grammar for the queries in our corpus, we encountered a selection of interesting noun phrase constructions which caused us to examine the treatment of adjunct modification of nominals within HPSG. This has resulted in a proposal which should be of interest to other researchers developing natural language interfaces.

2 Complex NPs in Queries

We began the project by collecting a corpus of 68 English language queries from three senior executives at Rogers. Our corpus contains constructions paradigmatic of a wide selection of natural language queries that the executives would like to pose to their database. A selection of these queries are shown in (1–6).

(1) *Give me the western region outage log summary.*

(2) *Give me the system reliability performance.*

(3) *Compare the basic service problem statistics per thousand customers.*

(4) *Compare the terminal equipment problems.*

The sentences contain complex NP constructions and there is a large amount of variation with respect to the location and ordering of the modifiers. For example, most pre-nominal modifiers may also appear as post-nominal modifiers.

(5) **Vancouver** *system reliability performance*

(6) *system reliability performance* **for Vancouver**

Prepositional phrases like *for Vancouver* can be viewed as an abbreviated form of the prepositional phrase *for the Vancouver division.*

The NPs within these sentences contain a great deal of syntactic ambiguity. Consider the complex NP in (1). The adjective *western* can either modify *region* or *outage* or *log* or *summary*. Similarly, *region* could modify any of the nominals appearing to its right. However, much of this syntactic ambiguity does not have a semantic interpretation in the database semantics. For example, (1) has only a single interpretation although there are numerous syntactic analyses.

We have gone into detail about the corpus to show the rich structure of noun phrases and to motivate the reasons for the design choices in our semantics and grammar.

3 Complex NPs in HPSG

3.1 Overview of HPSG

HPSG is one of the best known unification-based grammar formalisms. It employs attribute value matrices (called signs) to represent lexical entries, grammar rules and principles. HPSG borrows freely from other formalisms. For example, the treatment of syntactic categories, syntactic features, and some of the principles are from generalized phrase structure grammar (GPSG) [GKPS85]. The main syntactic categories in HPSG are heads (the head constituents of phrases), adjuncts (traditionally called modifiers) and complements (traditionally called arguments). The principles of HPSG include the Constituent Order Principle, Subcategorization Principle, Head Feature Principle, and Semantics Principle.

HPSG contains three grammar rules for combining heads with complements.

(7) $[\text{SUBCAT}\,\langle[\;]\rangle] \rightarrow H[\text{LEX}\,+,\ \text{INV}\,-],\ C^*$

(8) $[\text{SUBCAT}\,\langle\;\rangle] \rightarrow H[\text{LEX}\,-],\ C$

(9) $[\text{SUBCAT}\,\langle\;\rangle] \rightarrow H[\text{LEX}\,+,\ \text{INV}\,+],\ C^*$

One rule (7) combines a lexical head with everything but its final complement. This rule can also be used to convert a lexical head requiring only a single complement into a non-lexical constituent still requiring a single complement. Another rule (8) combines a non-lexical head with its final complements. Yet another rule (9) works for inverted constructions: those involving a lexical head that is marked for inversion.

As in GPSG, generalizations about the relative order of sister constituents is factored out of the phrase structure rules and expressed in independent linear precedence (LP) constraints. The LP constraints are used by the Constituent Order Principle. HPSG rules are immediate dominance (ID) rules. Consequently, a single ID rule of the form $X \rightarrow H\,A$ could describe a head constituent H either preceded or followed by an adjunct A — the relative ordering of H and A is determined by the LP constraints.

3.2 Issues in the Treatment of Adjuncts

Nominal modification is treated in HPSG by having heads that contain a set valued feature called ADJUNCTS [PS87]. Each element of this set is a sign which describes a potential adjunct. For instance, the ADJUNCTS feature for a noun will contain an entry for adjectives, one for nouns, one for prepositional phrases and one for verb phrases.

An alternative, which was also discussed in [PS87] and has been adopted in other grammar formalisms (e.g., [Usz86, CKZ88]) and some variations of HPSG [Coo90, Pol91], is to allow adjuncts to select their heads.[1] The head feature called HEADS contains a set of descriptions, one for each construction that can be modified by the adjunct. For example, the HEADS feature for an adjective will contain a sign for a noun.

In our corpus, a head has more possible classes of modifiers than modifiers have classes of possible heads. For example, the set of modifiers for NPs and $\overline{\text{N}}$s (i.e., NPs lacking determiners) includes adjectives, nominals, PPs and even VPs (relative clauses). In §3.4 we shall see that each of these modifiers can have only one or two possible heads. Furthermore, the task of reducing the size of the HEADS or ADJUNCTS set, by discovering common semantic features for which a constituent can select, meets with greater success if modifiers select their heads. That is, one is more likely to find commonality among the constituents which an adjunct can modify than among the modifiers which a head can take. Selections of heads by adjuncts permits a greater range of subcategorization to be specified through default inheritance rather than explicit specification.

Some aspects of adjunct semantics are impossible if adjuncts are selected by heads rather than heads selected by adjuncts. Predicates, both adjectives and verbs, have argument structure which coerces their arguments into thematic roles. For example, the adjective *modern* imposes on its argument the thematic role of *Theme*.[2] It is not obvious how the nominal argument of the adjective receives its thematic role unless it is the adjective which selects the nominal, parallel to the assignment of thematic roles by verbs to their NP arguments. If *modern* selects its head, then the thematic role of the head may be specified in the HEADS

[1] Cooper [Coo90, Ch.3, §6] looks in some detail at the arguments in favour of adjuncts selecting their heads.

[2] In [Pol91, §1.3], Pollard and Sag introduce semantic features like AGENT, GOAL and THEME within the feature structure containing the semantic CONTENT.

attribute and inherited by the head when it unifies with the HEADS attribute. If instead, heads subcategorize for their adjuncts, this information must be inherited in some other fashion, perhaps through structure sharing from the adjuncts list.

The problem and its solution are evident when derivational morphology are considered. The verb *read* imposes the thematic role of *Agent* (Ag) on its subject and the thematic role of *Theme* (Th) on its object. When this verb is coerced into an adjective by the derivational suffix *-able*, the resulting adjective assigns the thematic role of *Theme* to its argument. If adjectives select their heads, then the derivational rule is evident.

$$(10)\ V[\text{SUBCAT}\ \langle \text{NP}_{\text{Th}}, \text{NP}_{\text{Ag}} \rangle] \Longrightarrow \text{Adj} + \text{``able''}[\text{HEADS}\ \{\text{N}_{\text{Th}}\}]$$

Given that adjuncts will select their heads, a grammar rule for adjuncts can be stated most concisely if we combine a head with a single adjunct at a time. Thus, our constituent structures will contain an ADJUNCT-DTR feature which will take the adjunct as its value, rather than a list-valued ADJUNCT-DTRS feature which would take a list of adjuncts as its value. A head that is modified by more than one adjunct will require more than one application of the grammar rule.

One disadvantage of this approach is that a complex nominal like *system reliability for Vancouver* will have two analyses: one where the PP *for Vancouver* modifies the head noun *reliability* and another where it modifies the head nominal *system reliability*. If the adjuncts rule combined a head with all of its adjuncts at the same time, there would be only one analysis. However, one could argue that there should be two interpretations for the phrase and that both should be reflected in the grammar. Pollard and Sag note that "there is evidence that noun-noun and adjective-noun structures share some syntactic properties with lexical nouns as opposed to typical common noun phrases, e.g. they can occur themselves as modifiers in noun-noun structures" [PS87, p.73]. They propose analyzing noun-noun and adjective-noun constructions as [LEX +] even though they have internal structure. By adopting this treatment of complex noun phrases, we can prevent analyses for ungrammatical constructions like *system for Vancouver reliability*, plus we can prevent ambiguity in the analysis of phrases like *system reliability for Vancouver*. In our grammar we introduce two rules for adjuncts, which are designed to give wide coverage and to avoid spurious ambiguities.

3.3 Two Rules for Adjuncts

One adjunct grammar rule is required for combining saturated lexical adjuncts with their heads. That is, for lexical adjuncts which have empty subcategorization lists, like adjectives, proper nouns (specifically, the proper nouns corresponding to months and cities) and adverbs. The rule will be restricted so that it will apply to phrases with unsaturated heads. Heads that fall into this category are $\overline{\text{N}}$s, PPs,[3] VPs, and APs. The specific pairing of adjuncts to heads is determined by the HEADS feature of the adjunct (§3.4). Additionally, if the head modified by the adjunct is marked [LEX +] then the resulting constituent will also be [LEX +], thus implementing the analysis of adj-noun and noun-noun constructions discussed in the previous section. Using the schematic notation for grammar rules introduced in [PS87], we can present the rule as shown in (11).

$$(11)\ [\text{SUBCAT}\ \langle [\,] \rangle, \text{LEX}\ \boxed{1}] \rightarrow H[\text{LEX}\ \boxed{1}],\ A[\text{SUBCAT}\ \langle \rangle, \text{LEX}\ +, \text{HEADS}\ \{...H...\}]$$

Note that the two appearances of $\boxed{1}$ in (11) indicate that the head and the resulting constituent share the same value for their LEX features. The Subcategorization Principle will ensure that the head and the resulting constituent will have the same value for their SUBCAT features. Since the grammar rule is an ID rule, it does not place any restriction on the linear ordering of the head (H) and adjunct (A). This rule is designed so that it applies before a head is combined with its final complement (8). It can be viewed as the HPSG counterpart to the adjunct rule from X-bar theory [Cho82] shown below, where the *ADJUNCT* is required to be lexical and not subcategorize for any arguments.

$$(12)\ \overline{\text{X}} \rightarrow \overline{\text{X}}\ \text{ADJUNCT}$$

In order for heads to be modified by unsaturated adjuncts, we propose a second grammar rule.

$$(13)\ [\text{SUBCAT}\ \langle [\,] \rangle, \text{LEX}\ \boxed{1}] \rightarrow H[\text{LEX}\ \boxed{1}],\ A[\text{SUBCAT}\ \langle [\,] \rangle, \text{LEX}\ \boxed{1}, \text{HEADS}\ \{...H...\}]$$

[3] Like [PS87, p.70], we propose that prepositions have two elements on their subcategorization list, the first being the prepositional object and the second its subject. A PP is obtained by combining a preposition with its object NP. We do not propose lexical entries for prepositions having only the object NP on its SUBCAT list since this would complicate the LP rules (§3.5) and grammar rules (7) and (8).

Rule (13) requires the adjunct to have a single element in its SUBCAT list, thus allowing PP, VP and $\bar{N}$ modifiers to modify PPs, VPs and $\bar{N}$s. Of course, the contents of the HEADS feature will restrict the applicability of this rule (§3.4). Unlike rule (11) which allowed a lexical adjunct to modify either a lexical or non-lexical head, rule (13) requires the head, adjunct and resulting constituent to possess the same values for their LEX features, as reflected by the coindexing with $\boxed{1}$. With this rule, a "lexical" compound noun can modify a lexical noun to yield a "lexical" compound noun (e.g., $N \rightarrow N, N$), or a (non-lexical) PP can modify a non-lexical nominal to yield a non-lexical nominal ($\bar{N} \rightarrow \bar{N}, PP$).

Direct consequences of our two adjuncts rules are that prepositions and verbs are not allowed to modify anything (these have two or more elements in their SUBCAT lists), sentences or complex noun phrases cannot appear as adjuncts, and NPs, Ss, adjectives, verbs and prepositions cannot be modified by anything. Our grammar does not prevent nouns from being modified, since rule (7) can be applied to a lexical noun to yield a non-lexical nominal (essentially, $\bar{N} \rightarrow N$). If we allowed full NPs or Ss to be modified, the result would be a syntactic ambiguity which would not have any semantic relevance.

3.4 The HEADS Feature

The applicability of the two adjuncts grammar rules is restricted by the value of the HEADS feature of the adjunct. For prepositions (lexical entries with SYN|LOC|HEAD|MAJ = P), the value of the HEADS feature will be a set containing a sign for $\bar{N}$ constituents (N[SUBCAT $\langle [\,] \rangle$, LEX $-$]) and a sign for VP constituents.[4] Lexical entries for nouns and adjectives will have a single element in their HEADS set. It will contain a sign for lexical nouns, which includes compound nouns (N[SUBCAT $\langle [\,] \rangle$, LEX $+$]). We are proposing that pre-nominal modifiers, like adjectives and (compound) nouns, will be combined with their head nouns before post-nominal modifiers, like PPs. We adopted this decision because applying modifiers in different orders does not result in any difference in the resulting semantic interpretation. Specifically, the semantic representation associated with *[the [[system reliability] for Vancouver]]* is the same as that for *[[the [system reliability]] for Vancouver]* and *[the [system [reliability for Vancouver]]]*. With our proposal, we obtain only one analysis for the phrase discussed above. Finally, in order to allow relative clauses (MAJ=V), we need only propose that they contain a sign for $\bar{N}$ in their HEADS set. Thus, we effectively treat relative clauses like *restrictive* relative clauses. As was the case with PP adjuncts, the same semantic representation is obtained regardless of whether the relative clause modifies an $\bar{N}$ (restrictive relative) or an NP (non-restrictive relative).

3.5 Linear Precedence

We adopt the same LP constraints for heads and complement daughters as proposed in [PS87]. Lexical heads are required to precede their complement(s), while non-lexical heads follow their complement(s). Sister complements appear in the reverse order of their appearance in the SUBCAT list of their head. The LP constraints for adjuncts require signs with MAJ=A or MAJ=N (+N categories in terms of the classification present in [Cho82]) to precede their heads, while adjuncts with MAJ=V or MAJ=P (−N categories) are required to follow their heads. Thus adjectives and nominal modifiers will precede the nouns they modify, while PPs and relative clauses will follow the constituents they modify.

3.6 Semantics

Due to the close relationship between syntax and semantics in HPSG, we can avoid syntactic ambiguities which do not correspond to distinct semantic analyses. Semantic information, consisting of TYPE and content (CONT), can be used to prevent certain analyses. The TYPE of a complex constituent will be the same as that of its head. The Semantics Principle is responsible for creating the CONT of a complex constituent from that of its daughters (subconstituents) [PS87]. We adopt a version of this principle for building up semantic information for database structures, which we call the Database (DB) Semantics Principle [McF91].

We incorporate selectional restrictions based on a semantic type hierarchy which incorporates aspects of the database design. The Rogers Technical Operations Database is a statistical database; that is, each table in the database contains one or more category attributes (columns) whose values define sets of entities of a single type, and one or more statistic attributes (columns) whose values summarize these sets. The

[4] In our corpus PPs do not appear to modify any VPs, so we can actually simplify the HEADS feature so that it contains only the $\bar{N}$ entry.

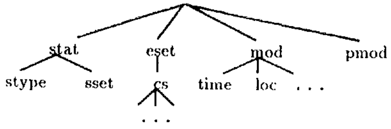

Figure 1: Semantic Type Hierarchy

complex noun phrases used in natural language queries to this database consist of nominals, or nominal modifiers which belong to five general classes: statistical type (stype), statistical set (sset), entity set (eset), modifier (mod) and pre-modifier (pmod). Each of these classes may be divided into subclasses using information from the conceptual database design. These five classes are arranged in a semantic type hierarchy as shown in Figure 1. Using this hierarchy, we can incorporate selectional restrictions into the HEADS feature of modifiers. Nouns like *summary, sum,* and *ratio* are used to refer to particular (sets of) statistics. Members of the sset class (e.g., *log, performance, activity*) may be used to modify stypes. Nouns from the sset class may be semantically vacuous, that is, we assume that all requests are for *some* set of statistics and these nouns may not carry any information that can help identify the particular statistics sought by a user. We allow (compound) nouns within the eset class (e.g., *problem, outage, call, reliability*) to modify (compound) nouns of type **stat** (i.e., sset or stype). Adjuncts of type **mod** may modify subclasses of eset. For example, a user can request either *system reliability statistics* or *service calls.* The type **pmod** may modify other modifiers and selected types of eset.

The selectional restrictions distilled from our type hierarchy are by themselves not powerful enough to eliminate all of the "spurious" ambiguities. Just as we can use the TYPE feature from the semantics of the sign, we can also use the CONT to restrict possible analyses. To do this, we have modified the DB Semantics Principle with an Adjunct Contribution Constraint so that an adjunct is required to contribute semantic information to a head-adjunct constituent — in particular, adjuncts must contribute references to database constructs — hence the constraint disallows semantically vacuous adjuncts from combining with a head. A complex constituent like *outage log summary,* in which *outage* has semantic content but *log* makes no contribution of database information, would have only one analysis. The noun *log* would not be allowed to modify *summary,* but *outage* could modify *log,* and then *outage log* could modify *summary.*

Sent	Parse	Total	Edges
(1)	14 (33)	19 (43)	99 (153)
(2)	5 (6)	7 (8)	58 (65)
(3)	12 (21)	16 (27)	96 (125)
(4)	5 (5)	8 (8)	60 (60)

Table 1: Parsing Performance

4 Implementation

Our treatment of complex NPs has been incorporated into the SX natural language interface [MC90]. The SX system uses grammar developed within the HPSG-PL grammar development system [PV91a]. The semantic representations built up by an HPSG parser are directed to a module which converts them into an SQL query. The query can then be directed to an Oracle database to obtain the requested information.

SX makes use of chart parsing implementations of HPSG developed in LISP by McFetridge [MC90] and in Prolog by Popowich and Vogel [PV91b]. Chart parsing is a type of parsing in which all syntactic structures which are built are placed on a single graph structure called a chart. Nodes in the chart correspond to positions in an input sentence, with edges between the nodes describing analyses of substrings of the input. A successful parse corresponds to an edge that spans the entire input sentence. The performance of the Prolog parser on sentences (1)–(4) are summarized in Table 1. For each sentence, the table shows the time in CPU seconds for obtaining the first parse (Parse) and for searching for all possible interpretations (Total). The table also contains the number of edges created by the chart parser while searching for these interpretations. To illustrate the effect of the Adjunct Contribution Constraint discussed in §3.6, Table 1 also shows (in brackets) the number of edges and CPU times when this constraint is not used. The tests were performed on a SUN SPARCstation 1 running Quintus Prolog 3.0.

5 Discussion

Natural language interfaces to statistical databases are still rare but, with the growing interest in Executive Information Systems and increasing needs of executives to have immediate access to summary (i.e., statistical) data, the demand for such interfaces is likely to expand. To our knowledge, the only other natural language interface to a statistical database is the EasyTalk natural language interface produced by Intelligent Business Systems. EasyTalk can apparently cope with tables that contain "summary- or detail-leveled values" [Hwa89]. However, IBS has not released much information about their interface because it is a commercial product, so a comparison of the two interfaces is not possible.

Besides being one of the first interfaces to a statistical database, our front-end has other novel features: a treatment of adjuncts in HPSG that synthesizes ideas from other treatments (§3.3ff), a semantic hierarchy derived from our database and sentence corpus (§3.6), and a modification of the Semantics Principle used in HPSG (§3.6).

In future work, we plan to further investigate the processing of complex NPs, particularly conjunction and relative clause construction.

Acknowledgements

We have benefited greatly from discussions with Carl Vogel and from his comments on earlier drafts of the paper. We would also like to thank Nick Cercone and Sandi Kodrič.

The authors are members of the Institute for Robotics and Intelligent Systems (IRIS) and wish to acknowledge the financial support of the Networks of Centres of Excellence Program of the Government of Canada, the Natural Sciences and Engineering Research Council (NSERC), the participation of PRECARN Associates Inc., and a research grant from Canadian Cable Labs. Fred Popowich is supported by a Fellowship from the BC Advanced Systems Institute and by NSERC under Grant No. OGP0041910. We are grateful to the Centre for Systems Science at Simon Fraser University for the use of its facilities.

References

[Cho82] Noam Chomsky. *Lectures on Government and Binding, the Pisa Lectures, 2nd (Revised) Edition.* Foris Publications, Dordrecht, Holland, 1982.

[CKZ88] Jo Calder, Ewan Klein, and Henk Zeevat. Unification categorial grammar: A concise, extendable grammar for natural language processing. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 83–86. Budapest, Hungary, 1988.

[Coo90] Richard Cooper. *Classification-based Phrase Structure Grammar: An Extended Revised Version of HPSG*. PhD thesis, University of Edinburgh, Scotland, 1990.

[GKPS85] Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. *Generalized Phrase Structure Grammar*. Basil Blackwell, London, England, 1985.

[Hwa89] Diana Hwang. IBS to unveil EasyTalk software. *Digital News*, April 17th 1989.

[MC90] Paul McFetridge and Nick Cercone. The evolution of a natural language interface: Replacing a parser. In *Proceedings of Computational Intelligence 90*. Università di Milano, Milan, Italy, 1990.

[McF91] Paul McFetridge. Processing English database queries with head-driven phrase structure grammar. In *Proceedings of the 2nd Japan-Australia Joint Symposium on Natural Language Processing*, pages 25–31. Iizuka City, Japan, 1991.

[Pol91] Carl Pollard. *Topics in Constraint-Based Syntactic Theory.* Third European Summer School in Language, Logic and Information, Universität des Saarlandes, Saarbrücken, Germany, 1991.

[PS87] Carl Pollard and Ivan Sag. *Information-Based Syntax and Semantics, Volume 1: Fundamentals.* Centre for the Study of Language and Information, Stanford University, CA, 1987.

[PV91a] Fred Popowich and Carl Vogel. The HPSG-PL system. Technical Report CSS-IS TR 91-08, School of Computing Science, Simon Fraser University, Burnaby, B.C., 1991.

[PV91b] Fred Popowich and Carl Vogel. A logic based implementation of head-driven phrase structure grammar. In C.G. Brown and G. Koch, editors, *Natural Language Understanding and Logic Programming, III*, pages 227–246. Elsevier, North-Holland, 1991.

[Usz86] Hans Uszkoreit. Categorial unification grammars. In *Proceedings of the 11th International Conference on Computational Linguistics*, pages 187–194. Bonn University, West Germany, 1986.