Combining Research and Pedagogy in the Development of a Crosslinguistic Grammar Resource

Emily M. Bender
University of Washington

Proceedings of the GEAF 2007 Workshop

Tracy Holloway King and Emily M. Bender (Editors)

CSLI Studies in Computational Linguistics ONLINE

Ann Copestake (Series Editor)

2007

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

This paper describes how a graduate course in multilingual grammar engineering has been used to inform the development of the LinGO Grammar Matrix. When the course was first taught (in 2004), the Grammar Matrix consisted only of the cross-linguistic core grammar. Over time, the lab instructions for the students in the course sparked the development of extensions to the Matrix providing 'libraries' of analyses of crosslinguistically variable phenomena. At the same time, the students' course work has provided valuable feedback both in error checking of the core grammar and refinement of the libraries. Based on the experience of teaching this class for four years, I suggest that grammar engineering courses present a rich opportunity for the combination of pedagogy and research. Involving even beginning grammar engineers in on-going investigations can be rewarding for all involved.

1 Introduction

This paper is an exploration of how student work can be harnessed to further current research goals in a way that also benefits the students, and how individual faculty or researchers can efficiently combine both teaching and research roles. In particular, I present some reflections on these ideas from the vantage point of research on the LinGO Grammar Matrix (Bender et al., 2002) and the University of Washington's Linguistics 567: "Knowledge Engineering for Natural Language Processing". In Section 2, I briefly present the Grammar Matrix. Section 3 describes how the course is structured. Section 4 gives examples of how student work has provided valuable feedback to the Grammar Matrix. Section 5 outlines possible future directions for the course.

2 The Grammar Matrix

This section briefly introduces the Grammar Matrix, situating it in its theoretical context, and describing the contents of the core grammar and the phenomenon specific libraries.

2.1 Theoretical context

The LinGO Grammar Matrix (Bender et al., 2002; Flickinger and Bender, 2003; Bender and Flickinger, 2005; Drellishak and Bender, 2005) is a starter-kit designed to facilitate the rapid development of broad-coverage precision grammars. In addition, it is intended to promote consistency in semantic representations such that

[†]The Grammar Matrix project is a collaborative effort, and I would like to acknowledge the contributions of Scott Drellishak, Chris Evans, Dan Flickinger, Stephan Oepen, Kelly O'Hara, and Laurie Poulson, as well as the students in Linguistics 471 and 567 in 2004-2007. The original inspiration for the course came from Petter Haugereid. The Grammar Matrix project is supported by NSF grant BCS-0644097 and a gift to the Turing Center from the Utilika Foundation.

broader applications using a Matrix-derived or Matrix-compatible grammar can easily be adapted to additional languages by switching in different Matrix-derived or Matrix-compatible grammars. The Grammar Matrix has also emerged as an interesting platform for exploring generalizations and the range of variation across languages in syntax and the syntax-semantics interface. In this respect, it represents a kind of computational linguistic typology.

The Grammar Matrix is couched within the Head-driven Phrase Structure Grammar framework (HPSG; Pollard and Sag 1994), and uses Minimal Recursion Semantics (MRS; Copestake et al. 2005) for the semantic representations. It is implemented in tdl, a typed-feature formalism interpreted by the LKB grammar development environment (Copestake, 2002) and the PET parser (Callmeier, 2000). The Grammar Matrix is developed within the context of the DELPH-IN consortium (www.delph-in.net), as part of a larger constellation of grammars and associated software.

More broadly, the Grammar Matrix is an instance of multilingual grammar engineering. In this sense, it is similar in spirit to the ParGram project (Butt et al., 2002; King et al., 2005), the MetaGrammar project (Kinyon et al., 2006), KPML (Bateman et al., 2005), Grammix (Müller, 2007) and OpenCCG (Baldridge et al., 2007). Among approaches to multilingual grammar engineering (Bender et al., 2005), the Grammar Matrix's distinguishing characteristics include the deployment of a shared core grammar for crosslinguistically consistent constraints and a series of libraries modeling varying linguistic properties.

2.2 Core grammar

The core grammar consists of types and constraints which are meant to be crosslinguistically useful. The core grammar always, in fact, represents a set of working hypotheses about language universals within the general framework of the Grammar Matrix. As constraints are found to be incorrect for languages analyzed using the Matrix, they are removed from the core grammar. Types found to be irrelevant for particular languages are also intended to be moved out to the libraries, though in practice this process is slower, as the presence of irrelevant types in a grammar does not affect the analyses assigned to strings by the grammar, provided those types are not used in any rule or lexical item instances.

The core grammar focuses on the following six aspects of a grammar: (i) the basic feature geometry, (ii) basic construction types (e.g., head-subject phrases, head-adjunct phrases, and coordination phrases), (iii) semantic composition, or the way in which the semantic representations of phrases are computed on the basis of the semantic representations of their daughters and the contribution of the rule licensing the phrase, (iv) basic lexical types, including linking types associating syntactic with semantic arguments, (v) basic types for lexical rules, and (vi) collateral files for interaction with the parsing, generation and grammar development software (the LKB and PET).

2.3 Libraries and customization

The core grammar itself has already proven useful, jump-starting the development of several grammars including the NorSource grammar of Norwegian (Hellan and Haugereid, 2003), the Spanish Resource Grammar (Marimon et al., 2007) and the Modern Greek Resource Grammar (Kordoni and Neu, 2005). If the goal is the reuse of grammar code across languages, however, restricting the Matrix to those types and constraints which are valid across all languages is quite limiting: In other words, it seems likely that the analysis (or implementation) of, say, verb-final word order used in Japanese ought to also be applicable to another verb-final language, such as Malayalam.[1] This is the motivation behind the development of phenomenon-specific libraries which provide analyses of different variations on the same phenomenon (e.g., major constituent word order, AND-coordination, etc.) (Bender and Flickinger, 2005; Drellishak and Bender, 2005). These analyses are accessed through a website[2] which presents the user with a typological questionnaire and outputs a working grammar combining the Matrix core grammar with information from the libraries on the basis of the user's answers to the questionnaire.

The statement above about cross-linguistic applicability of particular analyses is a hypothesis to be tested: Given the interconnectedness of analyses of disparate phenomena within a grammar, it is not *a priori* obvious that one and the same analysis of a given phenomenon (e.g., verb-final order) will integrate properly with the required analyses of all the rest of the phenomena in two different languages. In fact, in developing the libraries to date, we have found them to be non-modular in several respects. It is not possible, for example, to fully specify the head-complement rule which is output by the basic word order module without also knowing whether adpositions, complementizers, and auxiliaries (if present) precede or follow their complements. Nonetheless, it is interesting to work towards universal coverage in the libraries while attempting to properly account for their interactions. It is in this way that the development of the Matrix becomes an exercise in computational linguistic typology.

The Matrix libraries are a type of parameterization of linguistic variation, and in that sense, this approach is similar to the Principles and Parameters approach (P&P) (Chomsky, 1981, *inter alia*). However, where P&P work typically tries to derive multiple disparate surface phenomena from each parameter, the Matrix libraries target one phenomenon at a time. Another important difference is that the Matrix is a grammar engineering project, producing grammar fragments which can be run against test suites to validate the interaction of the analyses (Oepen and Flickinger, 1998; Bender, 2006; Bender et al., 2007).

The current libraries address major constituent order, strategies for expressing sentential negation and (matrix) yes-no questions, a handful of lexical prop-

[1] Asher and Kumari (1997) give SOV as the basic order in Malayalam, but also state that there is a good deal of freedom of order of constituents even in unmarked sentences.

[2] http://www.delph-in.net/matrix/customize/matrix.cgi

erties (optionality of determiners, NP v. PP arguments of verbs, intransitive and transitive argument frames) (Bender and Flickinger, 2005) and AND-coordination (Drellishak and Bender, 2005). The coordination library in particular is based on a thorough typological study of the phenomenon in question (Drellishak, 2004). Current work is targeting case, verb-argument agreement in person, number and gender, tense and aspect, argument optionality, and demonstratives. In addition, we are developing general mechanisms for handling lexical rules and the interactions between them through the customization interface.

The general methodology for constructing libraries begins with a survey of the typological and syntactic research literature to map out the typological domain. Then we create analyses for each variant and construct questions to elicit information needed to decide between the variants from the linguist-user. The next step is to create the software to select and output analyses on the basis of the linguist-user's answers, while accounting for interactions with other existing libraries. Finally, we also create test items and filters for the regression testing system (Poulson, 2006; Bender et al., 2007) to validate the new functionality, check for regressions in previously covered territory, and document the new functionality for future regression testing purposes.

2.4 Goals of the project

This section has outlined the current state of the Grammar Matrix project. Our long-term goals for this project are: (i) to increase the gain of the jump-start, i.e., the size of the initial grammar fragments provided by the customization system, (ii) to facilitate the deployment of NLP technology such as grammar checkers, machine translation systems and computer assisted language learning software for low-density languages, (iii) to integrate the Grammar Matrix with other technologies for language documentation and to foster collaboration between field linguists and grammar engineers (Bender et al., 2004), and (iv) to further develop the research field of computational linguistic typology.

In the next sections, I describe the class which has been the driving force behind much of the development of the Grammar Matrix over the past four years, and the ways in which student work in the course provides feedback which is folded back into the Grammar Matrix.

3 Course overview

The pedagogical goals of the course are (i) to give students hands-on experience in the development of substantial linguistic resources for NLP; (ii) to illustrate the importance of test suite creation in the development and evaluation of such resources; and (iii) to explore the nature of linguistic hypothesis testing given the interconnectedness of subsystems within grammars. The students are typically graduate students in computational linguistics,[3] though graduate students and advanced undergraduates in general linguistics and computer science also attend. All have taken an introductory theoretical HPSG syntax course as a prerequisite.

3.1 Course outline

The course is organized around weekly lab assignments. The course meetings are divided into lectures covering background material and discussion sessions which are driven by student questions and which typically involve interactive work with the grammar development environment (the LKB; Copestake, 2002).

In the first week, the students get to know the LKB by extending a grammar for a small fragment of English. They also choose the language they will be working with for the rest of the quarter and find reference grammars. Each student must choose a different language which has not been studied before in the class. In four years, we have covered 42 languages, from American Sign Language to Zulu, representing the language families Indo-European (15 languages), Afro-Asiatic (3), Niger-Congo (3), Altaic (2), Austronesian (2), Dravidian (2), Na-Dene (2), Sino-Tibetan (2), Uralic (2), Eskimo-Aleut (1), and Uto-Aztecan (1).[4] In addition, the languages covered include two creoles, three languages the Ethnologue classifies as isolates or quasi-isolates (Basque, Japanese, and Korean), one signed language (ASL), and one invented language (Esperanto). Students typically end up working with languages they have not studied before.[5]

In the second and third weeks, the students create test suites for their languages covering the phenomena to be analyzed in the class. In the process of creating these test suites, they become familiar with their reference grammars and in some cases seek out native speaker consultants to ask for acceptability judgments. The test suites include both positive and negative examples, with the latter typically outnumbering the former. Students are encouraged to use a restricted vocabulary, to illustrate each phenomenon with sentences that are as simple as possible, and to include examples illustrating the interaction of multiple phenomena (e.g., negative questions).

In the remaining seven weeks, students incrementally extend grammars for fragments of their languages. They begin by customizing and downloading a copy of the Matrix through the Matrix customization web page. If the libraries do not yet include an analysis of the appropriate variation on some phenomenon, the starter grammars will be correspondingly smaller (omitting coordination, say, or sentential negation).

In subsequent labs, students add case and agreement (as appropriate for their

[3] Particularly in the professional MA program in computational linguistics, for which this course serves as an elective, see http://compling.washington.edu.

[4] These counts are based on the Ethnologue's classifications of the languages in question (Gordon, 2005).

[5] Artificial languages are generally not allowed. The criterion is that the language must have or have had native speakers, which Esperanto does (Gordon, 2005).

languages), move from a full-form lexicon to one incorporating lexical rules, and add the rules and types needed to treat argument optionality (*pro*-drop), demonstrative adjectives or determiners, (other) adjectives and adverbs, embedded declaratives and polar interrogatives, and expressions of ability (e.g., modals). In this, the students are guided by lab instructions describing the phenomena to be analyzed, enumerating known variations on those phenomena, and suggesting Matrix-compatible analyses for each variation. When the lab instructions do not anticipate a variation that turns up in one of the languages, the student working on that language and I work together to produce an appropriate analysis, which the student tests by implementing it.

The lab requirements crucially include a write-up explaining how the phenomena treated that week manifest in the student's language, detailing the analyses that the student developed, and describing any difficulties that the student encountered. These write-ups are critical for communication between the students and the instructor in the on-going development of the grammar, and for the incorporation of feedback from the course grammars into the Grammar Matrix itself.

As a grand finale, we use the LOGON open-source translation software (Oepen et al., 2004a; Bond et al., 2005) to put the grammars together into an NxN machine translation system. The coverage is necessarily limited, but the students are always excited to see their grammars 'talking' to each other, and it serves as a motivating end point for the grammar development.

3.2 Test suites and grammar evolution

The students track the progress of their grammars using the [incr tsdb()] competence and performance profiling system (Oepen, 2002). [incr tsdb()] allows the students to compare the coverage, ambiguity, and overgeneration of their grammars over their test suite across different stages in the grammar's development, and to discover which test items have different analyses across test runs. Students are encouraged to use this not only for overall benchmarking but also to explore the consequences of particular changes to the grammar in the process of grammar development.

While the bulk of the test suite development is done at the beginning of the quarter, the test suites continue to evolve over time for a variety of reasons: In some cases, students change their transcription system or opt for a more morphophonologically abstract representation, and end up editing their test suites consistently. In other cases, in the course of grammar development, students discover and correct errors in their test suites. A third possible reason for test suite evolution is the addition of further examples to test interactions and corner cases unnoticed in the original test suite development.

Figure 1 illustrates the evolution of the test suites on the one hand and coverage and ambiguity over those test suites on the other for two grammars: a Hebrew grammar developed in 2006 by Margalit Zabludowski and a Zulu grammar devel-

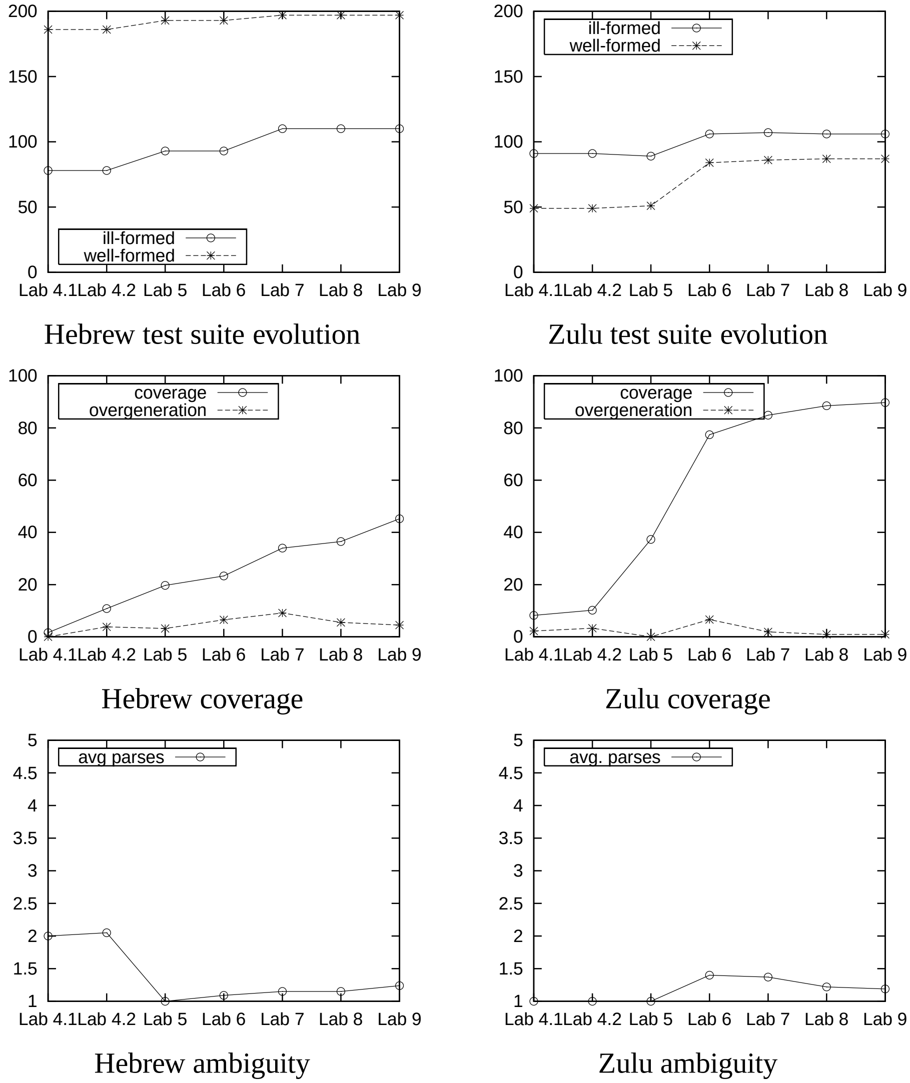

Figure 1: Test suite and grammar evolution for two languages

oped in 2007 by Kelly O'Hara. Each point on the graph represents one week's lab.[6] As can be seen from the graphs, the grammar writers were able to substantially increase their coverage over grammatical sentences without overgenerating.[7] The ambiguity rates reflect typical progress as well: Some changes to the grammars introduce spurious ambiguity, which is corrected in later versions. At the same time, there is a gradual increase in real ambiguity as the grammars expand to cover more phenomena.

3.3 Challenges and benefits

The course described here is tremendously challenging for both the students and the instructor, but also has benefits in proportion to the challenges. For the students, the challenges largely come from the fact that they are asked to master quite a bit of material quickly and in parallel: The software we use (the LKB, [incr tsdb()], emacs), the Grammar Matrix itself (including its assumptions and general structure), and the language they are working with (which is often completely new to them). For students coming from a primarily linguistic background, there are also engineering skills to master, such as the debugging process. In addition, the assignments are generally open-ended. In my experience, the students have typically taken this open-endedness as a challenge to extend their grammars further and stretch for broader coverage and higher precision. The flip side of these challenges are the benefits: Because the course is intimately connected with a larger research project, the students are participating in knowledge creation, which they tend to find extremely motivating. At the same time, having all of their effort in the course directed to a single, original, course project gives them a potential spring board for further research as well as valuable experience to point to when they are on the job market

From the instructor's point of view, the first challenge is again the open-endedness of the assignments. This makes student evaluation difficult, as the students submit highly diverse work, both in terms of the specific aspects of their languages they work on and in terms of the distance they go with the assignments. A second challenge is dealing with many languages at once. Here, I found that giving specific instructions for lab write-ups guiding students to provide the relevant background information about the analyses they are implementing has been very important. In fact, the ability to deal with many languages at once is one of the main benefits from my point of view as well, especially dealing with the same or similar phenomena in many languages. In order to answer a question about a particular analysis (e.g., of case) in a Matrix-derived grammar, I need to review the relevant aspects of the Matrix. Not only is it easier to answer a question about case in another language while all the information is fresh in my mind, it is also very useful to get a compar-

---

[6] The first entry for Lab 4 represents the grammar fragment as downloaded from the customization system. The second represents the addition of some of the missing vocabulary.

[7] The higher coverage of the Zulu grammar as compared to the Hebrew mainly reflects the fact that the Hebrew test suite staked out a more ambitious fragment of the language.

ative view on case (or any other phenomenon) by considering data and especially difficulties in analysis from multiple languages all at once. More generally, the students' work provides feedback to the Matrix which allows for error detection, library development, and library refinement, as discussed in the next section.

4 Feedback to the Matrix

The Grammar Matrix project faces sizeable hurdles in evaluation and validation (Poulson, 2006). The Matrix core grammar itself cannot be directly tested, as it is not a grammar of any particular language, and cannot parse or generate any strings. The Matrix customization system and libraries can be validated through autogeneration of test suites for abstract language types (Bender et al., 2007), but this is only validation, and not evaluation. The test suite generation system allows us to verify that the libraries have the behavior we intend them to, and that they interact in reasonable ways. It does not provide a means of testing the linguistic correctness of the system, i.e., its typological predictions. Are the libraries complete? Do the libraries interact in ways that predict the facts of actual human languages? This section will describe how student work in the class can help with detecting errors in the core grammar (Section 4.1) and with detecting lacunae in existing libraries (Section 4.3). In addition, the course played an important role in the development of the initial set of libraries (Section 4.2).

4.1 Error detection

The Grammar Matrix core was originally derived from the English Resource Grammar (ERG; version of November 2001), by taking ERG and removing all types or constraints that appeared to be English-specific. This process was informed by comparison with the JACY grammar of Japanese (Siegel and Bender, 2002). Nonetheless, this process was expected to be somewhat error-prone: English-specific constraints could easily be missed, and constraints which are in fact general could also be removed. In the intervening six years, the core grammar has grown and been refined. We have added linking and lexical rule types, which were not modeled directly on the analogous types in the ERG (though the ERG was a reference point), and we have explored new analyses of phenomena such as the marking of illocutionary force and the licensing and interpretation of dropped arguments.

The core grammar is abstract in the sense that it cannot in itself parse or generate any sentences. That is, while the Matrix core grammar has many of the essential ingredient for actual rules and lexical entries, in includes no fully specified rules. Among the types that define the phrase structure rules, for example, there is an abstract head-complement phrase type which describes the syntactic and semantic effects of combining a head with a complement it is seeking. This type does not specify the order of its head and non-head daughters. In any particular grammar it will be cross-classified with either the head-final or the head-initial type (or both, instantiated by separate rule instances).

Unfortunately, abstract does not necessarily mean simple, and even these relatively underspecified types bear constraints which relate to many different analyses. For example, the Matrix provides a feature MC (for 'main clause') which records whether a constituent displays phenomena restricted to root (or alternatively subordinate clauses). For example, in English, subject-auxiliary inversion is limited to root clauses, and in German (and similar V2 languages) the major constituent order differs between the clause types. A constituent which is [MC +] is restricted to main clauses, while a [MC −] constituent is restricted to subordinate clauses. A constituent which is underspecified for MC can appear in either. Following the ERG, we provide a third possible (non-underspecified) value *na* ('not applicable') for non-clausal constituents. Analyses which make use of this feature rely on its being specified for every constituent, and thus all of the phrase structure rules must inherit a specification for the feature from some supertype.

This is just one example of the way in which individual types bear constraints which relate to multiple different linguistic phenomena, even in the abstract core grammar. Since the core grammar was derived from a broad-coverage grammar and since it aims to support the development of similarly broad-coverage grammars, it is a complex object. Thus in general, even with perfect knowledge of linguistic typology, it would not be possible to examine the core grammar directly and find all of the errors it contains. It is only in applying the Matrix to particular languages that we can hope to find out where the current working hypotheses are incorrect.

In addition to removing constraints which turned out not to be universal, the analysis of new languages sometimes leads to the addition of types to the Matrix core. The most interesting example from the past year is the introduction of ternary rules for certain constructions. The ternary rules types were added to handle negation in Hausa (for the grammar developed by Kelsey Hutchins), which is marked by two particles, one on either side of the clause:[8]

(1) Hausa (hau)
bàà rashìn nāmàà zâi kashè mùtûm ba
NEG lack.of meat FUT kill person NEG
'It is not that lack of meat will kill a person.' (Newman, 2000, 363)

The analysis we developed for such examples involves a ternary rule which requires a finite clause as its middle daughter and specific lexical items (the left and right negation markers) as its left and right daughters, as illustrated in Figure 2. The middle daughter is the syntactic head, but the construction itself functions as the semantic head.

Previously, Matrix-derived grammars (following the ERG) had handled all constructions with unary or binary phrases, using recursing binary phrases to model

[8]For improved automatic discovery of this document, all examples are labeled with the name of the language represented, followed by the ISO 639-3 language code, in parentheses.

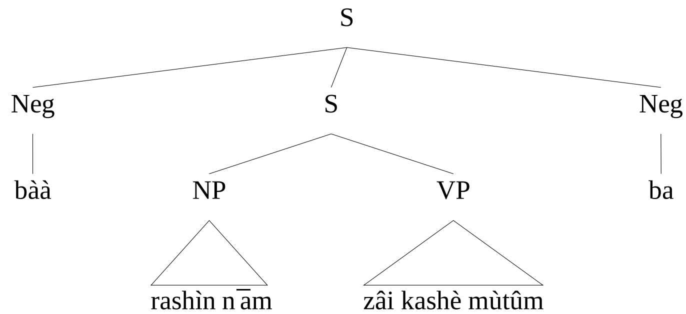

Figure 2: Schematic tree for Hausa example (1), 'It is not that lack of meat will kill a person.'

variable arity in, for example, head-complement constructions with multiple complements for a single head and coordination of more than two coordinands. This construction in Hausa is different, however, in that it requires specifically three daughters. While it would certainly have been possible to attach one of the two markers lower than the other, such an analysis would have required diacritic features in order to require the second marker (and ensure that nothing else attached in between).

The ternary rule types were immediately useful for another grammar being developed at the same time by Sarah Churng for American Sign Language (ASL). In ASL, many grammatical features are signaled through 'non-manual markers' (NMMs) (Baker and Padden, 1978), typically facial expressions. These NMMs extend over whole constituents, and in fact constitute a separate parallel channel to the signal. For the purposes of building a Matrix-derived grammar, we developed a transliteration system which indicates the NMMs through left and right brackets.

(2) American Sign Language (ase)

JOHN	⟨ne	BUY	HOUSE	ne⟩
John	nm-neg	buy	house	nm-neg

'John is not buying a house.'

These left and right pairs (for negation and yes-no questions) were then parsed by the same kind of ternary-branching rules required for Hausa.[9]

Error detection in the Grammar Matrix depends on the deployment of the Matrix in grammars for many languages, but building such grammars is time-consuming, even with the jump-start provided by the Matrix. Feedback does come

[9]The types for ternary rules are another case where there is tension between the strict interpretation of the core grammar as universally valid types and constraints and other classifications of the types. It is quite likely that there are languages for which we will never need ternary rules, and yet since these types do much the same work as the basic types for unary and binary rules, they are currently stored as part of the core grammar.

On the other hand, these types might turn out to be useful in many languages when we consider punctuation.

in from research groups around the world using the Grammar Matrix, but the communication is uneven in such cases: users may simply decide to edit the Matrix core grammar without reporting back to the Matrix developers. In the context of the class, on the other hand, I get to explore as many languages as there are students, with extensive information on how each grammar was able to use the Matrix, and where the Matrix needed to be modified.

4.2 Library development

The initial set of libraries (Bender and Flickinger, 2005) also grew directly out of the course. In the first two years of teaching this course (in 2004 and 2005), I developed and expanded an initial set of lab instructions, covering basic word order, case, agreement, modification, the expression of ability, sentential negation, argument optionality, and matrix and embedded statements and yes-no questions. In some cases, these directions were fairly open-ended. In others, by the end of the second time teaching the course, they were specific enough that it was clear that there was only a little work left to make them so precise a machine could follow them.

With the initial libraries and customization system in place, the Grammar Matrix can provide a greater jump-start (provided appropriate options are available within the libraries for the language in question). This means that students can explore their languages in greater detail within the 10-week course. In the most recent course, in addition to the phenomena from years one and two, we have been looking into the marking of discourse status (in particular definiteness and demonstratives) and coordination, with the latter supported by and providing feedback to the Matrix coordination library (Drellishak and Bender, 2005).

4.3 Library refinement

Just as the class provides a chance to find errors and lacunae in the Matrix core grammar, it also provides crucial feedback to the libraries. As the negation and yes-no question libraries were not based on thorough studies of the typological literature, it is not surprising that we have already turned up cases that were not covered. For negation, this includes the circumfixal negation described above for Hausa and ASL. For the question library, Wendy Bannister's work on Malayalam turned up what is likely a common strategy: question marking via inflection on the main verb, as illustrated in (3) (Asher and Kumari, 1997, 8):

(3) Malayalam (mal)

a. Avan vannu
He come.past
'He came.'

b. Avan vann-oo
He come.past-Q
'Did he come?'

In addition, the French grammar developed by Fabiola Henri and Gwendoline Fox[10] turned up another problem with the question library: Unlike the negation library, the question library only allowed for one kind of question marking per language. In fact, however, French is representative in allowing multiple strategies: a sentence-initial question marker (*est-ce que*) and subject-verb inversion, as illustrated in (4).

(4) French (fra)

a. Est-ce qu' il est parti?
Q 3SG.NOM.MASC be.3SG leave.PAST-PART.
'Has he left?

b. Est-il parti?
be.3SG-3SG.NOM.MASC leave.PAST-PART.
'Has he left?

Even with the coordination library, which was based from its inception on a typological study, the course grammars have turned up an unhandled case: The coordination library currently provides for multiple coordination strategies within a single language, each with its own coordination mark, but any given strategy will use only one mark. Michelle Neves's work on Indonesian showed this to be inadequate, as Indonesian can mix the coordinators *serta* and *dan* in the same coordinate structure, typically using *dan* for all but the last coordinand pair of coordinands, which are joined instead by *serta* (Sneddon, 1996, 339-340):[11]

(5) Indonesian (ind)
Ini untuk hiasan dinding dan meja serta kursi
DEM. for decoration wall CONJ table CONJ chair
'These are decorations for walls, tables and chairs.'

The course grammars also provide interesting information on the interaction between libraries. For example, one of the course grammars for 2007 (developed by Ryan Georgi) was for Modern Standard Arabic (MSA). In MSA, word order interacts with agreement: Both VSO and SVO are possible. In VSO word order, the verb agrees with the subject in person, number, and gender, whereas in SVO word order, there is only agreement in person and gender (Soltan, 2006, 240).

[10]Students in a similar course at the LSA Institute, Stanford 2007, taught by Stephan Oepen, Dan Flickinger, and myself

[11]Sentence (5) is constructed on the basis of the Sneddon, but does not appear in this form in the book.

(6) Modern Standard Arabic (arb)

a. ʔal-ʔawlaad-u qaraʔ-uu d-dars-a
the-boys-NOM read-3.PL.MASC the-lesson-ACC
'The boys read the lesson.'

b.*ʔal-ʔawlaad-u qaraʔ-a d-dars-a
the-boysNOM read-3.SG.MASC the-lesson-ACC
'The boys read the lesson.'

c. qaraʔ-a l-ʔawlaad-u d-dars-a
read-3.SG.MASC the-boysNOM the-lesson-ACC
'The boys read the lesson.'

c.*qaraʔ-uu l-ʔawlaad-u d-dars-a
read-3.PL.MASC the-boysNOM the-lesson-ACC
'The boys read the lesson.'

The word order library does not yet allow for variable word order of this kind. Verb-final (i.e., variation between SOV and OSV), verb-initial (variation between VSO and VOS), and free word order (of major constituents) are allowed, but not yet variation between VSO and SVO. It is not clear how soon the customization system will achieve the level of complexity required to specify an MSA-style system through the customization interface. Nonetheless, as we extend the word order library and begin to develop a library for agreement as well, MSA provides an interesting case to work towards.

4.4 Summary

This section has described how student work based on the Grammar Matrix in the grammar engineering course has contributed to error detection in the core grammar as well as to the development and refinement of the libraries. To a certain extent, any context in which the Matrix is applied to new languages will have similar benefits. However, there are some ways in which the classroom context is particularly helpful, compared to, for example, feedback from other research groups using the Grammar Matrix. The first is the degree of detail that is available. The students turn in multiple versions of their grammars, along with write-ups of the linguistic data analyzed and the analyses themselves. The grading work for the course thus doubles as information gathering for the Matrix project. The second major benefit of the classroom context is the coordinated, focused attention of many participants (students and instructor) on the same phenomena at the same time. It is much easier to integrate new information about different languages in this format, than when the information comes in a less coordinated fashion. Finally, as noted above, the pedagogical work of developing the lab instructions fed directly into the research/engineering work of developing the libraries. At the same time, it should be noted that the course grammars alone are not sufficient to test the Grammar Matrix. In particular, they remain small grammars, with only 10 weeks of development time. To learn how the various proposed analyses scale as grammars reach both interesting coverage and interesting ambiguity, the Matrix needs to be embedded in grammars undergoing sustained development.[12]

5 Future Directions

As the jump-start provided by the Grammar Matrix grows, the course grammars should attain greater complexity, even within the same 10-week time period. Initially, this will simply mean grammars which cover more phenomena. Beyond a certain level of complexity, however, I anticipate bigger changes. Within a few years, it should be feasible to have the students collect small corpora, and then process those corpora with their grammars. This will quickly turn up additional phenomena to work on (cf. Baldwin et al., 2005).[13]

In addition, as the grammars gain complexity they will also display more ambiguity. This makes it interesting to explore the creation of treebanks in the Redwoods style (Oepen et al., 2004b), where grammar engineers select among the trees proposed by the grammar on the basis of minimal discriminants. These treebanks can be used to train parse selection models (Toutanova et al., 2002). They will also represent small but interesting resources for low-density languages.

Finally, a recent project at the Turing Center at the University of Washington[14] has been piloting a many-to-many machine translation system based on the LOGON machine translation infrastructure (Oepen et al., 2004a; Bond et al., 2005) and using nine grammars from the first four years of the grammar engineering class.[15] The current system has only a toy vocabulary but works across an interesting range of grammatical phenomena. As this MT system grows more robust, it will be interesting to explore adding the course grammars to it as they are produced.

6 Conclusion

The Grammar Matrix project is well-suited to harnessing student work. It needs input from many languages, and relatively basic input is still quite valuable: the first grammatical phenomena one might try to account for in a language (e.g., word order, valence patterns, case, agreement) are currently under development in the Grammar Matrix. As the customization system grows, each new grammar will still provide useful information: either the libraries will handle the variants found in a grammar and the student can explore additional phenomena at the boundaries of

[12]Thanks to an anonymous reviewer for highlighting this point.

[13]Note that in order to make this practical, however, we will need to handle standard orthography as well as morphophonology. Currently, in order to focus on morphosyntax, I advise students to abstract away from these, working with a transliteration system and assuming a morphophonological preprocessor.

[14]http://turing.cs.washington.edu

[15]The grammars are for Armenian, Esperanto, Farsi, Finnish, Hausa, Hebrew, Icelandic, Italian and Zulu. In addition, we have a purpose-built grammar for English with similar coverage.

Matrix development, or the language will turn up a new variant to be incorporated into one of the libraries, or both.

There is no denying that the course is very intense for both the students and the instructor. To the extent that it is intense, it is also rewarding, certainly for me and I believe also for the students. I find that most of the additional work I put into teaching this course (above the commitment required for other courses) is effectively research effort, in that it feeds back into the Grammar Matrix project.

I also believe that there should be similar opportunities to integrate student course work into sustained projects elsewhere within the field of grammar engineering, as we need detailed attention to many separate linguistic phenomena. The problem in many cases is to find ways to lower the barriers to entry, such that the student projects become tractable, while also maintaining checks on the quality of the data or analyses added to the system.

References

Asher, Ronald E. and Kumari, T.C. 1997. *Malayalam*. London: Routledge.

Baker, Charlotte and Padden, Carol. 1978. Focusing on the Non-manual Components of American Sign Language. In P. Siple (ed.), *Understanding Language Through Sign Language Research*, pages 27–57, New York: Academic Press.

Baldridge, Jason, Chatterjee, Sudipta, Palmer, Alexis and Wing, Ben. 2007. DotCCG and VisCCG: Wiki and Programming Paradigms for Improved Grammar Engineering with OpenCCG. In Tracy Holloway King and Emily M. Bender (eds.), *Proceedings of the GEAF 2007 Workshp*, Stanford, CA: CSLI.

Baldwin, Timothy, Beavers, John, Bender, Emily M., Flickinger, Dan, Kim, Ara and Oepen, Stephan. 2005. Beauty and the Beast: What running a broad-coverage precision grammar over the BNC taught us about the grammar — and the corpus. In Stephan Kepser and Marga Reis (eds.), *Linguistic Evidence: Empirical, Theoretical, and Computational Perspectives*, Berlin: Mouton de Gruyter.

Bateman, John A., Kruijff-Korbayová, Ivana and Kruijff, Geert-Jan. 2005. Multilingual Resource Sharing Across Both Related and Unrelated Languages: An Implemented, Open-Source Framework for Practical Natural Language Generation. *Research on Language and Computation, Special Issue on Shared Representations in Multilingual Grammar Engineering* 3(2), 191–219.

Bender, Emily M. 2006. Grammar Engineering for Linguistic Hypothesis Testing. In *Proceedings of Texas Linguistic Society X*, Stanford: CSLI Publications.

Bender, Emily M. and Flickinger, Dan. 2005. Rapid Prototyping of Scalable Grammars: Towards Modularity in Extensions to a Language-Independent Core. In

Proceedings of the 2nd International Joint Conference on Natural Language Processing IJCNLP-05 (Posters/Demos), Jeju Island, Korea.

Bender, Emily M., Flickinger, Dan, Fouvry, Frederik and Siegel, Melanie. 2005. Introduction. *Research on Language and Computation, Special Issue on Shared Representations in Multilingual Grammar Engineering* 3(2).

Bender, Emily M., Flickinger, Dan, Good, Jeff and Sag, Ivan A. 2004. Montage: Leveraging Advances in Grammar Engineering, Linguistic Ontologies, and Mark-up for the Documentation of Underdescribed Languages. In *Proceedings of the Workshop on First Steps for Language Documentation of Minority Languages: Computational Linguistic Tools for Morphology, Lexicon and Corpus Compilation, LREC 2004*, Lisbon, Portugal.

Bender, Emily M., Flickinger, Dan and Oepen, Stephan. 2002. The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. In John Carroll, Nelleke Oostdijk and Richard Sutcliffe (eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan.

Bender, Emily M., Poulson, Laurie, Drellishak, Scott and Evans, Chris. 2007. Validation and Regression Testing for a Cross-linguistic Grammar Resource. In *ACL 2007 Workshop on Deep Linguistic Processing*, pages 136–143, Prague, Czech Republic: Association for Computational Linguistics.

Bond, Francis, Oepen, Stephan, Siegel, Melanie, Copestake, Ann and Flickinger, Dan. 2005. Open Source Machine Translation with DELPH-IN. In *Open-Source Machine Translation: Workshop at MT Summit X*, pages 15–22, Phuket.

Butt, Miriam, Dyvik, Helge, King, Tracy Holloway, Masuichi, Hiroshi and Rohrer, Christian. 2002. The Parallel Grammar Project. In John Carroll, Nelleke Oostdijk and Richard Sutcliffe (eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 1–7.

Callmeier, Ulrich. 2000. PET — A Platform for Experimentation with Efficient HPSG Processing Techniques. *Natural Language Engineering* 6 (1) (Special Issue on Efficient Processing with HPSG), 99 – 108.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

Copestake, Ann. 2002. *Implementing Typed Feature Structure Grammars*. Stanford, CA: CSLI Publications.

Copestake, Ann, Flickinger, Dan, Pollard, Carl and Sag, Ivan A. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language & Computation* 3(2–3), 281–332.

Drellishak, Scott. 2004. A Survey of Coordination in the World's Languages, MA thesis, University of Washington.

Drellishak, Scott and Bender, Emily M. 2005. A Coordination Module for a Crosslinguistic Grammar Resource. In Stefan Müller (ed.), *The Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar, Department of Informatics, University of Lisbon*, pages 108–128, Stanford: CSLI Publications.

Flickinger, Dan and Bender, Emily M. 2003. Compositional Semantics in a Multilingual Grammar Resource. In Emily M. Bender, Dan Flickinger, Frederik Fouvry and Melanie Siegel (eds.), *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Devel, ESSLLI 2003opment*, pages 33–42, Vienna, Austria.

Gordon, Raymond G., Jr. (ed.). 2005. *Ethnologue: Languages of the World*. Dallas, TX: SIL International, fifteenth edition, online version: http://www.ethnologue.com.

Hellan, Lars and Haugereid, Petter. 2003. NorSource: An Exercise in Matrix Grammar-Building Design. In Emily M. Bender, Dan Flickinger, Freerik Fouvry and Melanie Siegel (eds.), *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development, ESSLLI 2003*, pages 41–48, Vienna, Austria.

King, Tracy Holloway, Forst, Martin, Kuhn, Jonas and Butt, Miriam. 2005. The Feature Space in Parallel Grammar Writing. *Research on Language and Computation, Special Issue on Shared Representations in Multilingual Grammar Engineering* 3(2), 139–163.

Kinyon, Alexandra, Rambow, Owen, Scheffler, Tatjana, Yoon, SinWon and Joshi, Aravind K. 2006. The Metagrammar Goes Multilingual: A Cross-Linguistic Look at the V2-Phenomenon. In *In Proceedings of the Eighth International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+8)*, Sydney, Australia.

Kordoni, Valia and Neu, Julia. 2005. Deep Analysis of Modern Greek. In Keh-Yih Su, Jun'ichi Tsujii and Jong-Hyeok Lee (eds.), *Lecture Notes in Computer Science*, volume 3248, pages 674–683, Berlin: Springer-Verlag.

Marimon, Montserrat, Bel, Núria and Seghezzi, Natalia. 2007. Test-suite Construction for a Spanish Grammar. In Tracy Holloway King and Emily M. Bender (eds.), *Proceedings of the GEAF 2007 Workshop*, Stanford, CA: CSLI Publications.

Müller, Stefan. 2007. The Grammix CD-ROM: A Software Collection for Developing Typed Feature Structure Grammars. In Tracy Holloway King and Emily M.

Bender (eds.), *Proceedings of the GEAF 2007 Workshop*, Stanford, CA: CSLI Publications.

Newman, Paul. 2000. *The Hausa Language: an Encyclopedic Reference Grammar*. New Haven: Yale University Press.

Oepen, Stephan. 2002. *Competence and Performance Profiling for Constraint-based Grammars: A New Methodology, Toolkit, and Applications*. Ph. D.thesis, Universität des Saarlandes.

Oepen, Stephan, Dyvik, Helge, Lønning, Jan Tore, Velldal, Erik, Beermann, Dorothee, Carroll, John, Flickinger, Dan, Hellan, Lars, Johannessen, Janne Bondi, Meurer, Paul, Nordgård, Torbjørn and Rosén, Victoria. 2004a. Som å kapp-ete med trollet? Towards MRS-Based Norwegian – English Machine Translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*.

Oepen, Stephan, Flickinger, Daniel, Toutanova, Kristina and Manning, Christopher D. 2004b. LinGO Redwoods. A Rich and Dynamic Treebank for HPSG. *Journal of Research on Language and Computation* 2(4), 575 – 596.

Oepen, Stephan and Flickinger, Daniel P. 1998. Towards Systematic Grammar Profiling. Test Suite Technology Ten Years After. *Journal of Computer Speech and Language* 12 (4) (Special Issue on Evaluation), 411 – 436.

Pollard, Carl and Sag, Ivan A. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics, Chicago, IL and Stanford, CA: The University of Chicago Press and CSLI Publications.

Poulson, Laurie. 2006. Evaluating a cross-linguistic grammar model: Methodology and test-suite resource development, MA thesis, University of Washington.

Siegel, Melanie and Bender, Emily M. 2002. Efficient Deep Processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization at the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.

Sneddon, James Neil. 1996. *Indonesian: A Comprehensive Grammar*. London and New York: Routledge.

Soltan, Usama. 2006. Standard Arabic subject-verb agreement asymmetry revisited in an Agree-based minimalist syntax. In Cedric Boeckx (ed.), *Agreement Systems*, Amsterdam: John Benjamins.

Toutanova, Kristina, Manning, Chris and Oepen, Stephan. 2002. Parse Ranking for a Rich HPSG Grammar. In *Proceedings of The First Workshop on Treebanks and Linguistic Theories (TLT2002)*, Sozopol, Bulgaria.