Language COLLAGE: Grammatical Description with the LinGO Grammar Matrix

Emily M. Bender
University of Washington
Department of Linguistics
Box 352425
Seattle WA 98195-2425
ebender@uw.edu

Abstract

Language COLLAGE is a collection of grammatical descriptions developed in the context of a grammar engineering graduate course with the LinGO Grammar Matrix. These grammatical descriptions include testsuites in well-formed interlinear glossed text (IGT) format, high-level grammatical characterizations called 'choices files', HPSG grammar fragments (capable of parsing and generation), and documentation. As of this writing, Language COLLAGE includes resources for 52 typologically and areally diverse languages and this number is expected to grow over time. The resources for each language cover a similar range of core grammatical phenomena and are implemented in a uniform framework, compatible with the DELPH-IN suite of processing tools.

Keywords: precision grammar fragments; interlinear glossed text; typologically diverse dataset

1. Introduction

This paper presents Language COLLAGE (Collection of Language Lore Amassed through Grammar Engineering), a collection of grammatical descriptions of 52 languages (and counting) developed on the basis of the LinGO Grammar Matrix. It begins with a brief overview of the Grammar Matrix and its associated customization system, before explaining the provenance and contents of the descriptions in Language COLLAGE and how they will be distributed. Finally, I conclude with a discussion of potential use cases for this linguistic resource.

2. The LinGO Grammar Matrix

The LinGO Grammar Matrix (Bender et al., 2002; Bender et al., 2010) is an open-source online repository of grammatical analyses which facilitates the rapid development of linguistically-motivated deep grammars compatible with the DELPH-IN[1] suite of processing tools. It combines a core grammar providing constraints and structures which are crosslinguistically useful with a series of libraries of analyses of crosslinguistically variable phenomena.

Users access the Grammar Matrix through a web-based questionnaire[2] which elicits a high-level grammatical description of a language and then outputs a customized 'starter grammar' consistent with that description. The starter grammars are in the framework of HPSG (Pollard and Sag, 1994) and map strings to semantic representations in the format of Minimal Recursion Semantics (Copestake et al., 2005). They are encoded in the DELPH-IN joint reference formalism (Copestake, 2000), which is both machine- and human-readable; the starter grammars are functional (if small coverage) and ready for expansion to broader coverage.

The choices that the user makes in the customization system questionnaire are stored in a plain text 'choices' file. The Grammar Matrix customization system can be viewed as a function mapping from relatively simple grammatical descriptions (the choices files) to relatively complex ones (the grammars themselves). Furthermore, because the output of the system is working grammars, which can be used to parse and generate, the customization system supports iterative development of the descriptions, where users create initial versions, test them over sets of strings, and then refine the descriptions, retest and so on.

3. Linguistics 567

Annually since 2004, the LinGO Grammar Matrix has been used in Linguistics 567, a course on Grammar Engineering at the University of Washington (Bender, 2007). In this course, students develop a grammar fragment for a language on the basis of the Grammar Matrix, in two phases: First, over the course of three weeks, the students develop a testsuite documenting the phenomena their grammar will cover. This testsuite includes both grammatical and ungrammatical test items, each provided with morpheme-by-morpheme glosses as well as free translations. That is, the testsuites are collections of IGT (interlinear glossed text), and relatively clean collections, adhering closely to the Leipzig Glossing Rules (Bickel et al., 2008). In parallel, the students develop and refine a starter grammar through the Grammar Matrix customization system. The starter grammars cover only some of the phenomena illustrated in the testsuite. In the second phase, over six weeks, the students refine and extend their grammars by hand to cover more of the phenomena mapped out in the first phase. The extension of the grammars invariably also leads to further refinements of the testsuites.

For the most part, the students are not working on languages they speak or otherwise have expertise in. In the first week of the quarter, they identify reference grammars, and use these grammars to guide the development of both their testsuites and their grammars. Lab instructions guide the students to create testsuites based on the examples and

[1] http://www.delph-in.net

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

descriptions in their reference grammars. Students are instructed to record in the testsuite the provenance of each test item, i.e. a specific page reference for items drawn from the descriptive grammars or a notation indicating that the item was constructed by the student. The testsuites then guide the grammar development: In general, an analysis is suitable if it accounts for the available data (coverage and overgeneration), maps the grammatical strings to suitable semantic representations and will plausibly generalize to larger grammar fragments. The [incr tsdb()] grammar competence and performance profiling environment (Oepen and Flickinger, 1998) aids students in exploring the effects of different analyses and implementations of analyses on the behavior of the grammar (especially coverage, ambiguity, and overgeneration). The LKB grammar development environment (Copestake, 2002) provides tools for inspecting the structures produced by the grammar, and in particular the semantic representations which can be compared to the glosses for the examples in the testsuite. Finally, the course instructor provides further guidance about likely generalizability.

4. Language COLLAGE Contents

Since 2004, grammars have been developed for 92 languages, and as of this writing, I have permission from the grammar writers to distribute at least some of the types of resources described in this section for 52 of them. This sample of languages is typologically, genealogically and areally diverse, including both languages with large speaker populations (e.g., Mandarin [ISO 639-3 code: cmn], Malayalam [mal], Hausa [hau] and Thai [tha]) as well as languages spoken by far fewer (e.g., Breton [bre], Ingush [inh], Western Sissala [ssl], Penobscot [aaq-pen], and Hawai'ian [haw]). Linguistics 567 is offered annually, and Language COLLAGE will grow over time as additional languages are analyzed in this course.

Each language description in Language COLLAGE contains some (usually most or all) of the following:

- Testsuites
- Choices files
- Grammars
- Write ups
- Instructor feedback

As the grammars are developed over a series of 8 weekly assignments, many languages in Language COLLAGE include both intermediate and final versions of each resource type. The resource types are further described below.

4.1. Testsuites

The testsuites are constructed on the basis of reference materials, taking into account both the examples presented in the reference materials and the descriptions of those examples. In many cases, the reference materials do not include all of the example types that are required. For instance, a single sentence may be used to illustrate agreement between a verb and its arguments, but a testsuite for these purposes needs to include grammatical examples of all agreeing forms as well as ungrammatical examples illustrating agreement clash. If such examples are not available directly from their reference materials, students are instructed to construct them. Another situation which can lead students to construct examples is when the examples provided by the reference are too complicated. Testsuites for grammar engineering, especially in the context of small grammars, must consist of simple examples which minimize the interference of phenomena beyond those each example is meant to illustrate. Accordingly, students may create a simple single-clause example on the basis of a more complex item in the reference grammar, remove modifiers or make other simplifications.

```
# 114/G matrix yes no + neg
Source: b:151
Vetted: s
Judgment: g
Phenomena: q,tam,neg
waŋláke šni he?
waŋ-Ø-l-yáŋkA šni he
see-3SG.PAT-2SG.AGT-STEM NEG Q
'did you not see him?'
```

Figure 1: Sample testsuite item for Lakota [lkt]

Figure 1 provides an example from the Lakota [lkt] testsuite.[3] The first line is a comment describing the example. The next three lines are metadata fields showing the source for the example ('Source: b:151'[4]) which has not been further vetted with a native speaker but is presumed to have been vetted by the author of the source document ('Vetted: s') and which is intended as grammatical ('Judgment: g'). The 'Phenomena' line tags the example with the phenomena it is meant to illustrate (questions, tense/aspect/mood, negation). The remaining lines of the example are a detailed IGT representation, giving the example in a form without markup (in some languages, this will be the standard orthography), a version with morpheme boundaries marked and morphemes regularized to an underlying form, a morpheme-by-morpheme gloss and a translation into English. In languages with non-Latin-based orthographies, the language may be represented in the testsuite in transliteration. Similarly, in some cases students chose to replace non-ascii characters with ascii symbols. Both kinds of representation are supposed to be non-lossy.

The final testsuites in the collection so far range in size from a few tens of examples to a few hundred examples, usually about half of which are positive (grammatical) examples. These testsuites map out the grammatical territory that the students will attempt to cover throughout the course. In recent years (since 2012), students have also

[3]This example has been adapted for expository purposes in collaboration with Chris Curtis, one of the developers of the Lakota grammar.

[4]The tag 'b:151' indicates page 151 of grammar source b, specified elsewhere in the file as Ullrich (2011). Constructed examples are tagged with 'Source: author'.

collected small 'test corpora': 10-20 sentence samples of naturally occurring text, formatted as IGT. These are also included where they are available.

4.2. Choices files

Students begin grammar development by answering the Grammar Matrix customization system's web-based questionnaire. This questionnaire allows them to specify grammatical properties of the languages they are working on, including both high-level properties (e.g. major constituent word order or coordination strategies) as well as definitions of specific lexical classes and lexical rules for attaching affixes. Lexical rule definitions include both information about morphotactics (the order and co-occurrence restrictions between morphemes) and morphosyntax/morphosemantics, to the extent that the syntactic and semantic effects of each affix are among the phenomena modeled by the customization system.[5]

As the Grammar Matrix customization system has gotten more complex, covering more phenomena, the 'choices' files have also grown in size. The choices files from the 2013 iteration of the class range in size from 335 to 820 individual pieces of information in the grammatical descriptions, the bulk of which describe lexical classes and morphological rules. A portion of the choices file for Lakota is shown in Figure 2. This excerpt shows the high-level description of negation, which is marked with a single morpheme per clause that appears as an auxiliary, as well as the definition of a negative auxiliary.

```
section=sentential-negation
neg-exp=1
neg-aux=on
neg-aux-index=2
...
  aux2_name=neg
  aux2_sem=add-pred
    aux2_feat1_name=form
    aux2_feat1_value=negative-form
    aux2_feat1_head=verb
  aux2_subj=np
    aux2_compfeature1_name=form
    aux2_compfeature1_value=finite,\
                nonfinite, irrealis-form
    aux2_stem1_orth=šni
    aux2_stem1_pred=neg_rel
```

Figure 2: Choices file snippets for Lakota [lkt]

4.3. Grammars

The current course syllabus provides three weeks for testsuite and choices file development. This is intended to give students time to get as rich a starting point as possible from the customization system while also leaving plenty of course time to dedicate to hands-on grammar engineering. Once students finish with the customization system, they begin working with the grammar that the customization system produced for their final choices file, editing it directly in order to extend its coverage to phenomena not yet handled by the customization system and/or to refine the customization system-provided analyses to better fit their language.

Figure 3 provides a snippet of grammar source code from the Lakota grammar. This is the definition of the negative auxiliary that corresponds to the choices file specification shown in Figure 2. In this case, the code is preserved as-is from customization system output. Note that this particular type inherits much information from supertypes, defined elsewhere in the grammar.

A key benefit of using the Grammar Matrix as a starting point for grammar development is that it facilitates the creation of 'harmonized' semantic representations, i.e. representations that may not be identical across languages but which minimize spurious variation. Figure 4 shows the semantic representation produced by the Lakota grammar for the example in Figure 1, which involves the auxiliary defined in Figures 2 and 3. This representation uses English lemmas for the predicate names. This convention is followed throughout the resource not because English is in any sense suitable as an interlingua, but because it facilitates rapid grammar development by non-speakers. Should these grammars be built out to broader coverage, the predicate names should be related instead to same-language lemmas.

$$\langle\, h_1,\ \left|\begin{array}{l} h_3{:}\_\mathsf{see\_v}(e_2\{\mathsf{ASPECT}\ \textit{no-aspect}, \mathsf{SF}\ \textit{ques}\}, \\ \qquad\qquad x_4\{\mathsf{PERS}\ 2, \mathsf{NUM}\ \textit{sg}, \mathsf{COG\text{-}ST}\ \textit{in-foc}\}, \\ \qquad\qquad x_5\{\mathsf{PERS}\ 3, \mathsf{NUM}\ \textit{sg}, \mathsf{COG\text{-}ST}\ \textit{in-foc}\}) \\ h_6{:}\_\mathsf{neg\_r}(e_8, h_7) \end{array}\right| \\ \{\, h_7 =_q h_3 \,\} \,\rangle$$

Figure 4: MRS meaning representation for the example in Figure 1.

4.4. Write ups and instructor feedback

Each weekly lab assignment includes a written portion. The write up instructions elicit from students descriptions of the phenomena they are documenting in their testsuites and modeling with their choices files and/or hand-developed grammars. The write ups also include discussions of examples that are proving problematic. The instructor feedback includes suggested revisions to analyses as well as alternative analyses to explore.

4.5. Phenomena covered

Phenomena which are covered in the testsuites and grammars (and to a lesser extent the choices files) include:

- major constituent word order
- a small range of valence patterns
- case marking (if relevant)

[5] Morphophonological effects, however, are not modeled. The Grammar Matrix customization system produces grammar fragments intended to be paired eventually with morphophonological analyzers, and as such targets the regularized representation given in the morpheme-segmented line of the IGT.

```
neg-aux-lex := neg-subj-raise-aux-with-pred &
  [ SYNSEM.LOCAL [ CAT [ HEAD.FORM negative-form,
                         VAL.COMPS.FIRST.LOCAL [ CAT.HEAD.FORM nonfinite+irrealis-form,
                                                 CONT.HOOK.INDEX #index ] ],
   CONT.HOOK.INDEX #index ] ].
```

Figure 3: Grammar file snippet for Lakota [lkt]

- agreement (if relevant)
- coordination
- sentential negation
- marking of information structure
- pronouns
- dropped arguments
- main-clause yes-no questions
- clausal complements
- tense and aspect
- demonstratives and definiteness
- attributive adjectives
- non-verbal (NP, AP, PP) predicates

In addition, most grammars include coverage of some phenomena not included in that list but which either were critical to handling testsuite examples which could not be simplified further or which came up in the test corpora (for grammars from 2012 or later).

These grammars are of course small compared to what is required for broad-coverage precision parsing and generation. Though they individually lack *analytical breadth*, they do provide *analytical depth* (mapping to explicit semantic representations) and as a collection *typological breadth*. Furthermore, being built on a common, shared resource (the Grammar Matrix), and targeting a shared range of linguistic phenomena, they are interestingly interoperable.

5. Distribution

Language CoLLAGE is available free of charge, under the MIT license, from the project website:

`http://www.delph-in.net/matrix/language-collage/`

Users can download the whole collection or individual items of interest. In addition, the whole-collection download will be versioned, so that work building on this resource can point to a specific version of the Language CoLLAGE, to facilitate reproducibility.

6. Conclusion: Use Cases

A few different types of use cases for this data can be foreseen. A subset of the data (testsuites and choices files for some 30 languages) is already being used as training and test data by the AGGREGATION project, which seeks to learn to create grammars for low resource languages by automatically answering the Grammar Matrix customization system questionnaire on the basis of IGT data (Bender et al., 2013). The grammars are also of potential interest to other grammar developers, working on the same or related languages, as they allow grammarians to explore possible analyses and implementations of analyses of grammatical phenomena of interest. Finally, the grammars may be useful as a source of seeds for unsupervised approaches to parsing and/or syntax-based machine translation (Klein and Manning, 2004; Liu and Gildea, 2009).

7. Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0644097. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

The development of Language CoLLAGE would not be possible without the contributions of the students who developed its component grammars nor those of the Grammar Matrix developers who have contributed libraries to the customization system. The examples in this paper are taken from the Lakota resources included in Language CoLLAGE and contributed by Chris Curtis and David McHugh.

8. References

Bender, E. M., Flickinger, D., and Oepen, S. (2002). The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In Carroll, J., Oostdijk, N., and Sutcliffe, R., editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan.

Bender, E. M., Drellishak, S., Fokkens, A., Poulson, L., and Saleem, S. (2010). Grammar customization. *Research on Language & Computation*, pages 1–50. 10.1007/s11168-010-9070-1.

Bender, E. M., Goodman, M. W., Crowgey, J., and Xia, F. (2013). Towards creating precision grammars from interlinear glossed text: Inferring large-scale typological properties. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sci-*

ences, and Humanities, pages 74–83, Sofia, Bulgaria, August. Association for Computational Linguistics.

Bender, E. M. (2007). Combining research and pedagogy in the development of a crosslinguistic grammar resource. In King, T. H. and Bender, E. M., editors, *Proceedings of the GEAF 2007 Workshop*, Stanford, CA. CSLI Publications.

Bickel, B., Comrie, B., and Haspelmath, M. (2008). The Leipzig glossing rules: Conventions for interlinear morpheme-by-morpheme glosses. Max Planck Institute for Evolutionary Anthropology and Department of Linguistics, University of Leipzig.

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. A. (2005). Minimal recursion semantics: An introduction. *Research on Language & Computation*, 3(4):281–332.

Copestake, A. (2000). Appendix: Definitions of typed feature structures. *Natural Language Engineering*, 6:109–112.

Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, CA.

Klein, D. and Manning, C. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 478–485, Barcelona, Spain, July.

Liu, D. and Gildea, D. (2009). Bayesian learning of phrasal tree-to-string templates. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1308–1317, Singapore, August.

Oepen, S. and Flickinger, D. 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, C. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Ullrich, J. (2011). *New Lakota Dictionary*. Lakota Language Consortium, Bloomington IN, 2nd edition.