Abstract

I describe an analysis of valence-changing verbal morphology implemented as a library extending the LinGO Grammar Matrix customization system. This analysis is based on decomposition of these operations into rule components, which in turn are expressed as lexical rule supertypes that implement specific, isolatable constraints. I also show how common variations of these constraints can be abstracted and parameterized by their axes of variation. I then demonstrate how these constraints can be recomposed in various combinations to provide broad coverage of the typological variation of valence change found in the world's languages. I evaluate the coverage of this library on five held-out world languages that exhibit these phenomena, achieving 79% coverage and 2% overgeneration.

Introduction

In this paper, I describe a cross-linguistic analysis of valence-changing morphology that is implemented in a meta-grammar engineering system, the LinGO Grammar Matrix (Bender, Flickinger & Oepen, 2002)[1]. The core of the Grammar Matrix is a collection of implemented analyses for cross-linguistic phenomena, developed by linguists and grammar writers over many years, in a framework that provides infrastructure and context for reuse in development of precision grammars. The Grammar Matrix customization system (Bender, Drellishak, Fokkens, Poulson & Saleem, 2010) combines a structured means of eliciting typological characteristics, validating responses for consistency, and using those choices to combine Matrix core grammar elements with stored analyses of various linguistic phenomena into a customized grammar. The Grammar Matrix uses Minimal Recursion Semantics (MRS; Copestake, Flickinger, Pollard & Sag 2005) as its semantic representation. MRS can be naturally expressed in terms of feature structures and so is integrated into its HPSG mechanisms and feature structures.

One category of linguistic phenomenon not previously covered by the Grammar Matrix customization system is valence change: verbal morphology that alters the argument structure, either increasing or decreasing the valency, and changing the relationship of realized arguments to syntactic roles. To extend the customization system to include these operations, I developed a library that implements valence-changing operations that can be selected as part of a customized grammar. In building this library, I separated each high-level operation into foundational rule components, or "building blocks," which can then be composed as needed to implement valence change for a wide variety of the world's languages.

I first provide a brief typological survey of valence change (Section 1), followed by some examples of my analyses of these phenomena (Section 2); in particular, I illustrate the separation of these operations into rule components, as well as their re-composition into complete lexical rule types. I then describe my implementation of the library and evaluate its coverage of valence change variation (Sections 3-4).

[1] http://matrix.ling.washington.edu

1 Typology of valence change

I describe the cross-linguistic range of these operations below: following the broad conceptual framework provided by Haspelmath & Müller-Bardey (2004) (henceforth H&MB), operations are grouped by whether they reduce or increase valency and whether they affect the subject or object. I also retain their focus on verbal valence-changing morphology, excluding e.g., periphrastic constructions.

1.1 Valence-reducing operations

Both anticausative and passive constructions remove the subject and move the former object into the subject position; the essential distinction between them is that the anticausative removes the subject argument entirely, while the passive merely moves it to the periphery (H&MB). Analogous to the anticausative, the object-removing operation where the object is completely removed is referred to as the deobjective Haspelmath & Müller-Bardey (2004) or the absolutive antipassive (Dayley, 1989, as cited in H&MB). The deaccusative (H&MB) or antipassive (Dixon & Aikhenvald, 2000) is similar, but, instead of completely removing the underlying object argument, moves it out of the core to the periphery. The Turkish [tur] anticausative is illustrated in (1) as an example of a typical valence-reducing operation.

(1) a. Anne-m kapı-yı aç-tı
mother-1SG door-ACC open-PAST(3SG)
'My mother opened the door.' [tur]

b. Kapı aç-tı-dı
door open-ANTIC-PAST(3SG)
'The door opened.' [tur] (H&MB, p. 5)

1.2 Valence-increasing operations

Cross-linguistically the most common valence-changing category (Bybee, 1985), the causative adds a new subject, the causer of the event described by the verb. The addition of a causer to an intransitive verb can simply move the underlying subject into an object position. The situation with underlying transitive verbs is more complex, as there are different strategies for dealing with the underlying subject (causee), given the presence of an already-existing direct object. Other subject-adding constructions are structurally similar to the causative, such as the affective ('indirect passive') in Japanese [jpn]. A crucial aspect of the causative and similar constructions is the addition of a new elementary predication (EP) which functions as a scopal operator with respect to the verb's own EP and also takes as an argument the added participant.

Object-adding constructions can collectively be grouped under the term 'applicative,' which subsumes a broad variation in potential roles for the added structural argument. The prototypical applicative is the benefactive; however, in many languages (such as the Bantu family), applicatives can serve many additional functions, including locative, possessor-raising, and instrumental variations. Unlike the causative, the applicative is non-scopal. An example of object adding is presented below in (2).

As this brief survey highlights, there is a broad variety of specific valence-changing lexical operations in the world's languages, but strong threads of similarity also run through them. To cover this variety, I followed a "building block" approach in my analysis and implementation, as described in the following sections.

2 Analysis

The overall approach I followed was to decompose the high-level, linguistically-significant valence-changing operations into their component operations on feature structures. These individual component operations can then be selected by the customization system and composed to achieve the intended high-level result. The components I selected to analyze and implement included addition and removal of subjects and objects, case constraints and alternations, and argument reordering. In this section, as an illustrative example, I discuss my analysis of object addition, its breakdown into rule components, and the resulting effects on the feature structures.

Object addition covers the general category of the applicative, which subsumes a variety of different types of oblique roles cross-linguistically, including the instrumental and benefactive (H&MB, p. 7). In adding an argument, there are several underlying operations in my analysis: (a) adding an argument to the COMPS list;[2] (b) constraining the added argument, e.g. to be an np or pp (HEAD *noun* or *adp*), or applying a CASE constraint; (c) appending the new argument's non-local dependencies to the rule mother's list; (d) contributing an added EP (via C-CONT); (e) linking the new EP's first argument to the daughter's INDEX; and (f) linking the new EP's second argument to the new argument's INDEX.

The first two of these operations, (a) and (b), are directly grounded in the addition of a new argument and are straightforward; appending the new argument's non-local dependencies simply maintains the threading analysis of Bouma, Malouf & Sag (2001) and is similarly straightforward to motivate.

The addition of a new EP to the rule output is not as straightforward, and requires some additional discussion. To motivate this analysis, consider the example of the benefactive from Indonesian in (2). In this example, the addition of the benefactive applicative suffix *-kan* in (2b) adds an argument position to the verb, which is filled by *ibunja* "his mother".

[2]Note that, cross-linguistically, the added argument can be added either more- or less-obliquely to the verb's existing dependencies (i.e., at the head or tail of the COMPS list)

(2) a. Ali memi telefisi untuk ibu-nja
Ali TR.buy television for mother-his
'Ali bought a television for his mother.' [ind]

b. Ali mem-beli-kan ibu-nja telefisi
Ali TR-buy-APPL mother-his television
'Ali bought his mother a television.' [ind]
(Chung, 1976, in Wunderlich, 2015, p. 21)

Notionally, the benefactive is adding a third semantic argument to the verb, which would add a hypothetical third argument to the EP contributed by the verb;[3] however, this would seem to violate the principles of semantic composition in Copestake et al. 2005 for Minimal Recursion Semantics (MRS), namely, that composition consists soley of concatenation of daughter RELS values, not modification. More concretely, there is no EP-modifying operation available within the algebra of Copestake, Lascarides & Flickinger (2001).[4]

The solution is to have the lexical rule contribute a new EP, which takes both the EP contributed by the verb and the additional syntactic argument as semantic arguments. The predicate value for this new EP will provide the particular species of applicative (e.g., benefactive, as here). This new EP contributes its own event, and takes as its arguments the respective indexes of the input and the added argument. In this analysis, I treat the relationship between the added EP and the verb as non-scopal, with no intervening handle relationships. This contrasts with my analysis of subject addition; in my survey of valence change, the scopal relationship appears to only arise with added subjects.

The introduction of a new event by this EP differs from the analysis of the benefactive presented by Müller (2018, p. 68); my analysis here makes the relation contributed by the EP potentially available for modification separately from the event of the main verb (as in the English [eng] periphrastic form *Kim read the book, probably for Sandy*).

The MRS resulting from this analysis is shown in (3):

[3] Or a lexical rule which has previously been applied to the input.

[4] Although this principle is generally strongly embraced by DELPH-IN grammars, it is not entirely settled whether this necessarily should be as strictly applied within lexical rules (see e.g. Copestake, Lascarides & Bender, 2016, and Bender, 2015); it also is not prohibited by the DELPH-IN joint reference formalism (Copestake, 2002). My analyses in this work are not frustrated by adhering to this principle, so I retain it as applying throughout a grammar.

(3)
$$\begin{bmatrix} \text{RELS} & \left\langle \begin{array}{l} \begin{bmatrix} \textit{\_memi\_v\_rel} & \\ \text{ARG0} & \boxed{4}\ \textit{event} \\ \text{ARG1} & \boxed{1} \\ \text{ARG2} & \boxed{2} \end{bmatrix}, \begin{bmatrix} \textit{\_Ali\_n\_rel} & \\ \text{ARG0} & \boxed{1} \end{bmatrix}, \begin{bmatrix} \textit{\_telefisi\_n\_rel} & \\ \text{ARG0} & \boxed{2} \end{bmatrix}, \\ \begin{bmatrix} \textit{\_ibu\_n\_rel} & \\ \text{ARG0} & \boxed{3} \end{bmatrix}, \begin{bmatrix} \textit{benefactive\_rel} & \\ \text{ARG0} & \textit{event} \\ \text{ARG1} & \boxed{4} \\ \text{ARG2} & \boxed{3} \end{bmatrix} \end{array} \right\rangle \end{bmatrix}$$

With all these elements combined, a complete rule implementing the benefactive (with the argument being added less-obliquely, in this example) is illustrated in (4).

(4)
$$\begin{bmatrix} \textit{benefactive-lex-rule} & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL}|\text{CAT}|\text{VAL}|\text{COMPS} & \left\langle \boxed{1}, \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \\ \text{CONT}|\text{HOOK}|\text{INDEX} & \boxed{2} \end{bmatrix} \\ \text{NON-LOCAL} & \begin{bmatrix} \text{SLASH} & \boxed{3} \\ \text{QUE} & \boxed{4} \\ \text{REL} & \boxed{5} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{NON-LOCAL} & \begin{bmatrix} \text{SLASH} & \boxed{7} \oplus \boxed{3} \\ \text{QUE} & \boxed{8} \oplus \boxed{4} \\ \text{REL} & \boxed{9} \oplus \boxed{5} \end{bmatrix} \end{bmatrix} \\ \text{C-CONT} & \begin{bmatrix} \text{RELS} & \left\langle ! \begin{bmatrix} \textit{event-relation} & \\ \text{PRED} & \textit{``benefactive\_rel''} \\ \text{ARG1} & \boxed{6} \\ \text{ARG2} & \boxed{2} \end{bmatrix} ! \right\rangle \end{bmatrix} \\ \text{DTR} & \begin{bmatrix} \textit{verb-lex} & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT}|\text{VAL}|\text{COMPS} & \boxed{1} \\ \text{CONT}|\text{HOOK}|\text{INDEX} & \boxed{6} \end{bmatrix} \\ \text{NON-LOCAL} & \begin{bmatrix} \text{SLASH} & \boxed{7} \\ \text{QUE} & \boxed{8} \\ \text{REL} & \boxed{9} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

This rule, however, in combining the distinct operations identified above, obscures common elements that can be reused for other similar object-adding operations. Reviewing the five operations, it is evident that they vary along several different axes, as shown in Table 1.

This leads to a simplification and optimization: these building-block operations can be treated as being parameterized along their their axes of variation, and then

rule component	varies by
added argument	position (obliqueness), number of existing args
constraint on new argument	position (obliqueness), constraint (*e.g.* case, head)
non-local dependencies	position (obliqueness)
new EP's pred value	predicate
new EP's arg1	does not vary
new EP's arg2	position (obliqueness)

Table 1: Rule components

combined to make the final rule type. Concretely, taking these operations in turn, the first operation (adding the argument) needs to have variants for adding an argument: (a) to intransitive or transitive verbs, and (b) at the front or end of the COMPS list. That is, the lexical rule type implementing each of the component operations can be viewed as the output of a function: $f : tr \in \{intrans, trans\} \times pos \in \{front, end\} \rightarrow lrt$. To illustrate one variation, the rule type at (5) adds an argument at the *end* of the COMPS list for an *intransitive* verb,[5] and the rule at the rule type shown in (6) adds an argument at the front of the COMPS list for a transitive verb and links the INDEX of that argument to its second semantic argument (ARG2).

(5)
$$\begin{bmatrix} \textit{added-arg2of2-lex-rule} & \\ \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{VAL}|\text{COMPS} & \left\langle \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT}|\text{VAL} & \begin{bmatrix} \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{CONT}|\text{HOOK}|\text{INDEX} & \boxed{1} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{C-CONT}|\text{RELS} & \langle! \begin{bmatrix} \text{ARG2} & \boxed{1} \end{bmatrix} !\rangle \\ \text{DTR}|\text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{VAL}|\text{COMPS} & \langle\ \rangle \end{bmatrix}$$

(6)
$$\begin{bmatrix} \textit{added-arg2of3-lex-rule} & \\ \text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{VAL}|\text{COMPS} & \left\langle \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT}|\text{VAL} & \begin{bmatrix} \text{SPR} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \\ \text{CONT}|\text{HOOK}|\text{INDEX} & \boxed{1} \end{bmatrix} \end{bmatrix}, \boxed{2} \right\rangle \\ \text{C-CONT}|\text{RELS} & \langle! \begin{bmatrix} \text{ARG2} & \boxed{1} \end{bmatrix} !\rangle \\ \text{DTR}|\text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{VAL}|\text{COMPS} & \langle \boxed{2} \rangle \end{bmatrix}$$

In a similar fashion, constraining the head type of the added argument can be isolated to an individual rule supertype, as in (7):

[5]Naturally, for an intransitive input there is no difference between the front and end of the comps list. The same rule would be generated for either specification; formally, $f(intrans, front) \equiv f(intrans, end)$.

(7)
$$\begin{bmatrix} \textit{added-arg2-np-head-lex-rule} \\ \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CAT}\,|\,\text{VAL}\,|\,\text{COMPS} \quad \left\langle \begin{bmatrix}\text{LOCAL}\,|\,\text{CAT}\,|\,\text{HEAD} \quad \textit{noun}\end{bmatrix}, [\ ] \right\rangle \end{bmatrix}$$

The non-local dependencies carried by the added argument, as analyzed in the threading analysis of Bouma et al. (2001), are implemented in the Grammar Matrix as difference-list append operations. This is normally handled in the Grammar Matrix by a lexical type's mapping from argument structure to valence lists, with the additional difference-list append constraints provided via inheriting the appropriate lexical supertype (*basic-one-arg*, *basic-two-arg*, etc.). As this analysis operates outside these existing mechanisms, an additional constraint, parameterized on the position of the added argument, must be added to perform these appends.[6] An example of this operation is shown in (8):

(8)
$$\begin{bmatrix} \textit{added-arg2of3-non-local-lex-rule} & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL}\,|\,\text{CAT}\,|\,\text{VAL}\,|\,\text{COMPS} & \left\langle \begin{bmatrix} \text{NON-LOCAL} & \begin{bmatrix} \text{SLASH} & \boxed{4} \\ \text{REL} & \boxed{5} \\ \text{QUE} & \boxed{6} \end{bmatrix} \end{bmatrix}, [\ ] \right\rangle \\ \text{NON-LOCAL} & \begin{bmatrix} \text{SLASH} & \boxed{1} \oplus \boxed{4} \\ \text{REL} & \boxed{2} \oplus \boxed{5} \\ \text{QUE} & \boxed{3} \oplus \boxed{6} \end{bmatrix} \end{bmatrix} \\ \text{DTR}\,|\,\text{SYNSEM}\,|\,\text{NON-LOCAL} & \begin{bmatrix} \text{SLASH} & \boxed{1} \\ \text{REL} & \boxed{2} \\ \text{QUE} & \boxed{3} \end{bmatrix} \end{bmatrix}$$

The most variable, individualized component is the predicate (pred value) of the added semantic relation. For example, the benefactive and instrumental rules may be entirely common in structure, but would need to be distinguished by their predicate. A separate rule supertype, therefore, can be created as in (9) to constrain the PRED value appropriately.

(9)
$$\begin{bmatrix} \textit{benefactive-pred-lex-rule} \\ \text{C-CONT}\,|\,\text{RELS} \quad \langle !\ \begin{bmatrix}\text{PRED} \quad \textit{“benefactive\_rel”}\end{bmatrix}\ ! \rangle \end{bmatrix}$$

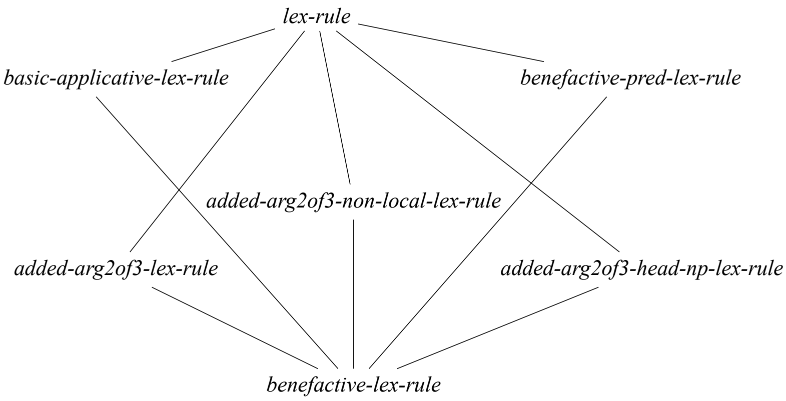
Finally, the (invariant) core of the "generic" applicative can be isolated and analyzed as in (10). The function of this rule supertype is to contribute the new predication, and link its ARG1 to the daughter's intrinsic variable (i.e., the ARG0).

[6] I have arbitrarily selected the ordering here of the added non-local dependencies; this analysis may need to be refined in the event that order of non-local dependency satisfaction becomes relevant.

(10)
$$\begin{bmatrix} \textit{basic-applicative-lex-rule} & \\ \text{C-CONT} & \begin{bmatrix} \text{RELS} & \left\langle !\ \begin{bmatrix} \textit{event-relation} \\ \text{ARG1} \ \boxed{1} \end{bmatrix}\ ! \right\rangle \end{bmatrix} \\ \text{DTR} \mid \text{SYNSEM} \mid \text{LOCAL} \mid \text{CONT} \mid \text{HOOK} \mid \text{INDEX} \ \boxed{1} & \end{bmatrix}$$

These building-block rule component supertypes can then be assembled as inherited constraints on a complete applicative rule type, ready to be instantiated in a grammar. The partial inheritance tree showing these rule component supertypes for the original example full rule type in (4) is illustrated below in (11):

(11)



In the case of a subject-adding operation, such as the causative illustrated from Georgian [kat] in (12), I treat the added ("causing") EP as a scopal predicate: it outscopes the underlying verb's EP and so provides the HOOKfeature values for the entire VP. The resulting MRS should be as shown in (13).

(12) Mama-m Mzia-s daanteb-in-a cecxli
father-ERG Mzia-DAT light-CAUS-AOR:3SG fire(ABS)
'Father made Mzia light the fire.' [kat] (Harris, 1981, in H&MB, p. 12)

(13)
$$\begin{bmatrix} \text{HOOK} & \begin{bmatrix} \text{LTOP} & \boxed{5} \\ \text{INDEX} & \boxed{6} \\ \text{XARG} & \boxed{3} \end{bmatrix} \\ \text{RELS} & \left\langle \begin{bmatrix} \textit{\_daanteb\_v\_light} & \\ \text{LBL} & \boxed{8} \\ \text{ARG0} & \boxed{4}\ \textit{event} \\ \text{ARG1} & \boxed{1} \\ \text{ARG2} & \boxed{2} \end{bmatrix}, \begin{bmatrix} \textit{causative} & \\ \text{LBL} & \boxed{5} \\ \text{ARG0} & \boxed{6}\ \textit{event} \\ \text{ARG1} & \boxed{3} \\ \text{ARG2} & \boxed{1} \\ \text{ARG3} & \boxed{7} \end{bmatrix}, \begin{bmatrix} \textit{\_mama\_n\_father} & \\ \text{ARG0} & \boxed{3} \end{bmatrix}, \begin{bmatrix} \textit{named} & \\ \text{ARG0} & \boxed{1} \end{bmatrix}, \begin{bmatrix} \textit{\_cecxli\_n\_fire} & \\ \text{ARG0} & \boxed{2} \end{bmatrix} \right\rangle \\ \text{HCONS} & \left\langle !\ \begin{bmatrix} \textit{qeq} & \\ \text{HARG} & \boxed{7} \\ \text{LARG} & \boxed{8} \end{bmatrix}\ ! \right\rangle \end{bmatrix}$$

Note that, consistent with the strategy in Copestake et al. (2001), the scopal relationship is expressed by a handle constraint (hcons) rather than directly, representing equality modulo quantifiers ($=_q$). This handle constraint predicts that quantifiers can scope in between the EPs of the causative and embedded verb.

Similarly to my analysis of the applicative, the causative can also be decomposed into component operations, again parameterized along the axes of cross-linguistic variation. In the next section, I describe how these analyses were added to the Grammar Matrix.

3 Implementation in the Grammar Matrix

The Grammar Matrix customization system (Bender et al., 2010) combines a structured means of eliciting typological characteristics, validating responses for consistency, and using those choices to combine Matrix core grammar elements with stored analyses of various linguistic phenomena into a customized grammar. These stored analyses can include both static representations of cross-linguistically common phenomena as well as dynamically-generated implementations that embody language-specific variations. Elicitation is accomplished via a dynamic, iteratively-generated HTML questionnaire, which records the responses (while validating the consistency of both individual responses and their combination) in a structured choices file. This choices file is then processed by the customization script to produce the customized grammar.

My implementation of a library leverages the existing morphotactics machinery in the customization system (Goodman, 2013) by adding options to the questionnaire for grammar writers to attach valence-changing operations to lexical rule types, along with the relevant parameters (e.g., position of erstwhile subject) necessary to generate the operations. My extensions to the grammar customization scripts, in turn, uses the selections in the choices file to generate the appropriate parameterized and common rule components, and then combine them into types to be instantiated.

While developing the library, two types of tests were used. Initially, I developed small, abstract pseudolanguages to exercise specific operations and combinations; I then attempted to model valence change in three natural languages, Lakota [lkt], Japanese [jpn], and Zulu [zul], and produced test suites of grammatical and ungrammatical examples. During this phase of development, I continued to revise my analyses and code to achieve full coverage of the examples. Once this phase was complete, I then froze library development and moved to the evaluation phase, described in the next section.

4 Evaluation

To evaluate the library as developed against a representative sample of the world's languages, I selected five held-out languages, from different familial and areal groups, that had not been used during development. Two languages were selected from descriptive articles intentionally held out, and the rest were selected by drawing randomly from a large collection of descriptive grammars, discarding those without valence changing morphology, until sufficient evaluation languages were collected.

I created test suites for each of these languages consisting of grammatical and ungrammatical examples of valence change, and attempted to model the corresponding phenomena using only the facilities available in the customization system questionnaire. I then attempted to parse the test suites using the customization system-generated grammars and recorded which grammatical examples were correctly parsed, which ungrammatical examples were erroneously parsed, and to what extent the parses generated spurious ambiguity. These results[7] are summarized in Table 2.

On the test suites for the five held-out languages, this approach as implemented in my library achieved an overall coverage of 79% and an aggregate overgeneration rate of only 2%. The language with the poorest coverage (55%), Rawang [raw], suffered almost entirely due to a relatively rich system of reflexive and middle constructions; my library lacked the ability to fill a valence slot while coindexing with an existing argument and so these examples could not be modeled. The sole example of overgeneration, from Javanese [jav], was similarly due to the inability of the current library to apply a HEAD constraint to an already-existing argument. Neither of these failures appear to be particularly difficult to add to the library, which would significantly improve its flexibility and applicability.

[7] None of the test suites generated spurious ambiguity.

		examples		performance		
Language	Family	pos	neg	parses	coverage	overgen.
Tsez [ddo]	NE Caucasian	11	8	10	91%	0%
West Greenlandic [kal]	Eskimo-Aleut	15	14	12	73%	0%
Awa Pit [kwi]	Barbacoan	7	7	5	71%	0%
Rawang [raw]	Sino-Tibetan	11	6	6	55%	0%
Javanese [jav]	Austronesian	13	8	12	92%	13%
Total		57	43	45	79%	2%

Table 2: Test languages test summary and performance

5 Conclusion

In this work I have presented an HPSG analysis of valence-changing verbal morphology, implemented in the LinGO Grammar Matrix, which I evaluated against several held-out languages. The results appear to support the hypothesis that a "building-block" based approach is an effective way to provide significant typological coverage of valence change. By developing and implementing this analysis within the larger Grammar Matrix project, these elements of valence change can be combined and recombined in different ways to test linguistic hypotheses and compare modeling choices, including the interactions of valence change with other phenomena. Although the scope of this work was limited to valence change expressed through verbal morphology, future work might include determining whether this approach can be extended to other phenomena, including, for example, periphrastic valence-changing constructions.

References

Bender, Emily M. 2015. (Un-)orthodoxy in use of HOOK features, semantic algebra compliance. MRS representation issues [discussion], presented at the Eleventh DELPH-IN Summit. `http://moin.delph-in.net/SingaporeSemanticAlgebraCompliance`.

Bender, Emily M., Scott Drellishak, Antske Fokkens, Laurie Poulson & Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation* 8(1). 23–72.

Bender, Emily M., Dan Flickinger & Stephan Oepen. 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 & Richard Sutcliffe (eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 8–14. Taipei, Taiwan.

Bouma, Gosse, Robert Malouf & Ivan A. Sag. 2001. Satisfying constraints on extraction and adjunction. *Natural Language & Linguistic Theory* 19(1). 1–65.

Bybee, Joan L. 1985. *Morphology: A study of the relation between meaning and form*. Amsterdam: John Benjamins Publishing.

Chung, Sandra. 1976. An object-creating rule in Bahasa Indonesia. *Linguistic Inquiry* 7. 41–87.

Copestake, Ann. 2002. Definitions of typed feature structures. In Stephan Oepen, Dan Flickinger, Jun-ichi Tsujii & Hans Uszkoreit (eds.), *Collaborative language engineering*, 227–230. Stanford, CA: CSLI Publications.

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

Copestake, Ann, Alex Lascarides & Emily M. Bender. 2016. Algebra replacement. Revising the algebra [discussion], presented at the Twelth DELPH-IN Summit. `http://moin.delph-in.net/StanfordAlgebraAdditions`.

Copestake, Ann, Alex Lascarides & Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* ACL '01, 140–147. Stroudsburg, PA, USA: Association for Computational Linguistics.

Dayley, Jon Philip. 1989. *Tümpisa (Panamint) Shoshone grammar*, vol. 115. Univ of California Press.

Dixon, R. M. W. & Alexandra Y. Aikhenvald. 2000. *Changing valency*. Cambridge University Press.

Goodman, Michael W. 2013. Generation of machine-readable morphological rules from human-readable input. *University of Washington Working Papers in Linguistics* 30.

Harris, Alice C. 1981. *Georgian syntax: A study in relational grammar*. Cambridge: Cambridge University Press.

Haspelmath, Martin & Thomas Müller-Bardey. 2004. Valence change. *Morphology: A handbook on inflection and word formation* 2. 1130–1145.

Müller, Stefan. 2018. *A lexicalist account of argument structure: Template-based phrasal LFG approaches and a lexical HPSG alternative* (Conceptual Foundations of Language Science 2). Berlin: Language Science Press.

Wunderlich, Dieter. 2015. Valency-changing word-formation. In Peter O. Müller, Ingeborg Ohnheiser, Susan Olsen & Franz Rainer (eds.), *Word-formation*, vol. 3, 1424–1466. Berlin/Boston: De Gruyter Mouton.