Principled Multilingual Grammars for Large Corpora

Sharon Flank and **Paul Krause**
Systems Research and Applications Corporation
2000 15th Street North, Arlington, VA 22201
(703) 558-4700
flanks@sra.com, krausep@sra.com

Carol Van Ess-Dykema
Department of Defense*
9800 Savage Road
Fort Meade, MD 20755
cjvanes@afterlife.ncsc.mil

Abstract

In-depth text understanding for large-scale applications requires a broad-coverage, robust grammar. We describe a multilingual implementation of such a grammar, and its advantages over both principle-based parsing and ad-hoc grammar design. We show how X-bar theory and language-independent semantic constraints facilitate grammar development. Our implementation includes innovative handling of (1) syntactic gaps, (2) logical structure alternations, and (3) conjunctions. Each of these innovations enhances performance in both large-scale and multilingual natural language processing applications.

Phrase structure grammars are hardly new. The novelty in this paper comes from the use of practical guidelines and real numbers based on our experience with three languages and tens of thousands of texts. The issue of grammar design is worth revisiting because of the increasing bifurcation between semantic phrase grammars on the one hand, and principle-based parsing in toy domains on the other. Semantic grammars are brittle and must be rewritten for each new domain and language; principle-based parsing is not yet mature enough for our applications. We offer an extensible, multilingual application of the traditional approach that extends theoretical linguistic insights to industrial strength data.

The Problem

The development of multilingual grammars presents a dilemma. There is considerable commonality in grammars of human languages, as represented by the insights of so-called "universal grammar," (Chomsky 1967). However, the need for speedy development has led many to abandon full-sentence parsing and substitute either phrase parsing or very specific semantic patterns. This approach ignores the universality of syntax across languages, and requires that patterns be created not only for every language but also for every domain. Other researchers have embraced portability and extensibility by using principle-based parsers. Principle-based parsing attempts to use syntactic theory to specify grammatical principles and parameters so that a single core grammar can be used as the basis for multilingual processing. Principle-based parsing (Berwick 1987; Fong & Berwick 1989; Lin 1993) relies on Government and Binding Theory, as initially described in Chomsky (1981). Unfortunately, neither principle-based parsing nor Government and Binding Theory itself has been proven viable for the fifty-word sentences with multiple conjunctions, gaps, and asides that are typical of text understanding applications. Here, for example, is a not atypical Spanish sentence to be handled in a large-scale application:

> La información la proporcionó ayer el Ministro de Salud doctor Juan Giaconi, quien dio a conocer que durante el último trimestre se detectaron 14 casos de Sida en el país, lo que arroja hasta ahora una cifra total acumulada de 42 contagios declarados desde 1984, año en que se registró el ingreso del virus al territorio.

> The information was provided yesterday by the Minister of Health Doctor Juan Giaconi, who made it known that during the last quarter 14 AIDS cases were detected in the country, which to date amounts to a total accumulated sum of 42 infections reported since 1984, the year in which the entrance of the virus into the region was recorded.

Correct attachment of, for example, *the year...* to *1984*, requires that the semantic similarity of the two be evident. That is, it must be clear that 1984 is a year, requiring semantic information be available to the parser. While this is not precluded by Government and Binding theory, current implementations do not facilitate access to such information. In sum, what is needed is a fast parser that can capture commonalities, both syntactic and semantic, across languages and domains.

*The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Department of Defense or the United States Government.

Table 1: Core Grammar Rules

Category	X-Bar Levels to Be Included
Noun Phrase	Minimum of 4 X-bar levels,[a] with 3 possibilities at each level (SPEC,[b] COMP, Noun-noun and Adj-noun modification, and conjunction)
Prepositional Phrase	SPEC, COMP, and conjunction
Verb Phrase	Direct and indirect objects, verb particles, prepositional and sentential arguments, conjunction
Adjective Phrase	SPEC, COMP, and conjunction
Adverb Phrase	SPEC, COMP, and conjunction
Quantifier Phrase	SPEC, COMP, and conjunction
Sentence	Left and right complements, conjunction, punctuation
INFL/Auxiliaries	Correct ordering of all types requires several rules
Conjunction Phrase	Commas with and without "and" etc.
Complementizer/ Subordinating Conjunction Phrase	Specifiers, commas
Specifiers	Several simple and complex types, including negation
Complements	S', S, PP, appositive
Punctuation Rules	Period, semicolon, colon, question mark, etc.
Specific Rules	Latitude/longitude, age appositives, id numbers, and other special constructions

[a]Orthodox X-bar theory allows only three bar levels, but noun-noun and adjective-noun modification presents a clear intermediate level, and we handle it as if it were a bar level

[b]SPEC and COMP refer to specifier and complement in X-bar terminology. They are modifiers of the phrasal head; for details, see Jackendoff (1977). PP refers to prepositional phrases; S is a sentence, while S' is a sentence with a complementizer. INFL refers to inflection.

Solutions

Multilingual Architecture

We claim that an X-bar-based grammar framework with language-independent semantic functions provides the extensibility of principle-based parsing without sacrificing the practicality of the phrase parsers. Phrase parsers are easy to write, easy to use, and nearly theory neutral, e.g. VP -> ADVP VP.

Principle-based parsing relies on broad postulates that are difficult to implement in practical terms, e.g. "A moved element must c-command its trace, where A c-commands B if A does not dominate B, but the parent of A dominates B," (Lin 1993); "Assign indices freely to all noun phrases" (Fong 1990). Our natural language understanding architecture contains five processing modules: Preprocessing, Syntactic Analysis, Semantic Interpretation, Discourse Analysis and Pragmatic Inferencing. The modules are language-independent; only the data are language-specific. Likewise, the modules are domain-independent, since they contain no domain data.

Our multilingual system architecture (and a cognitive view of language) assumes that Semantic Interpretation is language-independent. Semantic Interpretation accepts as input the structures that are output by the grammars of the various languages. Therefore, the output of the various language grammars must be consistent, suggesting close coordination between grammar structures across languages.

Syntactic Analysis consists of a processing algorithm (the parser), and its associated data (the grammar). The parser is entirely language-independent. There is a separate grammar for each language, but, as we will demonstrate, the linguistic core of each grammar is the same. This semi-language-independent strategy provides maximum robustness while taking advantage of the insights provided by generative grammar.

Certain grammar formalisms make some kinds of development easier. Semantic grammars, e.g. SRI's FASTUS system, can be built quickly, but they are domain-dependent and fragile. Phrase parsers, like AT&T's Fidditch, will find partial parses quickly, but fail to provide detailed enough information to support reference resolution in discourse. More linguistically-based grammars, like the Lexical Functional Grammar as implemented at Carnegie Mellon University in Diogenes, or principle-based parsing as used at MIT, or in DBG at LSI, have not yet demonstrated that they are capable of scaling up to handle real applications. Our approach attempts to reap the advantages of linguistic generalizations while not sacrificing practicality. In particular, it captures language-independent features, and it allows for semantic constraints.

The Grammar

Phrase Structure Rules for a Core Grammar. We use a phrase structure grammar. Based on the principles of X-bar theory, we claim that a broad-coverage grammar should contain approximately 200 phrase structure rules. Our English grammar has 231 rules, while our Spanish and Japanese grammars, somewhat less complete, contain 161 and 176 rules,

Table 2: X-bar-based rules illustrated in several languages.

Rule	English	Spanish	Japanese
N" -▷ N" COMP	Judith Jones, director	María Velazquez, la directora	ディレクター、田中一男氏
N' -▷ ADJP N'	the green house	la casa verde	青い家
N' -▷ ADJP	the brave	los valientes	不自由
N' -▷ QP	2000 (of them)	2000 (de ellos)	(そのうち)2000人
N' -▷ N N'	health sector	sector salud	衛生部門
PP -▷ PP CONJP PP	between 3 and 5	entre 3 y 5	3と5の間で
ADVP -▷ SPEC ADV"	more slowly	más lentamente	さらにゆっくり
QP -▷ SPEC Q"	almost 100	casi 100	ほとんど100
	0.7 per 1000	0,7 por 1000	1000につき7
SPECP -▷ NEG SPEC"	not only him	no sólo él	彼だけでなく
SPECP -▷ SPEC"	all those;	todos ellos;	それらすべて(の)
	more costly	más costoso	もっと高価(な)
COMP -▷ NP	the multinational, Dow Corning	la multinacional Dow Corning	国際派ダウ・コーニング
COMP -▷ NP	Vicky Wong, 555-1212	Pedro Ramos, 555-1212	中村太郎、555-1212
COMP -▷ NP	Meg Smith, 37	Juanita Lopez, 37	鈴木花子、37歳
COMP -▷ S	(she wants) to eat	(quiere) comer	英語を話す(ことができる)

respectively. Basic coverage demands a minimum of about 120 rules, although punctuation irregularities and domain idiosyncracies may require more. A grammar with too few rules has insufficient coverage, while a grammar with too many rules (e.g. NP -▷ NP PP PP) overgenerates. Table 1 identifies the rules of the core grammar. Each category appears in the grammar with the elements identified in the second column.

Table 2 illustrates the application of core grammar rules to several languages. The word order may change, as in the English and Spanish adjective rules:

N' -▷ ADJP N' (English) N' -▷ N' ADJP (Spanish)

Nonetheless, the underlying X-bar structure remains constant. In practical terms, this allows the skeleton of the "universal" grammar to be used as scaffolding upon which to build grammars for new languages.

Table 3 lists the types of rules required for each phrasal category, including the X-bar elements *head, specifier,* and *complement,* as well as conjunction, modifiers, punctuation, and special idiomatic rules. We include English, Spanish, and Japanese, and a conjecture regarding universal grammar.

To determine the aggregate number of possible rules, assume two possibilities for each + mark. That is, each category can occur at a given level *with* a SPEC and *without* a SPEC, with and without a COMP, conjoined or not conjoined, etc. (except head, which is required). Add the numbers of + marks (times two) to reach the figures in the right-hand column.

Our phrase structure rules incorporate the insights of X-bar theory, as originated by Jackendoff (1977) and in use currently (in varying interpretations) by a wide range of linguists, including, in recent work on phrase structure, Speas (1990) and Rothstein (1991). X-bar theory imposes certain restrictions on the content of phrase structure rules. Only maximal projections can appear in rules in a non-head position:

XP -▷ YP X" is acceptable
XP -▷ Y' X" is not, because Y' is not a maximal projection

Rules that have three or four elements on the right, e.g.

NP -▷ N" ADJP PP PP

are badly designed rules. They should be replaced by more compositional rules that take advantage of the cross-categorial nature of specifiers and complements, instead of spelling out the specifiers and complements for each level in each category. Thus rules like

NP -▷ N" COMP

are preferred.

Augmenting the Phrase Structure Rules: Constraints. Although phrase structure rules provide an easy preliminary approach to grammar development, they are inadequate for accurate parsing in large-scale systems, since they overgenerate, as we will demonstrate below. We supplement the parsing algorithm with augmentations, similar to those used for ATN (Augmented Transition Network) grammars (Woods 1973). These augmentations are attached to the phrase structure rules in the grammar. They contain specialized functions for constraints and structure-building. The functions themselves are language-independent, although they are used in language-specific grammar rules. When a rule is reduced during parsing, its associated augmentations are evaluated.

The augmentations in the grammar contain constraints that halt the application of the rule under certain circumstances. These constraints can be used

Table 3: Multilingual rule matrix

Category	head	spec	comp	conj	mod	punc	idiom	Total
Noun Phrase	+++ **r**	+++ **r**	+++ **r**	+++ **r**	+++ +	+++ +	+++ +	13
Prepositional Phrase	+++ **r**	+++ +	+++ +	+++ +	−−− −	−−− +	+++ +	11
Verb Phrase	+++ **r**	+++ +	+++ **r**	+++ +	−−− −	+++ +	+++ +	11
Adjective Phrase	+++ **r**	+++ +	+++ +	++− +	−−− −	−−− +	+++ +	11
Adverb Phrase	+++ **r**	+++ +	+++ +	++− +	−−− −	−−− +	−−− +	11
Quantifier Phrase	+++ **r**	+++ +	+++ +	++− +	−−− −	−−− +	−−− +	11
Sentence	+++ **r**	−−− +	+++ +	+++ **r**	−−− −	+++ +	+++ +	11
INFL/ Auxiliaries	+++ **r**	+++ +	+++ +	+++ +	++− +	+++ +	+++ +	13
Conjunction Phrase	+++ +	++− +	+++ +	−−− −	−−− −	+++ +	−−− +	9
Complementizer/ Subordinating Conjunction Phrase	+++ +	+++ +	−−− +	+−− +	−−− −	+++ +	−−− +	11
Specifiers	+++ +	+++ +	+++ +	++− +	+++ +	−−− +	−−− +	13
Complements	+++ +	−−− −	−−− −	−−− −	−−− −	+++ +	+++ +	5

Key: Grammars: English/Spanish/Japanese / Universal Grammar; Rule Status: **r** = required; + = permitted; − = not present

to add context-sensitivity to the context-free phrase structure rules. Constraints can also be used for semantics, eliminating syntactically legal but semantically impossible parses, e.g.

```
;; Allow "Philadelphia, PA" but not "Philadelphia, yesterday"
(when (and (COMP.head.location)        ; comp's head is loc.
           (not N''.head.location))    ; n's head isn't loc.
  (halt))                              ; don't apply rule
```

In the appositive attachment rule, attach location noun phrases only to other location noun phrases.

Lexical constraints are language-specific, linked to a word or phrase in a particular language, e.g. *old enough.* Syntactic and semantic constraints are more likely to be language-independent. Syntactic structures are compositional: they are made up of small, simple structures, and thus they are learnable. In sum, we can take advantage of the relative language-independence of syntax and semantics in the construction of grammar contraints.

Certain functions recur in the grammar and should be optimized for efficiency: syntactic and semantic agreement checking (subject-verb agreement, adjective-noun agreement in Spanish, classifier-noun agreement in Japanese), as well as more complex functions. For example, only transitive verbs should take direct objects, rejecting **Frank coughed the computer,* and only those direct objects that are semantically compatible should be permitted, rejecting *#Judy drank the computer.* Our approach to these tests is novel in that it is multilingual, relying on calls to a (language-independent) knowledge base. Determining the appropriateness of an object is accomplished using a semantic call in the rule

V" $\rightarrow$ V' NP

This semantic call uses information from the knowledge base to determine whether the available object satisfies the verb's semantic restrictions. Of course, four-word sentences do not present much of a problem. The test's real utility comes from the fact that it is able to prevent spurious sub-parses inside of fifty-word sentences.

Similarly, semantic tests are used to determine prepositional phrase attachment. Thus *Maria rented a car with a stereo* has a semantically preferred bracketing:

[[Maria] [rented [a car [with a stereo.]]]]

while the syntactically similar *Pam rented a car with a credit card* is bracketed differently:

[[Pam] [[rented [a car]] [with a credit card.]]]

A credit card is a *financial instrument,* and renting is a *financial transaction.* Thus the financial instrument can be used as the instrument of a financial transaction, so *credit card* attaches to *rent,* based on the information in the knowledge base about financial transactions. A *stereo* is not a financial instrument, but it is a part of a car, so it can attach to *car,* but not to *rent.*

One of the uses of constraints is within the *conjunction rules.* Conjunction, particularly in long sentences, multiplies ambiguity drastically. To limit the ambiguity, we use a novel combination of syntactic and semantic constraints, implemented them in a language-independent fashion. Almost anything can be conjoined syntactically, but there are strong preferences for conjoining structures that are parallel both syntactically and semantically.

[cars in San Francisco] and [trucks in Boston]
NOT cars in [[San Francisco] and [trucks in Boston]]
NOT cars in [[San Francisco and trucks] in Boston]

The conjunction felicity tests are language independent. They examine semantic similarity (*car* and truck), as well as syntactic parallelism (common noun (*cars, trucks*) + same preposition (*in*) + proper noun (*San Francisco, Boston*)). They can be tailored to the needs of a particular language, but the overall function need not be written again for each language. We use

a series of tests (presence of modifiers, prepositional phrases, meaning similarities) and weight parses based on the overall similarity score, preferring the parse with the best semantic similarity and syntactic parallelism.

Structure-Building Augmentations. The output of Syntactic Analysis is not simply a parse tree, but also a language-independent structure called a functionally labelled template (FLT), with sentence functions like "subject" and "object" labelled. A parse tree represents only surface relationships, and a further level of representation is necessary, in order to handle, for example, alterations in surface relationships (e.g. passive vs. active) and syntactic gapping. Only this enhanced representation offers information detailed enough to determine thematic relations and perform reference resolution, which take place in the semantic and discourse modules.

In our implementation, the FLT resembles the f-structures in Lexical Functional Grammar (Bresnan 1982), while our parse tree is analogous to LFG's c-structure. The major difference between our approach and current LFG implementations is our experience with large-scale, multilingual text understanding applications. Idiomatic expressions, long sentences, and idiosyncratic elements like punctuation, addresses, and latitude/longitude, require an efficient grammar design that takes advantage of X-bar theory and language-independent semantic constraints, while allowing for language-specific features.

The f-structure contains functional information, not simply surface constituents as in the c-structure. Each phrase structure rule contains structure-building functions to generate the FLT. In the rule S $\rightarrow$ NP VP, for example, the NP will be the subject of S, while the VP will be the predicate.

Altering Surface Relationships (Logical Structure Alternations). Building structures showing functional relations is necessary for any non-toy application, since parse trees alone fail to represent necessary linguistic information. Our novel approach to this problem uses language-independent functions. Parse trees represent the surface order of constituents, but do not convey underlying relationships. The surface subject of a passive sentence is in some sense a logical object, a relationship that can be represented in the FLT. Similarly, in languages like Spanish, the subject is often elided, but the grammar must mark a logical subject for discourse purposes.

In Japanese, topic phrases (marked with *wa*) serve as logical subjects if the sentence has no other subject. We exploit the FLT to convey the logical relationship, a possibility the parse tree alone does not offer. The topic phrase is "moved" into the subject position in the FLT, and marked as a (surface) topic for later discourse purposes. Semantics can then find the subject (the surface topic) in the FLT and place it in the correct thematic role relationship to the main verb.

In English, Spanish, and Japanese, preposed set delimiters are "moved":

Of these, 24 were male.
$\Longrightarrow$ 24 of these were male.

The FLT itself is a means for forcing multilingual grammar output into sufficient conformity so that language-independent Semantic Interpretation can locate syntactic components. The FLT is a language-independent formalism in which to represent syntactic relations without recourse to word order. It explicitly labels X-bar relationships like specifier and complement.

Syntactic Gaps. In more complex situations, e.g. gap-filling, a dummy structure must be created and then, where possible, linked to the appropriate referent. For example, Spanish, like English, allows numbers as noun phrases with empty heads: *Hay dos.* ("There are two.") To create a gap to trigger reference resolution in Discourse Analysis, we build an empty head structure, with dos filling the quantifier slot. Only grammars that can identify gaps can supply enough information for Discourse Analysis to occur. Some gaps should be filled at the syntax level (e.g. relative clause gaps), since all the necessary information is sentence-internal. The following example illustrates an empty head in a quantifier phrase: *dos* (two) is a quantifier, but it forms a noun phrase by itself (cf. "I saw two"), since there is no head noun. For discourse purposes, the FLT contains a marker indicating an empty NP, so that later processing can determine the antecedent, i.e. "two what?"

These routines to create dummy structures are language-independent functions, as are the gap-filling routines to fill the structures. Because both functions are part of the language-independent grammar core, the writer of a grammar for an additional language does not need to create them from scratch.

Parser Implementation

To manipulate the grammars described above, we use a shift-reduce, bottom-up, all-parses parser, implemented in LISP. Because our parser uses Tomita's algorithm, which is quite fast, we have not had to sacrifice accuracy for speed. Tomita (1986) dealt only with rules; we have modified his parser to permit augmentations, for the reasons we have presented in the previous section. In large-scale applications, augmentations are vital in order to add semantic and contextual informationand to prevent thousands of incorrect parses. Augmentations allow us to select the preferred parse, which is sent on to Semantic Interpretation for further processing.

If more than one parse is possible, our system selects the preferred parse on the basis of a weighting scheme

drawn from scores assigned in the lexicon, idiom patterns, and grammar. When parsing is completed, the scores relating to each parse template are calculated, and the top parse is selected.

We use a top-down preparsing approach to take some of the burden off the bottom-up parser. Preparsed phrases enter Syntactic Analysis already parsed, and are attached as a unit to the rest of the parse. For a complementary perspective on combining top-down and bottom-up approaches, see Rau & Jacobs (1988).

Real-world text understanding tasks require graceful degradation, rather than outright failure. Backup procedures are required at every stage of processing. We supplement our parser with a Debris Parser, which uses phrases to recover syntactic data when a full sentence parse is unavailable (Kehler *et al.* 1990). The Debris Parser uses the incomplete parsing results, supplementing them with semantic information about the main verb's predicate argument structure. It then constructs the most likely thematic role structure. The Debris results reenter normal semantic processing, and are indistinguishable from normally parsed sentences.

Applications

This grammar approach has been implemented in a natural language processing core engine under development for several years. It has been used in multilingual applications for four projects, and on English alone for several more. Domains covered include terrorism, financial, medical, and trade texts in English, Spanish, and Japanese. The projects ranged in scale from several hundred single-page texts to a thousand texts a day. Most texts were between a paragraph and two pages long. The writing style ranged from simple business writing, with sentences averaging about 12 words, to newspaper writing, with sentences averaging about 25 words, and, particularly in Spanish and Japanese, complicated structures including many different kinds of ellipsis. The English terrorism texts were translated from Spanish, reducing some of the complexity. Most of the systems were demonstration prototypes; one was a fielded system. All of the systems were used for data extraction, and most fed a database, usually in Sybase, and usually object-oriented in design. Several used the extracted data to support visualization systems. The portability of the grammars to new domains and applications has demonstrated the success of our approach.

Conclusion

While a "universal grammar" approach to multilingual grammar development is tempting, our experience demonstrates that it is far too immature for large-scale natural language understanding. But neither should we build entirely language-specific grammars and lose the insights of several decades of research in generative grammar. We combine linguistic insight with robust, in-depth parsing. Our use of X-bar theory and language-independent semantic constraints draws on the strengths of current syntactic theory without sacrificing robustness and scalability. Our use of separate grammars for each language allows us the flexibility to capture language-specific lexical insights, while our use of language-independent functions captures linguistic generalizations. Our implementation includes innovative handling of diverse phenomena, such as (1) syntactic gaps, (2) logical structure alternations, and (3) conjunctions. These innovations strengthen our large-scale and multilingual natural language processing system.

References

Berwick, R. 1987. Principle-based parsing. Technical Report 972, MIT Artificial Intelligence Laboratory.

Bresnan, J., ed. 1982. *The Mental Representation of Grammatical Relations.* Cambridge, MA: MIT Press.

Chomsky, N. 1967. *Aspects of the Theory of Syntax.* Cambridge, MA: MIT Press.

Chomsky, N. 1981. *Lectures on Government and Binding.* Cambridge, MA: MIT Press.

Fong, S., and Berwick, R. 1989. The computational implementation of principle-based parsers. In *Proceedings of the International Workshop on Parsing Technologies.* Carnegie Mellon University.

Fong, S. 1990. Free indexation: Combinatorial analysis and a compositional algorithm. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics.*

Jackendoff, R. 1977. *X' Syntax: A Study of Phrase Structure.* Cambridge, MA: MIT Press.

Kehler, A.; Blejer, H. R.; Flank, S.; and McKee, D. 1990. A three-tiered parsing approach for operational systems. In *Proceedings of the AI Systems in Government conference*, 150–156.

Lin, D. 1993. Principle-based parsing without overgeneration. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics.*

Rau, L. F., and Jacobs, P. S. 1988. Integrating top-down and bottom-up strategies in a text processing system. In *Proceedings of the Second Conference on Applied Natural Language Processing*, 129–135.

Rothstein, S., ed. 1991. *Perspectives on Phrase Structure: Heads and Licensing.* Number 25 in Syntax and Semantics. San Diego: Academic Press.

Speas, M. 1990. *Phrase Structure in Natural Language.* Dordrecht: Kluwer.

Tomita, M. 1986. *Efficient Parsing for Natural Language.* Boston: Kluwer.

Woods, W. A. 1973. An experimental parsing system for transition network grammars. In Rustin, R., ed., *Natural Language Processing.* NY: Algorithmics Press.