Modeling Light Verb Constructions in the LinGO Grammar Matrix

Tara Wueger

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2025

Committee:

Emily Bender

Andrew Hedding

Program Authorized to Offer Degree:

Linguistics

University of Washington

Abstract

Modeling Light Verb Constructions
in the LinGO Grammar Matrix

Tara Wueger

Chair of the Supervisory Committee:
Emily Bender
Linguistics

This thesis describes the development of a library for modeling light verb constructions (LVCs) in the LinGO Grammar Matrix. LVCs are constructions involving the combination of a light verb and coverb, where the coverb contributes most of the meaning to the whole construction. Light verbs can range in meaning from semantically "light" to semantically bleached and coverbs can come from a variety of word classes (e.g. noun, verb, adjective). The syntactic and semantic representations of LVCs that make up the foundation of my analysis that are presented in this paper are done within the HPSG and MRS formalisms. I then implemented this analysis in the Grammar Matrix customization system, using illustrative and pseudo languages to do so (including Bardi, English, Japanese, and Persion). I also evaluated the library using held-out languages (Ch'ol, Daasanach, and Korafe-Yegha) in order to test its generalizability across different languages. The results of this evaluation include an average of ∼90% coverage, ∼25% overgeneration, and ∼2.0 ambiguity across the testsuites for all three held-out languages.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

1	first person
2	second person
3	third person
A	A-form (for Daasanach)
A	Set A (ergative) (for Ch'ol)
ABL	ablative
ABS	absolutive
ACC	accusative
ACT	actor-subject, contrastive
ALL	allative
ANTIP	antipassive
APPL	applicative
AQ	indicative assertion, information question
B	B-form (for Daasanach)
COMP	complementizer
CONT	continuous
DAT	dative
DECL	declarative
DEF	definite
DEM	demonstrative
DET	determiner
DO	direct object
DTV	derived transitive verb
ERG	ergative
F	feminine
FIN	finite
FUT	future
GEN	genitive
GER	gerund
HEST	hesternal (yesterday's past)
I	set I (for Korafe-Yegha)
II	set II (for Korafe-Yegha)
IMP	imperative
INDF	indefinite
INF	infinitive
INS	instrumental
IO	indirect object
IPFV	imperfective
IRR	irrealis
M	masculine
MIN	minimal

NEG	negative
NF	non-finite suffix
NOM	nominative
OBJ	object
OBL	oblique
PFV	perfective
PL	plural
POSS	possessive
PRF	perfect
PROG	progressive
PROX	proximal
PRS	present
PST	past
PTCP	participle
REC	recent
REDUP	reduplication
REL	relative
REM	remote
RS	realis subject
SBJV	subjunctive
SG	singular
THUS	causal subordinator
TOP	topic
TR	transitive
TS	transitive suffix

ACKNOWLEDGMENTS

First, I want to thank my advisor Emily M. Bender for all of her support and advice throughout the process of this thesis. I would not have been able to get this far without her (or her nearly 500 revision comments). The final draft of this paper is a far cry from the first, and I owe it to her tireless work in helping me describe my findings clearly, concisely, and accurately. She is an incredible advisor to have and I am a better researcher and writer because of her.

Next, I want to thank my reader Andrew Hedding for his valuable feedback, especially for being willing to dedicate time towards reviewing this paper while on leave. I also want to express my thanks to the members of DELPH-IN who gave me feedback in the early stages of this project. I am grateful to all of the people in the EMB students group who helped with debugging, writing advice, and general motivation. Additionally, this work could not have been completed without the work of countless linguists, whose analyses, descriptions, and examples I relied on in order to complete this project.

Finally, I want to thank my friends and family who have supported me not only through this thesis but the CLMS program as well. I especially want to give appreciation to my sister Kira, who not only had to deal with me dragging her to countless coffee shops while I wrote this paper but even helped proofread parts of it.

Chapter 1

INTRODUCTION

In this thesis, I describe the addition of a light verb construction library to the LinGO Grammar Matrix customization system (Bender, Flickinger & Oepen, 2002; Bender et al., 2010; Zamaraeva et al., 2022). This system is used to help users create grammars in the Head-driven Phrase Structure (HPSG) formalism (Pollard & Sag, 1994) based on the information about a language's phenomena that a user provides. Users provide this information through a online questionnaire, the responses to which are then used to generate the HPSG grammar. HPSG is the primary formalism for the syntactic representations in the grammar, while Minimal Recursion Semantics (MRS) is used for semantic representations. The Grammar Matrix provides a series of libraries, where each one represents an analysis of a different linguistic phenomenon. Each library is designed to give the user choices in order to model a phenomenon that is represented in the language they are implementing. One of the goals of the Grammar Matrix is to allow for faster development of machine-readable grammars for any language through the use of resource grammars or other linguistic descriptions. These are precision grammars, which have broad coverage for the phenomena that have been implemented.

My motivation behind choosing this topic for my thesis stems from my passion for working with languages that are often under-represented in natural language processing (NLP), which aligns with the goals of the Grammar Matrix. I believe that the importance of a language is not dependent on how many people speak it or how much data exists for it. I discovered light verb constructions (LVCs) by working on a grammar for a language (Bardi) that has only a few hundred speakers and relies heavily on LVCs to form sentences. At the time, the Grammar Matrix did not have a way to analyze these sentences, which limited parsing coverage for the language. Bardi became one of the primary languages that I used during the development of this library. My goal in developing this library is to expand the phenomena that the Grammar Matric covers and to encourage other grammar engineers interested in adding to the Grammar Matrix to pursue phenomena no matter how linguistically widespread they are.

Although LVCs do exist in many different languages, some languages use them more than others. In some languages, LVCs are used to productively create new predicates, especially in those where verbs are a closed class. In other languages, LVCs are a way to incorporate loanwords. LVCs are both a syntactic and semantic phenomenon, relying on both linguistic structural levels to be properly represented. This is why HPSG and MRS are good candidates for an analysis of the phenomenon. The LVC library that I developed accounts for noun and verb coverbs,[1] which can optionally take dependents and can appear before and/or after the light verb. It also allows for intransitive and transitive valence options and some semantic variability for the light verb (either full or bleached lexical meaning). Lastly, the library accounts for restrictions on whether elements are allowed between the coverb and light verb.

In order to evaluate this library, I tested how it performed on languages that were not used during development (known as held-out languages). The three languages that I selected for these tests are Ch'ol, Daasanach, and Korafe-Yegha. The main metrics that were used to do the evaluation were coverage, overgeneration, and ambiguity.[2] The LVC library performed fairly well in these metrics, with an average of ~90% coverage, ~25% overgeneration, and ~2.0 ambiguity across all three languages. I performed an error analysis, in order to diagnose any remaining issues with my implementation. I was able to fix some of them post-evaluation while others were left to future work.

In this thesis, I begin by providing some background on the formalisms that my analysis rely on and on the system that my implementation is adding to, which is done in chapter 2. In chapter 3, I provide an overview to the literature that exists on the LVC phenomenon,

[1]Coverbs are defined in section 3.1.1.

[2]These are defined in section 5.5.

which explains some of the typological variation that exists cross-linguistically. Next, in chapter 4, I describe my analysis of LVCs (within the HPSG and MRS formalisms) and how they were implemented into the customization system. Lastly, I explain the testing and evaluation process for the LVC library, which as done using a variety of different languages.

Chapter 2

LITERATURE REVIEW: FRAMEWORKS

In this chapter, I describe the syntactic and semantic formalisms that comprise the foundation of the analysis I present in this thesis. Head-driven Phrase Structure Grammar is used for syntactic representations (section 2.1) and minimal recursion semantics is used for semantic representations (section 2.2). I also explain what the LinGO Grammar Matrix is and how it produces a grammar, which uses these formalisms (section 2.3).

2.1 Head-driven Phrase Structure Grammar

Head-driven Phrase Structure Grammar (HPSG) is a phrase structure grammar framework that was developed by Carl Pollard and Ivan Sag (Pollard & Sag, 1994). It has close ties with computer science, as there are multiple computational implementations of this framework (with the Grammar Matrix being one of them). HPSG is implemented in the typed feature structure formalism. These feature structures are most commonly represented using attribute-value-matrixes (AVMs). A **typed feature structure** consists of features and their values, which can be simple strings, types with no further structure inside, or typed feature structures. Features are written in small caps (e.g. TENSE) and string values are written in italics (e.g. *prs*), as shown in the AVM in (1)[1].

(1) [TENSE *prs*]

As mentioned above, these feature structures are **typed**, which is a way of organizing or classifying aspects of languages. For example, words are commonly organized by their part of speech, which in HPSG is represented by the HEAD feature. Additional information

[1]Unless otherwise specified, all feature structures shown in this section are hypothetical and are not from a real grammar.

about the verb can also be added via other features. A feature structure for a word that is an intransitive verb could look like (2) (where the empty COMPS list indicates that the verb doesn't take any complements).

(2) $$\begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

There are four main kinds of feature structures that are relevant to this thesis: lexical entries, lexical types, lexical rule types, and phrasal types. A **lexical entry** is a representation of an individual lexical item (a word). A lexicon is made up of lexical entries, which are represented by a form (the orthographical representation of the word) and a feature structure. For example, the English verb *eat* could have the lexical entry shown in (3).

(3) $$\left\langle \text{eat}, \begin{bmatrix} \textit{word} & \\ \text{HEAD} & \textit{verb} \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \right\rangle$$

A **lexical type** is a representation of a group of lexical entries. It contains features that are common across a set of lexical entries. Lexical entries will inherit from a lexical type. Lexical types can also inherit from other lexical types (called supertypes) and can inherit from more than one supertype. Using the type hierarchy in this way allows for more generalizable grammars, since features do not have to be manually specified for each lexical entry or type. For example, it might be a good idea to classify verbs based on valence frames (e.g. intransitive or transitive). A possible feature structure for an intransitive verb, called *intransitive-verb-lex*, is shown in (4).[2] Instead of manually specifying that the English verb *eat* has an empty COMPS list on its lexical entry (as in (3)), this lexical entry (and any other intransitive verb lexical entry) could instead inherit from *intransitive-verb-lex*.

(4) $$\begin{bmatrix} \textit{intransitive-verb-lex} & \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{CAT} \mid \text{VAL} \mid \text{COMPS} & \langle\ \rangle \end{bmatrix}$$

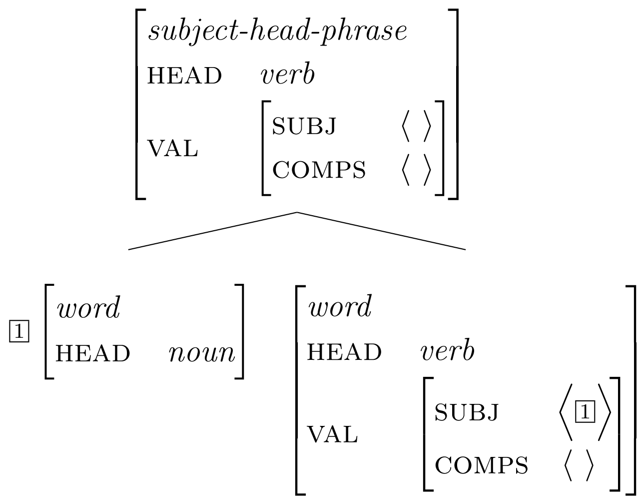
[2] This is part of a lexical type from the English grammar I developed using the Grammar Matrix customization system for this thesis. The *plural-lex-rule* on the next page is also from this grammar.

A **lexical rule type** relates a stem, which can be just a root or a root in addition to other lexical rule applications, to a fully inflected form. Inflectional lexical rules are those where the addition of an affix adds information about a grammatical feature. For example, the suffix *-s* can be used to make a noun plural in English. If the *-s* suffix is attached to the nominal root *hedgehog*, the result would be the plural noun *hedgehogs*. A lexical rule type for this would need to specify that the NUM feature on the noun is *pl*, as shown in (5). Non-inflectional lexical rules are very similar to the inflectional ones, except they would not associate a specific affix with the rule.

(5) $\begin{bmatrix} \textit{plural-lex-rule} & \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{CONT} \mid \text{HOOK} \mid \text{INDEX} \mid \text{PNG} \mid \text{NUM} & \textit{pl} \end{bmatrix}$

A **phrasal type** is a representation of a grammar rule, which indicates how smaller constituents can combine, in order to form phrases or sentences. Grammar rules in HPSG generally consist of a head daughter and a non-head daughter. The HEAD feature is always carried up from the head daughter in headed phrases while, for other features, it depends on what is specified in the rule. For example, in order to form the sentence *They eat*, the subject *they* needs to combine with the verb *eat*. This would be done with the subject-head rule. In the example of a simplified subject-head rule in (6), the verb identifies its subject with the noun (represented by the $\boxed{1}$). The resulting unified feature structure is subject empty and has the same HEAD value as the head daughter (*verb*). **Unification** here refers to the process of successfully combining two feature structures. Unification fails if there are conflicting feature values. For example, two verbs (e.g. *create eat*) would not be able to unify using the subject-head rule since the subject-head rule requires the non-head daughter to unify with the information in the SUBJ list of the head daughter, and the verb eat expects its subject (*create*) to have a head type of *noun*.

(6)



The concepts discussed in this section are core aspects of the Grammar Matrix and the grammars it produces. The HPSG feature structures discussed here focus on syntactic information. For semantic information, the Grammar Matrix relies on minimal recursion semantics, which will be described in the next section.

2.2 Minimal Recursion Semantics

Minimal recursion semantics (MRS) is the framework that the Grammar Matrix uses to implement semantic representations (Copestake et al., 2005). It is used for computational semantics and can be incorporated into HPSG feature structures, which means that it also uses types and the process of unification, except to implement semantic representations instead of syntactic ones.

MRS representations rely on a bag of predications or relations, which are stored in a feature known as the RELS list (Flickinger, Bender & Oepen, 2003). Each item on this list is represented by a predication value, a handle h (which is stored in the LBL feature), and its arguments (which are stored in numbered ARG features). Arguments can be handles h, events e, instances x, or underspecified i between events and instances. Events are typically associated with verbal constituents and contain information like tense, aspect, and mood. Instances are typically associated with nominal constituents and contain information like person and number. An example of the predication for the transitive verb *eat* is shown in (7), where it has the handle *h0*, an event *e2*, and *x2* is associated with the subject while *x3* is associated with the object.

(7) $$\begin{bmatrix} \textit{\_eat\_v\_rel} & \\ \text{LBL} & \textit{h0} \\ \text{ARG0} & \textit{e1} \\ \text{ARG1} & \textit{x2} \\ \text{ARG2} & \textit{x3} \end{bmatrix}$$

The handle is also used to describe partially underspecified scope relations, which are stored the HCONS list feature. Each item on this list is an *equal modulo quantifier* (*qeq*) relation and consists of a higher argument (HARG) and a lower argument (LARG). The lower argument contains the label for a constituent and connects it with its scopal quantifier (the higher argument). For example, in *a bath*, the handle of *bath* (*h4* in (8)) is the value for LARG while the restrictor (RSTR) value of *a* (*h2* in (8)) is the value for HARG. Both RSTR and BODY are scopal features, with RSTR relating to the top handle of the quantifier's restriction and BODY being left unbound to allow for variable scope possibilities (Flickinger, Bender & Oepen, 2003, p. 20). There are two more parts of the MRS that are important to discuss: the local top handle (LTOP) and the index (INDEX). The value of LTOP is a handle for the entire MRS. The value of INDEX is, in most cases, the value for ARG0 of the primary relation for the constituent's syntactic head (for a noun phrase, it would be an instance while for a verb phrase it would be an event). Both LTOP and INDEX are available for further composition as the constituent combines with other constituents up the tree. The full MRS for *a bath* is shown in (8).

(8)
$$\begin{bmatrix} \textsc{ltop} & h7 \\ \textsc{index} & x1 \\ \textsc{rels} & \left\langle \begin{bmatrix} \_a\_q\_rel & \\ \textsc{lbl} & h0 \\ \textsc{arg0} & x1 \\ \textsc{rstr} & h2 \\ \textsc{body} & h3 \end{bmatrix}, \begin{bmatrix} \_bath\_n\_rel & \\ \textsc{lbl} & h4 \\ \textsc{arg0} & x1 \end{bmatrix} \right\rangle \\ \textsc{hcons} & \left\langle \begin{bmatrix} qeq & \\ \textsc{harg} & h2 \\ \textsc{larg} & h4 \end{bmatrix} \right\rangle \end{bmatrix}$$

With the addition of MRSes to the HPSG feature structures, the Grammar Matrix grammars are able to model syntactic and semantic structure. What the Grammar Matrix is and how it produces grammars that rely on the HPSG and MRS formalisms will be explained in the next section.

2.3 The LinGO Grammar Matrix

The LinGO Grammar Matrix and its customization system (Bender, Flickinger & Oepen, 2002; Bender et al., 2010; Zamaraeva et al., 2022) are a tool that allows for the development of machine-readable grammars that are precise and provide broad coverage. The Grammar Matrix customization system serves as a starting point for implemented grammars for any language, which are HPSG grammars that use MRS to model the semantics. To a user, the Grammar Matrix customization system is a questionnaire that is filled out to produce a `choices` file, which can subsequently be used to generate a grammar. The questionnaire is made up of front-facing subpages that correspond to libraries, which represent the analysis and implementation of various cross-linguistic phenomena. Filling out the questionnaire is a process of making choices that model a language's rules and properties. Once the user fills out the parts of the questionnaire applicable to their language, they can create a `choices` file. However, before the file can be created, a number of validation checks are made on their responses in the questionnaire, which help to ensure that the grammars output by the customization system compile with the processing engines (e.g. the LKB) that use them. The `choices` file is a plain text file that contains the user's responses to the questionnaire and that is used by the customization system to systematically produce a grammar. This is the back-end part of the Grammar Matrix customization system that the user doesn't see. The grammar consists of a collection of files, many of which have the `.tdl` file extension and use Type Description Language (TDL) syntax (Krieger & Schafer, 1994). Some of more important files are `matrix.tdl` (which contains type definitions considered to be useful to all languages[3]), `{lang_name}.tdl` (which contains language-specific type definitions that inherit from the types defined in `matrix.tdl`), `lexicon.tdl` (which defines lexical entries), `irules.tdl` (which defines inflecting lexical rules), `lrules.tdl` (which defines non-inflecting lexical rules), and `rules.tdl` (which defines phrase structure rules). Once a grammar has been produced, software such as the the Linguistic Knowledge Builder (LKB) (Copestake, 2002) and the Answer Constraint Engine (ACE) (Crysmann & Packard, 2012) can be used to parse and generate sentences. Parsing is used to determine if a sentence is grammatical and to assign syntactic and semantic representations to the sentence while generation is used to create sentences based on their syntactic and semantic representations. Additionally, the `[incr tsdb()]` (Oepen, Netter & Klein, 1997; Oepen & Flickinger, 1998; Oepen, 2001; Oepen & Callmeier, 2000; Oepen & Carroll, 2000a,b) software can be used to analyze testsuites of sentences and is particularly useful when testing how a grammar performs throughout various stages of its development.

The state of the Grammar Matrix now is very different from what it was when the project was started. At the beginning (Bender, Flickinger & Oepen, 2002), it was just a core grammar. The customization system was added later (Bender & Flickinger, 2005; Drellishak, 2009). Over the years, grammar engineers have developed numerous libraries that expanded the phenomena that the Grammar Matrix is able to cover. Some of these include coordination (Drellishak & Bender, 2005), morphotactics (O'Hara, 2008; Goodman, 2013),

[3]This file will be the same for every grammar output by the Grammar Matrix.

argument optionality (Saleem, 2010), negation (Crowgey, 2012), evidentiality (Haeger, 2017), adnominal possession (Nielsen, 2018), and nominalization (Howell, Zamaraeva & Bender, 2018; Ruditsky, 2024). The light verb construction library interacts the most with the morphotactics library, since both light verbs and coverbs can undergo lexical rules.[4]

2.4 Summary

In this chapter, I described the fundamentals of how HPSG and MRS work to provide syntactic and semantic representations of language. I also explained what the Grammar Matrix is as well as how it works and how it is used. Next, I will give some background on what light verb constructions are and how they have been analyzed cross-linguistically.

[4] Interactions with the other libraries were not fully explored but they are possible.

Chapter 3

LITERATURE REVIEW: LVCS

In this chapter, I explore the existing literature on light verb constructions. In section 3.1, I go over terminology concerning light verb constructions and how they are discussed in the literature. In sections 3.2 and 3.3, I explore morphological, syntactic, and semantic properties of light verb constructions. Additionally, in section 3.4 I look at possible ways of representing LVCs in different grammar frameworks and, in section 3.5, I provide possible tests for determining whether or not a construction is a light verb construction.

3.1 Terminology

In this section, I define what a light verb construction (LVC) is and how this definition is situated within the literature. Light verb constructions have been referred to by many other names in the literature, some of which are not accurate. I clarify the terms and explain why they can or cannot be used to refer to LVCs. A summary of these terms can be found in Table 3.1.

Complex/Composite Predicate	Not a Complex Predicate
LVC	compound verb/verbal compound
conjunct predicate	auxiliary verb construction
compound verb/verbal compound	
SVC	

Table 3.1: Summary of Terminology

3.1.1 What is a Light Verb Construction?

For the purpose of this thesis, I define a **light verb construction** as a type of complex predicate comprised of a light verb and a coverb, where most (but not necessarily all) of the lexical meaning of the combined complex predicate comes from the coverb. A **light verb** is one that is semantically ‘light’ to some degree (Butt, 2010, p. 48), ranging from contributing no lexical meaning (i.e. semantically bleached), to some but never all of the lexical meaning to the construction. The other element in the LVC is the **coverb** (Butt, 2010; Wilson, 1999), which is a noun, verb, or adjective that could be used as its intended part of speech in the language and/or could be used to provide some or all of the lexical meaning in an LVC. This element has commonly been referred to as a **preverb** (Bowern, 2012, pp. 503, 776; Osgarby & Bowern, 2023, p. 292). However, the term “preverb” implies that this element comes before the light verb, which is not always the case cross-linguistically, while “coverb” alludes to this element co-occurring with the light verb. Therefore, I have chosen the latter to refer to this element. Examples of LVCs (from Bardi, Persian, and English, respectively) can be found in (9) below, where the light verb is in **bold** and the coverb is in *italics*.

(9) a. Garrinngan daab innyagal.

garrin-ngan	*daab*	**i-n-nya-gal**
hill-ALL	*go.up.to*	3-TR-**catch**-REC.PST

‘He went up the hill.’ [bcj] (Bowern, 2012, p. 227)

b. Maryam bāqčerā āb dād.

Maryam	bāqče=rā	*āb*	**dād**
Maryam	garden=DO	*water*	**give**.PST

‘Maryam watered the garden.’ [per] (Godard & Samvelian, 2021, p. 472)

c. She took a bath.

she	**took**	a	*bath*
3SG.F	**take**.PST	DET.INDF	*bath*

‘She took a bath.’ [eng]

3.1.2 Phenomena Associated with LVCs and Related Terminology

Conjunct predicate has been used (Abdullah, Ahmed, Anjum, et al., 2021) to refer to some LVCs, but this refers specifically to constructions with noun or adjective coverbs, while LVCs also include verb coverbs. Since the LVC library accounts for noun and verb coverbs, LVC is more appropriate than conjunct predicate. Although **compound verb** (or verb(al) compound) has previously been used to describe the same construction as an LVC (Thieberger, 2006, p. 224; Haspelmath, 1993, p. 178), it is not an apt description of the phenomenon as the two predicates in the LVC are not always forming a compound (Butt, 2010, p. 49). The process of compounding does not correctly describe the relationship between all light verbs and coverbs and, cross-linguistically, compounding does not need to involve a "light" element. In the South Efate example in (10), the two verbs *fis* "whip" and *ktof* "break" are forming a compound but whether the relationship between the two verbs is that of a light verb and coverb is unclear. More generally speaking, my library does not account for LVCs where the light verb and coverb are not free morphemes so I will not use this terminology in my analysis. Furthermore, I will leave the question as to whether some or all compounds are light verb constructions open.[1]

(10) Me tewan ifisktofi boy.

me	tewan	i=fis-ktof-i-∅	boy
and	so	3SG.RS=whip-break-TS-3SG.OBJ	boy

'And so he whipped the boy.' [erk] (Thieberger, 2006, p. 225)

Serial verb constructions (SVCs) are similar to LVCs but are distinct (Butt, 2010, p. 49). Although SVCs and LVCs are different constructions, clearly defining how they are different is not simple. In cases where the coverb in the LVC is a noun, no distinction is necessary. However, when the coverb in the LVC is a verb, the construction can be confused with an SVC. Seiss, Butt & King (2009) determine that no cross-linguistic properties can be

[1]For this reason, compound verb/verbal compound is included on both sides of table 3.1.

defined that distinguish SVCs and LVCs and that languages must be looked at individually in order to determine the properties which distinguish one from the other (pp. 515-516). Part of this difficulty stems from issues with generalizing cross-linguistic properties of SVCs, as described in Haspelmath (2016), who outlines generalizations but with exceptions. Examples of SVCs (from Edo and Alamblak, respectively) can be found in (11) below.

(11) a. Òzó sàán rrá ógbà.

Òzó	sàán	rrá	ógbà
Ozo	jump	cross	fence

'Ozo jumped across the fence.' (Lit. 'Ozo jumped (he) crossed the fence.') [bin] (Haspelmath, 2016, p. 294)

b. Mɨyt ritm muhhambrayanm.

mɨyt	ritm	muh-hambray-an-m
tree	insect	climb-search.for-1SG-3PL

'I climbed the tree searching for insects.' (Lit. 'I climbed the tree I searched for insects.') [amp] (Haspelmath, 2016, p. 294)

Osgarby & Bowern (2023) do attempt to distinguish between LVCs and SVCs, specifically for Australian languages. However, their definition of an LVC only accounts for constructions with nominal coverbs. This does not fully align with my analysis of LVCs, and therefore the implementation of the LVC library, since verbal coverbs are also possible. According to them, both of the predicating elements in an SVC are from the same word class and can act as verbal predicates independently. In an LVC, the light verb is from "a restricted class of predicating verbs" and needs the coverb to form the predicate (Osgarby & Bowern, 2023, p. 292). They compare complex predicates[2] from other constructions across different categories: composition, marking, contiguity, wordhood, valency, Aktionsart, vector configuration, and lexical semantics. For some of these categories, they specifically distinguish between SVCs and LVCs. The categories relevant to my analysis of LVCs are discussed here.

[2]They consider LVCs and SVCs to be kinds of complex predicates.

In terms of composition, while SVCs can be symmetrical (meaning that predicating elements come from unrestricted sets) or asymmetrical (meaning that at least one of the predicated elements comes from a semantically or grammatically restricted set), LVCs can only ever by asymmetrical (since the light verb always comes from a restricted set). In my analysis, light verbs are entered into the lexicon separately from other verbs and do form what could be considered a restricted set. As for marking, Osgarby & Bowern (2023) state that, in LVCs, only the light verb can be marked for verbal features, while, in other complex predicates, both predicating elements in the construction can be marked for these features. In my analysis, I leave this possibility open, as Grammar Matrix users may wish to model constructions where both light verb and coverb bear verbal features as LVCs. Contiguity has to do with whether the two predicating elements need to be directly next to each other and Osgarby & Bowern (2023) use this property to distinguish between complex predicates and two independent predicating elements. If there is an intervening element between the two predicating elements, they are independent. This also does not align with my analysis, which allows LVCs with subjects or objects intervening between the light verb and coverb (for languages where this is possible, like Persian, which is discussed in section 5.1.4). Overall, the LVC library follows an analysis that aligns with the definition stated in section 3.1.1. Grammar Matrix users can use the LVC library if my analysis aligns with the phenomenon they are trying to model. If an SVC library is added in the future, the person implementing it will need to make certain decisions on how to formally distinguish LVCs and SVCs.

In the literature, LVCs, and complex predicates in general, have been classified as **auxiliary verb constructions**. However, the auxiliary verb in an auxiliary verb construction, which is analogous to the light verb in an LVC, only expresses functional or grammatical meaning (e.g. tense or aspect) and no lexical meaning. As mentioned previously, while it is possible for a light verb to be semantically bleached, it often does contribute some lexical meaning. Moreover, from a diachronic perspective, pairs of light verbs and auxiliary verbs can be traced back to originating from the same main verb, since light verbs stay identical in form to the original main verb, while auxiliary verbs undergo phonological change (Butt, 2010, pp. 53, 68–69). Examples of auxiliary verb constructions (from Urdu and French, respectively) can be found in (12) below.

(12) a. Vo so rahii t^{h}ii.

vo	so	rah-ii	t^{h}-ii
3SG.NOM	sleep	PROG-F.SG	be.PST-F.SG

‘She was sleeping.’ [urd] (Butt, 2010, p. 63)

b. Paul a rapidement lu son livre.

Paul	a	rapidement	lu	son	livre
Paul	has	quickly	read.PST	3SG.POSS	book

‘Paul has quickly read his book.’ [fra] (Godard & Samvelian, 2021, p. 11)

Complex predicate (or composite predicate) is one of the most common ways to refer to LVCs, but is better used as an umbrella term for constructions like those involving light verbs, serial verbs, and conjunct predicates (Osgarby & Bowern, 2023). Butt (2010) defines a complex predicate as “a construction that involves two or more predicational elements (such as nouns, verbs, and adjectives) which predicate as a single unit” (p. 49). Alsina, Bresnan & Sells (1997) define complex predicates as “composed of more than one grammatical element (either morphemes or words), each of which contributes a non-trivial part of the information of the complex predicate” (p. 1).

3.1.3 Summary

In this section, I defined what an LVC is and the different terminology relating to it. To reiterate, an LVC consists of a semantically “light” verb and a coverb, where some or all of the lexical meaning of the combined complex predicate comes from the coverb. The light verb can contribute to the meaning of the construction but does not need to. I also explained how LVCs differ from other similar or related constructions. Constructions like LVCs, SVCs, and conjunct predicates are types of complex predicates while auxiliary verb constructions are not. Constructions with compound verbs could either be complex predicates or not, and that would depend on how that compound is being formed and on the relationship between the two elements in the compound. In future sections, I will refer to these kinds of constructions as LVCs — consisting of a light verb and a coverb — even when the authors I am citing use different terminology.

3.2 LVC Categorization and Properties

In this section, I introduce some approaches to categorizing light verbs constructions (specifically, light verb and coverb combinations). I also describe some cross-linguistic and language-specific properties of light verbs and coverbs.

3.2.1 LVC Categorization

When it comes to categorizing[3] LVCs, the most common approach is to do so based on the light verb (Bowern, 2012; Butt, 2010; McGregor, 2002), although exactly how this is done varies. Generally speaking, this means that coverbs, or groups of coverbs, are listed for each light verb they can combine with. The coverbs are usually grouped based on some common property (or properties). The descriptions below are important as they informed how I implemented the way that users add light verbs and coverbs to the lexicon. In the lexicon, for each lexical type that is a coverb, the user can input which light verb (or light verbs) it can combine with. This allows users to build groups of coverbs similar to how a linguist might have done.

Bowern (2012) splits light verbs into either monovalent or bivalent (there are exceptions for the light verb *joo* “to do/say”, which has both monovalent and bivalent readings, and for unproductive LVCs, where categorization doesn’t make sense). Since light verbs are a small, closed class in Bardi, Bowern (2012) details groups of coverbs for each light verb, which are categorized based on some property of the resulting LVC. For example, for the bivalent light verb *ma* “to put”, she creates categories based on whether the resulting LVC

[3]Bowern (2012) refers to this as “classifying.”

is transitive, causative, or intransitive. For the transitive and intransitive ones, she further groups light verb-coverb combinations as having to do with properties such as change of position, location, emotion, and body functions (Bowern, 2012, pp. 542–546).

McGregor (2002), on the other hand, doesn't specifically categorize Warrwa LVCs, but rather groups clauses in general into six transitivity types: intransitive, transitive, middle1, middle2, quasitransitive, and ditransitive. He further groups each type based on semantic domains (e.g. perception, emotions, communication, movement, or generic action). Many of his examples of clauses are LVCs and these groupings have parallels to the LVC categories in Bowern (2012).

3.2.2 Light Verb Properties

The purpose of this section is to give an overview of the properties of light verbs in order to show the range of variation in these properties across languages. The discussions of these properties influenced my analysis and how I implemented the LVC library. The properties discussed in the literature include role in the construction, argument structure, thematic properties, argument licensing, semantic defectiveness, and event predication.

According to Alba-Salas (2002, p. 64), a light verb's main role is to give the coverb and its complements syntactic structure and to carry any necessary inflectional morphology (e.g. tense, aspect, mood, person, number, etc.). He asserts that while linguists tend to agree that light verbs have what is called a 'defective' argument structure (meaning the argument structure is empty or incomplete), they disagree on the exact thematic properties of light verbs (Alba-Salas, 2002, p. 52). One of the main points of contradiction involves the subject licensing. More specifically, whether light verbs can select for specific subjects or not and what the conditions affecting such selection are. Another point involves complement licensing — whether light verbs can license their own complements and impose selection restrictions independent of the coverb (Alba-Salas, 2002, p. 9). In my analysis, argument licensing (of the light verb on both the subject and complements other than the coverb) is not restricted in any way. However, the user is able to add features that could restrict which arguments a light verb can license. The last point of contention is the degree of semantic defectiveness. Some linguists argue that light verbs cannot be truly semantically empty and that they do have some lexical content. This has led to discussion about what makes a verb light versus heavy, with some linguists arguing that there is a clear-cut distinction (although that distinction varies cross-linguistically), while others claim that this distinction is too ambiguous to be clearly ascertained (Alba-Salas, 2002, pp. 68–70). In my analysis, the user has the option of defining light verbs that contribute to the semantics and/or light verbs that don't. Other variability in the range of how much the light verb contributes to the semantics is not implemented.

When it comes to how light verbs contribute to event predications, there are two main approaches. The first is that light verbs introduce their own event predication (Bowern, 2012, p. 504). The other approach is that light verbs just add information to a pre-existing event predication. This means that light verbs in LVCs must rely on some other predicative element and, instead of contributing their own event, they supply information about the event, for example volition or valence (Butt, 2010, p. 72). In my analysis, light verbs that are not semantically empty introduce their own event. In constructions with semantically empty light verbs, the coverb contributes the event.

3.2.3 Coverb Properties

In his dissertation, Alba-Salas (2002) makes a list of generalizations about light verb constructions in Romance languages,[4] some of which are properties of the coverb[5] (pp. 57-59). However, he only looked at nominal coverbs, and not verbal ones, so these generalizations might not apply to coverbs across the board. That being said, he does include Japanese and Korean LVCs in his analysis and the coverbs for both of these languages are usually considered to be verbal. This section will focus on the generalizations he makes and coverb

[4]Although his focus is on Romance languages, he does include Japanese, Korean, and Telugu examples and how they fit into the generalizations he makes.

[5]Or, as he calls it, the verbal noun.

properties that he discusses. The properties discussed here include argument licensing and case-marking. Similar to the light verb properties in the last section, the descriptions of these properties informed my analysis of LVCs. Since there is such wide variation, my analysis of coverbs does not restrict them very much and instead leaves it up to the user to further restrict them if needed. In my analysis, coverbs can license their own complements and cannot license the arguments of the entire light verb construction. Only the light verb can license these arguments (e.g. subject or direct object). This does not fully align with the descriptions presented below, as, for some languages, coverbs can license these arguments. Additionally, coverbs can influence the transitivity of the light verb construction, which my analysis does handle.

One of the generalizations that Alba-Salas (2002, p. 23) makes is that coverbs cannot have their own genitive-marked subject, which he shows cross-linguistically.[6] He provides examples in Italian and Korean, which can be found in (13) below (Alba-Salas, 2002, p. 23). In the Italian example, the coverb *telefonata* "phone call" cannot have the genitive subject *di Paulo* "Paul's" because the agent of "calling" is coreferential with *Gianni*, which is the subject of the light verb. The same thing is shown in the Korean example, where the coverb *kongkyek* "attack" cannot have the genitive subject *Roma-uy* "of the Romans". The coverb shares its subject with the light verb, therefore it can't have its own. My analysis does not allow coverbs to have their own subjects, genitive-marked or otherwise.

(13) a. Gianni ha fatto una telefonata **(*di Paolo)** a Maria.

Gianni	ha	fatto	una	telefonata	(*di	Paolo)	a	Maria
Gianni	have.3SG.PRS	do.PST.PTCP	a	call	of	Paul	to	Mary

'Gianni made a/**(*Paul's)** phone call to Mary.' [ita]

b. Wulika Sabineytayhan **(*Romauy)** kongkyekul hayssta.

wuli-ka	Sabin-eytayhan	(*Roma-uy)	kongkyek-ul	hay-ss-ta
we-NOM	Sabines-OBL	Romans-GEN	attack-ACC	do-PST-DECL

[6] Alba-Salas (2002) only ever discusses genitive marked subjects but coverbs generally cannot have their own subjects.

‘We made an attack on the Sabines **(*by the Romans)**.’ [kor]

Another generalization Alba-Salas (2002) makes is based on the distinction between coverbs that can be marked with accusative case (what he calls unincorporated) and those that are zero-marked (what he calls incorporated). For example, Korean, Japanese, and Telugu coverbs allow both, but there are restrictions on incorporation for the latter two languages[7] (Alba-Salas, 2002, p. 57). On the other hand, in Romance languages, English, and Basque, coverbs cannot be incorporated. It is important to note that Alba-Salas (2002) relies heavily on the distinction between languages that allow this incorporation and languages that don't when it comes to many of the properties he describes for coverbs (as well as for light verbs and LVCs in general). My analysis does not rely on this distinction, therefore, I do not go into depth about all of the properties and generalizations he describes that are based on it. Additionally, the Grammar Matrix allows for different features to be specified on different lexical entries and the user can choose how to model properties described for a language using these features. In other words, a user can choose to model coverb incorporation if their language has it.

One property of a coverb is that it can impose a selectional restriction on the surface subject of the light verb (Alba-Salas, 2002, p. 21). This can be seen in the French example (in (14) below) where the sentence using *fait un effet* “does an effect” is grammatical, while the one with *fait une enquête* “does an inquiry” is ungrammatical (Alba-Salas, 2002, p. 21). This is because *effet* can license an inanimate, non-agentive subject (e.g. *resultat* “result”) while *enquête* cannot. In my analysis, users are only able to add restrictions on the subject through features on the light verb.[8]

[7]These restrictions are that two-place coverbs cannot be marked for accusative case when they occur with another argument that is accusatively marked and that unnacusative coverbs and unnacusative uses of transitive coverbs must be incorporated as well (Alba-Salas, 2002, pp. 57–58).

[8]Future work could make this possible by copying any features on a coverb to the light verb.

(14) a. Ce resultat fait **un effet** sur l'opinion.

ce	resultat	fait	un	effet	sur	l'opinion
DEM.PROX.SG	result	do.3SG.PRS	DET.INDF	effect	on	DET.DEF.opinion

'This result has an effect on public opinion.' [fra]

b. *Ce resultat fait **une enquête** sur l'opinion.

ce	resultat	fait	une	enquête	sur	l'opinion
DEM.PROX.SG	result	do.3SG.PRS	DET.INDF	inquiry	on	DET.DEF.opinion

'This result makes an inquiry into public opinion.' [fra]

The complement of the coverb can be realized as the direct dependent of the light verb or the coverb, the constraints for which vary cross-linguistically. For example, this can depend on whether a language does or doesn't allow coverb incorporation (Alba-Salas, 2002, p. 24). My analysis does not allow for complements of the coverb to be licensed by the light verb.[9]

Properties of non-subject arguments of the light verb can vary depending on the coverb being used. For example, in Japanese, when the light verb *suru* "to do" is combined with different coverbs, it can select a different number of complements (Alba-Salas, 2002, p. 21; Bowern, 2010, p. 43; Grimshaw & Mester, 1988, p. 207). In (15) below, the coverbs *hanashi* "talk", *shuppatsu* "departure", and *keikoku* "warn" lead to a different number of non-subject arguments being selected: zero (intransitive), one (transitive), and two (ditransitive), respectively (Grimshaw & Mester, 1988, p. 207). In my analysis, the user can specify how many complements (other than the coverb) the light verb can license. Currently, only intransitive and transitive light verbs are implemented. If the user wanted to model the coverb influencing the transitivity, they would need to define two separate light verbs (with different transitivity options). Then, they would need to specify which light verb each coverb combines with, choosing the light verb with the correct transitivity option.

[9]Future work could look into implementing this.

(15) a. Intransitive[10]

John-wa Mary-ni hanashi-o shi-ta
John-TOP Mary-DAT talk-ACC do-PST

'John talked to Mary.' [jpn]

b. Transitive[11]

John-wa Tokyo-kara shuppatsu-o shi-ta
John-TOP Tokyo-ABL departure-ACC do-PST

'John departed from Tokyo.' [jpn]

c. Ditransitive

John-wa murabito-ni okami-ga kuru-to keikoku-o shi-ta
John-TOP villager-DAT wolf-NOM come-COMP warn-ACC do-PST

'John warned the villagers that the wolf was coming.' [jpn]

3.2.4 Summary

This section focused on how linguists have approached categorizing LVCs and the properties associated with light verbs and coverbs. Some of these are based on semantic properties, which will be the focus of the next section.

3.3 General Semantics

In this section, I give an overview of the literature on the semantics of LVCs. I start by explaining why multi-word expressions in general are difficult for natural language processing technology to handle, which stems primarily from the semantics of LVCs. I also discuss the semantic compositionality of LVCs.

---

[10] Based on the description of this sentence, I assume that *Maryni* "to Mary" is an adjunct.

[11] Based on the description of this sentence, I assume that *Tokyo-kara* "from Tokyo" is a complement.

3.3.1 Why are MWEs Difficult for NLP?

Generally speaking, multiword expressions (MWEs) are difficult to represent in a way that natural language processing (NLP) technology can use (Sag et al., 2002). This relates to issues in generation (overgeneration) and analysis/parsing (idiomaticity), both of which revolve around word meanings (i.e. semantics). **Overgeneration** occurs because not every word in a group (e.g. part of speech or word class) can be interchanged within a MWE. NLP technology sometimes allows MWE combinations that are compositionally correct but are semantically unacceptable. For example, English has the light verb *take*, which can only take some nouns as coverbs — *take a seat* refers to the action of sitting down while *take a chair* is not acceptable in a similar sense. Though *seat* and *chair* are similar nouns, *chair* cannot function as a coverb. **Idiomaticity**[12] is used to refer to the notion that the meaning of an expression is not always a sum of its parts. For example, Bardi has the light verb *-(i)nya-* meaning "to catch" and the coverb *liyan* meaning "heart" and, when combined into the LVC, *liyan -(i)nya-*, it means "to breathe" (Bowern, 2012, p. 508).

One of the most common approaches to handling both overgeneration and idiomaticity problems involves treating each expression as one word (**words-with-spaces**), however, this approach leads to issues with flexibility and lexical proliferation (Sag et al., 2002). The words in an MWE might not always be directly next to each other. For example, in English, the LVC *took the train* can be modified with adjectives (e.g. *took the last train*) or with a different determiner (e.g. *took a train*). The lack of flexibility in the words-with-spaces approach cannot account for cases like these. The lexical proliferation problem revolves around predictability and generality. Light verb constructions can often be grouped together or follow a template. An example of this are cases where a light verb takes a group of coverbs to form the construction and the resulting meaning follows a pattern. The English light verb *give* exhibits this with *give a bath*, *give a kiss*, *give a shout*. Treating these kinds of expressions as a single word means that neither generalizations nor predictions can be made

[12] Megerdoomian, 2004 also talks about this issue in section 2.1.

based on a pattern.

3.3.2 Compositionality... or Lack Thereof

Compositionality is the principle that "the meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined" (Partee, 1984). For LVCs, the meaning of an expression comes from a combination of the light verb, the coverb, and inflecting morphology on the light verb (Bowern, 2012, pp. 507–514). This can also affect semantic properties such as semantic roles, temporal or aspectual information, and Aktionsart. Compositionality can often be an issue for idiomatic expressions, where the meaning of the whole expression is not a sum of its parts. This also applies to idiomatic LVCs, which exist in many of the languages that have LVCs. In this section, I will give an overview of some of the ways that LVCs are compositional, in addition to discussing the ways in which idiomatic LVCs are non-compositional.

Light verbs cannot contribute temporal or aspectual information or, in other words, the semantics or meaning of the light verb itself cannot provide this information (this is another example of a feature that distinguishes LVCs from auxiliary verb constructions) (Butt, 2010, p. 72). For example, one use of the auxiliary verb *will* in English has habitual aspect meaning — and no other meaning — as in *He will be late every day.* This contrasts with light verbs, which usually have some non-grammatical meaning, and can additionally express temporal or aspectual information through inflectional marking. Some cross-linguistically common glosses for light verbs include "do" (Bardi *-ji-*; Persian *kardan*; Urdu *kar*; Korean *ha*), "give" (Bardi -∅-; Persian *dādan*; Urdu *de*), "go" (Bardi *-jiidi-*; Jaminjung *jga*; Urdu *jaa*), and "hit" (Bardi *-boo-*; Persian *zadan*; Jamingjung *ma*; Urdu *maar*) (examples from Butt, 2010; Bowern, 2012; Abdullah, Ahmed, Anjum, et al., 2021; Bonami & Samvelian, 2010; Kim, Kim & Yang, 2007). Light verbs can also contribute to the Aktionsart of the LVC. For example, in Jaminjung, the coverb *bul* "appear" can be combined with the light verbs *ruma* "come" and *ma* "hit" however, *bul -rama-* means "appear" while *bul -ma-* means "appear (suddenly)" (Butt, 2010, p. 60).

The semantic/thematic roles for a sentence containing an LVC are determined based on all parts of the LVC (the light verb, coverb, and derivational morphology on the light verb) as well (Bowern, 2012, p. 510). In (16)–(19) below, examples of how each of these parts can influence semantic roles in Bardi are given (Bowern, 2012, pp. 510–511). (16) shows how adding the applicative suffix *-nginji* adds an accompanier role to an LVC that originally only had an agent role. (17) shows that the coverb *abrarrabarr* "confuse", when combined with the the light verb *ma* "to put", results in an LVC with a theme role but, when combined with the light verb *ga* "to carry", the resulting LVC has both an agent and a patient role. In (18), the use of two different coverbs with the same light verb leads to LVCs with different roles. The LVC with the coverb *niyarra* "taste" has an experiencer role while the LVC with the coverb *roowil* "to walk" has an agent role. Lastly, (19) shows how reduplicating the coverb *anggoorr* "at tears" changes the role in the LVC from experiencer to agent.

(16) Inflecting Morphology

a. yoor -ma-
come.down -put-
'to come down' (agent) [bcj]

b. yoor -mi-nginji
come.down -put-APPL
'to come down with him' (agent, accompanier) [bcj]

(17) Different Light Verbs

a. abarrabarr -ma-
confuse -put-
'to be careless' (theme) [bcj]

b. abarrabarr -ga-
confuse -carry-
'to lead someone astray' (agent, patient) [bcj]

(18) Different Coverbs

a. ni-yarra -(i)nya-
3MIN-taste -catch-

'to taste something' (experiencer) [bcj]

b. roowil -(i)nya-
walk -catch-

'to walk' (agent) [bcj]

(19) Reduplicating Coverbs

a. anggoorr -ma-
at.tears -put-

'to mourn for someone' (experiencer) [bcj]

b. anggoorr∼anggoorr -ma-
at.tears∼at.tears -put-

'to comfort someone' (agent) [bcj]

When it comes to the meaning of an LVC, it is not necessarily a sum of its parts and is frequently idiomatic. This is seen with Bardi as in the idiomaticity example in subsection 3.3.1. It is also possible for the light verb to be semantically empty, therefore not contributing any meaning to the LVC so that all of the meaning comes from the coverb. For example, in Japanese the light verb *suru* "to do" is bleached and, when combined with a noun coverb like *idoo* "movement", the resulting LVC has the verbal meaning "to move" (Matsumoto, 1996, p. 108). Sometimes the light verb will lend part of its meaning to the construction. In Urdu, for example, the coverb *yaad* "memory" can be combined with the light verb *kar* "to do" or with the light verb *aa* "to come". In both cases the LVC means "to remember", though in the latter case, *yaad aa* means "to remember" in the sense that "the memory comes to" someone (Butt, 2010, p. 52). This can also be found in the more semantically compositional LVCs in English. For example, the LVC *take a bath* refers to "bathing (oneself)" and the LVC *give a bath* refers to "bathing (someone else)". It is clear that both the light verb and the coverb can contribute to the meaning of the LVC, but that the light verb's contribution ranges from none (i.e. bleached) to some.

3.3.3 Summary

In this section, I explored the semantics of LVCs, specifically why MWEs like LVCs can be challenging for NLP technology and how compositionality works with respect to LVCs. In the next section, I explore some analyses of LVCs using different grammar frameworks, which provide representations of syntax and semantics.

3.4 LVC Analyses

This section looks at two grammar frameworks — Head-driven Phrase Structure Grammar (HPSG) and Lexical Functional Grammar (LFG) — and explores examples of analyses for LVCs that use each framework. While both rely on aspects of phrase-structure grammar (i.e. constituency grammar), they do so in different ways.

3.4.1 HPSG

As per the HPSG handbook, Godard & Samvelian (2021) claim that the light verb and coverb share one argument structure (p. 471) and their analysis relies on a property of HPSG that allows for heads to share "information with their expected complements" (p. 472). For the claim, they compare two example sentences in Persian: a non-LVC in (20a) and an LVC in (20b).

(20) a. Maryam be bāqče āb dād.

Maryam	be	bāqče	āb	dād
Maryam	to	garden	water	give.PST

'Maryam watered the garden.' [per] (Godard & Samvelian, 2021, p. 471)

b. Maryam bāqčerā āb dād.

Maryam	bāqče=rā	āb	dād
Maryam	garden=DO	water	give.PST

'Maryam watered the garden.' [per] (Godard & Samvelian, 2021, p. 472)

In (20a), the heavy verb *dādan* "to give" takes two complements: the PP[13] *be bāqče* "to [the] garden" and the noun *āb* "water". However, in (20b), the light verb *dādan* "to give" and the coverb *āb* "water" together take a single complement: the noun *bāqče* "garden", which is marked with the clitic *=rā*, indicating that the marked noun is the direct object.

To account for the semantics of LVCs, for cases where the light verb is semantically bleached, they propose making the relation of the light verb that of the coverb. For example, in the Persian example in (21), they propose a relation of *kick-relation* for the *zadan* "to hit" light verb. This is similar to my analysis of bleached light verbs, where the coverb contributes event for the sentence and its relation describes the action that is occurring.

(21) Olāq be Omid lagad zad.

Olāq	be	Omid	lagad	zad
donkey	to	Omid	kick	hit.PST

'The donkey kicked Omid.' [per] (Godard & Samvelian, 2021, p. 475)

For some cases where the light verb does contribute to the meaning of the construction, they propose using the BACKGROUND feature to give further information on the situation. The Grammar Matrix does not use the BACKGROUND feature so cases like this are not reflected in my analysis. In other cases, they propose having both the coverb and light verb contribute predications that together describe the action (to some extent). In the Persian LVC *gul zudan* (lit. deceit hit) "to deceive", Godard & Samvelian (2021, p. 477) make the coverb's predication *internal-problematic-state* and the light verb's predication *pretend-relation*. The relation of the light verb is expressing the concept of an action negatively affecting someone while the coverb's predication comes from a grouping of similar coverbs that all express states of internal problems (e.g. a state of fatigue). Although my analysis for unbleached light verbs does rely on each element having its own predication, those predications are equivalent to the glosses provided by the user for each the light verb and

[13] PP = prepositional phrase

coverb. In the last case, which is for an idiomatic LVC, they introduce a new feature (LID) but, similar to the BACKGROUND feature, the Grammar Matrix does not use it and therefore is not reflected in my analysis of LVCs.

3.4.2 LFG

In Butt (2010, pp. 55–57), she gives her own LFG analysis for light verbs, differentiating between constructions with light verbs and those with heavy verbs. Butt's (2010) analysis relies on the concept of monoclausality, where the light verb and coverb combine to form a "syntactically monoclausal predication" (p. 55). The distinction between constructions with light verbs versus heavy verbs is shown in the Urdu examples in (22). (22a) shows an example of a monoclausal permissive, which uses the light verb *dii-* "to give" and forms a monoclausal construction with the coverb *kat-* "to cut". (22b) shows an example of a biclausal tell-construction, which uses the heavy verb *kaah-* "to say" and forms a biclausal construction with the verb *kat-* "to cut". The concept of monoclauslity, where two elements form one predication with shared argument structure, is not possible in the Grammar Matrix. However, due to this analysis presented by Butt (2010), my analysis of LVCs sought to keep the light verb and coverb as closely connected (syntactically) as possible.

(22) a. Naadyaane Yassinko paodaa katne diiyaa.

naadyaa=ne	yassin=ko	paodaa	kat-ne	dii-yaa
Nadya.F.SG=ERG	Yassin=INS	plant.M.NOM	cut-INF	give-PRF.M.SG

'Nadya let Yassin cut the plant.' [urd] (Butt, 2010, p. 51)

b. Naadyaane Yassiinko paodaa katneko kaahaa.

naadyaa=ne	yassiin=ko	paodaa	kat-ne=ko	kaah-aa
Nadya.F.SG=ERG	Yassin=INS	plant.M.NOM	cut-INF=ACC	say-PRF.M.SG

'Nadya told Yassin to cut the plant.' [urd] (Butt, 2010, p. 55)

Alba-Salas (2002, pp. 85–86) describes analyses for LVCs in LFG that have been conducted by other linguists for Japanese and Korean. In Japanese, the standard account within

LFG is that the light verb *suru* "to do" "licenses an agentive subject and an 'open' complement headed by the [coverb]" (Alba-Salas, 2002, p. 85), but not all linguists agree with this. According to Alba-Salas (2002), Isoda (1991) argues that Japanese LVCs are monoclausal constructions (similar to Butt's (2010) analysis above) which consist of an inner and a matrix predicate (the coverb and light verb, respectively) while Matsumoto (1992, 1996) say that *suru* is just a control verb. As for the Korean light verb *ha* "to do", Alba-Salas (2002) cites Kim (1991, 1993), who argue that LVCs with *ha* use feature inheritance. Aside from the monoclausal constructions (see how my analysis relates to Butt's (2010) analysis in the last paragraph), most of what Alba-Salas (2002) presents is not used in my analysis. However, my analysis of light verbs does align with part of the standard account that Abdullah, Ahmed, Anjum, et al. (2021) states for Japanese, in particular that the light verb licenses a complement headed by the coverb.

3.4.3 Summary

In this section, I explored how LVCs can be represented in the HPSG and LFG grammar frameworks. In the next section, I will explore general linguistic tests (i.e. ones that don't rely on a specific grammar framework) that can be used to identify LVCs in different languages.

3.5 How to Test for LVCs

Tests determining whether a given construction is an LVC are difficult to perform as they are often extremely language dependent. For example, in Bardi, due to the fact that word order is very free, tests relying on the order of constituents are hard to apply while tests involving functions of predicates are more useful (Bowern, 2012, p. 505). In this section, I go over the tests outlined in PARSEME's Annotation Guidelines (Savary et al., 2023), which are presented as language-agnostic. Then, I go over some tests for monoclausality, *double analyse*, and syntactic variability, which vary cross-linguistically. Lastly, I provide some specific tests for Bardi from Bowern (2012).

3.5.1 PARSEME's Annotation Guidelines

Parsing and Multi-word Expressions (PARSEME) is a scientific network that is dedicated to improving the parsing of multi-word expressions (MWEs) in natural language processing and the documentation of them in corpora and lexicons (Savary et al., 2023). They defined annotation guidelines for their corpora, which can be used to determine if a given set of words is an MWE and what kind of MWE it is. These guidelines propose a set of cross-linguistic tests to identify LVCs, specifically for cases where the coverb is a noun.[14] They differentiate between two types of LVCs.[15] First, the **LVC.full** type requires that the noun coverb is abstract (**test 0**). For example, nouns like *priority* and *anger* in English are abstract while nouns like *chair* and *keyboard* are not. If the noun is abstract (or it is unclear), the person conducting the tests can move on to the next one. It is also required that the noun coverb is predicative (**test 1**). They explain this as the coverb having at least one semantic argument.[16] For example, in the English phrase *have strength*, *strength* is a property with one semantic argument, which is the entity having strength. In the English sentence, *Joe experienced a tornado*, *tornado* is an event that does not have any semantic arguments. This would not pass the test and is therefore not an LVC. Next, the light verb's subject must be the noun coverb's semantic argument (**test 2**). This is described as the light verb linking the coverb to "one of its semantic arguments that occurs as the subject of the [light] verb" (Savary et al., 2023). For example, in the English sentence *John made a presentation to his boss*, *John* is both the subject of the light verb and is also a semantic argument of the noun (in this case, *John* is "the presenter"). In the English sentence *John's boss interrupted his presentation*, the subject (*John's boss*) of the light verb (*interrupted*) is not a semantic argument of the coverb (*presentation*). This is because "a presentation does not necessarily

---

[14] I used version 1.3 of the annotation guidelines, which can be found at `https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.3/`.

[15] These two types are LVC.full and LVC.cause, however, I do not make this distinction in my own analysis so the tests that differentiate the two will not be explained here.

[16] This is defined as "a semantically mandatory and specific participant of the event or state expressed by the predicative noun" (Savary et al., 2023).

have an interrupter" (Savary et al., 2023). In the event that a phrase or sentence fails test 2, the person conducting the tests should move on to test 5, which distinguishes between a causative LVC and a non-LVC.[17] In the next test, it is required that the light verb is semantically light (**test 3**). They define a verb as "light" if "the semantics that [the verb] adds to [the noun] is restricted to [either] what stems from its morphological features (e.g. future, plural, perfective aspect, etc.) [or] pointing at the semantic role of [the noun] played by [the verb's] subject". For example, in the English phrase *take a walk*, the verb *take* does not add any meaning to the noun *walk* other than that of "performing an activity." However, in the English phrase *start a walk*, the verb *start* adds an aspectual meaning to *walk*. This means that *start a walk* is not an LVC. Lastly, it must be possible for the light verb to be reduced (**test 4**). This means that, in a noun phrase without the verb where the verb's subject becomes the noun's dependent, this verbless noun phrase must refer to the same event or state that the original construction refers to. For example, the English sentence *Paul had a walk* can be reduced to *Paul's walk*. Both of these sentences refer to the same "walking" event. However, the English sentence *Paul got news from his brother* cannot be reduced to *Paul's news from his brother* because *Paul's news* does not refer to the original "Paul got news" event.[18] Aside from the numbered tests, they also describe a hard-to-apply syntactic test,[19] where the construction is an LVC if the light verb's subject is an argument of the noun coverb.[20]

More generally speaking, Savary et al. (2023) describe how their analysis compares to literature on LVCs. Although they don't count most aspectual support verbs as LVCs, the

---

[17]I do not further explain this test as causative LVCs are not relevant to my analysis.

[18]The descriptions and examples given here can be found on the LVC subpage of the PARSEME annotation guideline's website at `https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.3/?page=050_Cross-lingual_tests/020_Light-verb_constructions__LB_LVC_RB_`.

[19]This was originally part of the list of tests but has since been removed due to the difficulty of applying it.

[20]For further explanation and examples, please refer to `https://parsemefr.lis-lab.fr/parseme-st-guidelines/1.3/?page=050_Cross-lingual_tests/020_Light-verb_constructions__LB_LVC_RB_`.

literature is generally not as selective. Many approaches do not count cases where the light verb is semantically light instead of bleached while their approach does. Moreover, they specifically do not consider a test which is often included in LVC literature, where if there is a morphologically related verb or adjective that has the same meaning as the proposed construction, that construction won't be an LVC (e.g. *to make a visit* means the same as *to visit*). This test does not always work for three reasons: (a) not all LVCs have this verb or adjective equivalent, (b) some non-LVCs do have this equivalent, and (c) some LVCs have this equivalent but it does not actually mean the same thing as the LVC or they have a different argument structure from the LVC. Although this equivalent can be useful in determining if the noun coverb is predicative (test 1) by using paraphrase, it is just not enough to be a test on its own.

3.5.2 Tests for Monoclausality

Part of Butt's (2010) analysis relies on monoclausality of the LVC (when the light verb and coverb combine to form a "syntactically monoclausal predication" (Butt, 2010, p. 55)). Determining such monoclausality is language-dependent, with each language having a different set of tests to prove this. For example, in Romance languages, monoclausal constructions can undergo clitic-climbing, passivization, and/or reflexivization while biclausal constructions cannot.[21] To determine whether a construction is monoclausal or not in Korean, Butt's (2010) tests rely on negative polarity items (NPI), negation, and the (non-)separability of the light verb and coverb. This is shown in (23). (23a) shows an example of a Korean LVC, where *ssele* "sweep" is the coverb and *chiwessta* "clean" is the light verb. In order for a Korean sentence with the NPI items *anwuto* "nobody" and *an* "not" to be grammatical, they must co-occur in the same clause. In (23b), the clause containing *anwuto* "nobody" is shown using brackets, and *an* "not" is not in that clause, which is why the sentence is ungrammatical. In (23c), both the NPI (*anwuto*) and its licensor (*an*) are distributed over

[21] As per (Butt, 2010), these are not applicable to all Romance languages (e.g. clitic-climbing is a test for Spanish, Italian, and French while passivization and reflexivization are tests for French and Italian).

the Korean LVC with coverb *meke* "eat" and light verb *chiwessta* "clean". The NPI items are in the same clause, which shows that the construction is monoclausal and therefore an LVC. Urdu monoclausality can also be determined with the NPI test, in addition to looking at object agreement, anaphora, and control.

(23) a. Chelswuka namwunipul ssele chiwessta.

Chelswu=ka namwunip-ul ssel-e chiw-ess-ta
Chelswu=NOM leaves-ACC sweep-E clean-PST-DECL

'Chelsu has swept up the leaves.' [kor] (Butt, 2010, p. 58)[22]

b. *Chelswunun anwuto pamul ilkesstako an malhaessta.

Chelswu-nun [anwuto pam-ul ilk-ess-ta]-ko an malha-ess-ta
Chelswu-TOP nobody chestnut-ACC eat-PST-DECL-COMP NEG say-PST-DECL

'Chelswu did not say that nobody ate the chestnut.' [kor] (Butt, 2010, p. 58)

c. Anwuto pamul an meke chiwessta.

anwuto pam-ul an mek-e chiw-ess-ta
nobody chestnut-ACC NEG eat-E clean-PST-DECL

'Nobody (children) has eaten up the chestnut.' [kor] (Butt, 2010, p. 58)

3.5.3 Language-Specific Tests

In the next three subsections, I describe language-specific tests for identifying LVCs. These include tests in Romance languages, English, and Bardi.

Tests for LVCs in Romance Languages

An important aspect of Alba-Salas's (2002) dissertation is the *double analyse*[23] phenomenon, where "the prepositional complement of the [coverb] can be analyzed as either being inside its maximal projection[24] or as a direct dependent of the light verb" (pp. 12-13).

[22] The *-e* suffix is glossed as E by Butt (2010) and is not explained further.

[23] This name comes from Lexicon-Grammar literature.

[24] This refers to the constituent that is at the highest level in the X-bar structure.

In other words, the prepositional complement of a coverb can be analyzed as an argument of either the light verb or the coverb. This phenomena is specifically introduced for LVCs in Romance languages and further research would have to be done in order to decide if it exists in other languages as well. Most Romance LVCs exhibit this phenomenon, so it can be useful in identifying them.[25] Citing Gross (1976), Alba-Salas (2002) lists passivization, clefting, relativization, and pronominalization as tests for *double analyse*. This can be shown using the French LVC in (24)[26] (with the light verb in **bold** and the coverb in *italics*). In this example, *contre Luc* "against Luc" is a prepositional complement of the coverb.

(24) Max a **commis** une *agression* contre Luc.
'Max **committed** an *assault* against Luc.' [fra] (Alba-Salas, 2002, p. 31)

In (25)–(28), examples of the four tests (passivization, clefting, relativization, and pronominalization, respectively) are shown.[27] The first example for each process indicates that the coverb undergoes the process independently from its prepositional complement while the second example indicates that they undergo the process together. This shows that the prepositional complement can be inside of the noun phrase that is headed by the coverb or it can be a direct syntactic dependent of the light verb, therefore exhibiting the *double analyse* phenomenon.

(25) Passivization

a. [Une agression] a été commise par Max [contre Luc].
'An assault was committed by Max against Luc.' [fra]

[25]The situation is a bit complicated, since not all Romance LVCs exhibit *double analyse* and some constructions with heavy verbs do exhibit it. However, Alba-Salas (2002) claims that constructions with heavy verbs exhibit a different kind of *double analyse* than LVCs. Additionally, he says that the Romance LVCs that do not allow *double analyse* require further exploration. The nuances of this go beyond the scope of this thesis but *double analyse*, and the ways of testing for it, are still an important tool in identifying Romance LVCs.

[26]Alba-Salas (2002) did not provide full IGT for the French examples used in this section.

[27]All examples come from Alba-Salas (2002, p. 32).

b. [L'agression contre Luc] a été commise par Max.
'The assault against Luc was committed by Max.' [fra]

(26) Clefting

a. C'est [une agression] que Max a commis [contre Luc].
'It's an assault that Max committed against Luc.' [fra]

b. C'est [une agression contre Luc] que Max a commis.
'It's an assault against Luc that Max committed.' [fra]

(27) Relativization

a. [l'agression] que Max a commise [contre Luc]
'the assault that Max committed against Luc' [fra]

b. [l'agression contre Luc] que Max a commise
'the assault against Luc that Max committed' [fra]

(28) Pronominalization

a. [Cette agression] Max l'a commise [contre Luc].
'lit. That assault, Max committed it against Luc.' [fra]

b. [Cette agression contre Luc], Max l'a commise.
'lit. That assault against Luc, Max committed it.' [fra]

In the heavy verb construction in (29a) (with heavy verb *rapporté* "recounted"), the noun *agression* alone cannot undergo each of the four tests (Alba-Salas, 2002, p. 33). Instead, the prepositional complement *contre Luc* "against Luc" must be included and the entire phrase *une agression contre Luc* must be used for each of the tests. This can be seen with the passivization test in (29b) and (29c), where (29b) is ungrammatical because *une agression* cannot be passivized on its own. (29c) is grammatical because the noun and its prepositional complement together are passivized. This shows that the prepositional complement cannot be analyzed as an argument of both the light verb and coverb, and therefore does not exhibit the *double analyse* phenomenon.

(29) a. Max a rapporté une agression contre Luc.
'lit. Max recounted an assault against Luc.' [fra]

b. *[Une agression] a été rapportée par Max [contre Luc].
'Intended: An assault was recounted by Max against Luc.' [fra]

c. [L'agression contre Luc] a été rapportée par Max.
'The assault against Luc was recounted by Max.' [fra]

Tests for English LVCs

According to Sag et al. (2002, p. 7), English LVCs can undergo passivization, extraction, and internal modification, which shows that they have syntactic variability and cannot be treated as words-with-spaces or full idioms.[28] Given the English LVC in (30a) (with coverb *demo* and light verb *give*), the LVC can be passivized (30b), the coverb can be extracted (30c), or the light verb can be modified with an adjective like *revealing* (30d).

(30) a. *give a demo* [eng]

b. Passivization
a demo was given [eng]

c. Extraction
How many demos did Kim give? [eng]

d. Internal Modification
give a revealing demo [eng]

Tests for Bardi

Bowern (2012) provides a list of tests for determining the status of LVCs, which in many cases doubles as evidence of syntactic closeness between the light verb and coverb.

[28] They did not explicitly state that these can be applied cross-linguistically and, based on the language-dependency of the tests provided in the previous sections, I am assuming that these can be applied to English LVCs and not LVCs in any language.

Even though coverbs are always separate syntactic words in an LVC, Bardi coverbs include monosyllabic ones which often cliticize to the light verb (in addition to being phonologically reduced). For example, in the LVC *arr indan* "he's coming/going" (with coverb *arr* "come/go" and light verb *indan* "do/say"), *arr* can cliticize to *indan* to form *andan.* Bowern (2012) uses this as an example of syntactic closeness. The only other words that regularly cliticize are sentential particles and possessive pronouns (Bowern, 2012, p. 506). Therefore, the presence of cliticization could be used as a test to determine if a noun, verb, or adjective are acting as a coverb, since the non-coverb form cannot cliticize.

Additionally, the placement of sentential clitics can be used to identify LVCs. Clausal clitics can only be placed after the first constituent in a clause; however, if an LVC is the first constituent of the clause, the clitic can either come after the coverb or the light verb with no difference in resulting meaning (Bowern, 2012, pp. 506–507). In the Bardi example in (31), Bowern (2012) shows how the phrasal clitic for allative case *-ngan* can attach to the coverb *roowil* "walk" (in (31a)) or the light verb *manyan* "catch" (in (31b)) and have the same meaning. This indicates that the coverb and light verb can be treated as a single constituent.

(31) a. Roowilngan manyan gornamb

roowil-ngan	ma-nya-n	gorn=amb
walk-ALL	GER-catch-CONT	good=THUS

'Walking is good/it's good to walk.' [bcj] (Bowern, 2012, p. 507)

b. Roowilngan manyan gornamb

roowil	ma-nya-n-ngan	gorn=amb
walk	GER-catch-CONT-ALLxw	good=THUS

'Walking is good/it's good to walk.' [bcj] (Bowern, 2012, p. 507)

Next, adjectives cannot be reduplicated unless they are acting as coverbs, which also means that being a coverb (as opposed to an adjective) determines what derivational rules the root can undergo (Bowern, 2012, p. 508). The Bardi sentence in (32a) is ungrammatical because the adjective being reduplicated is not acting as a coverb while the sentence in (32b) is grammatical because the adjective being reduplicated is acting as a coverb. Bowern (2012) uses this to show that the status of a word being a coverb changes derivational possibilities.

(32) a. *garrja∼garrja jamooyoon

garrjagarrja jamooyoon
sharp∼REDUP knife

'Intended: a sharp knife' [bcj] (Bowern, 2012, p. 508)

b. Garrjagarrja anama

garrja∼garrja a-na-ma
sharpen∼REDUP 2.IMP-TR-put-FUT

'Sharpen it!' [bcj] (Bowern, 2012, p. 508)

Finally, in a non-LVC construction, the negator *arra* must directly precede the inflecting verb, but in LVCs, the negator must go before the coverb (Bowern, 2012, p. 509). This means that if the negator is not directly before the inflecting verb, it is before the coverb and the construction is an LVC. A simple negative sentence in Bardi (with *arra* preceding the inflecting verb) is shown in (33a). (33b) shows a negative LVC in Bardi (with *arra* preceding the coverb *jiidara* "bewitch" instead of the inflecting light verb *ngalamanajiy* "put"). The Bardi sentence in (33c) is ungrammatical because the negator is intervening between the coverb and the light verb. This can be useful in determining when an item is acting as a coverb and when it is acting as its original part of speech. If the negator is in between an item that is possibly a coverb and an inflecting verb, it can be confirmed that the item is not acting as a coverb and the construction is therefore not an LVC.

(33) a. Arra oolalana.

arra oo-l-ala-na
NEG 3-IRR-see-REM.PST

'He didn't see it.' [bcj] (Bowern, 2012, p. 509)

b. Arra jiidara ngalamanajiy irrola.

arra	jiidara	nga-la-ma-na=jiy	irrola
NEG	bewitch	1-IRR-put-REM.PST=3MIN.IO	spear

'I didn't bewitch your spear.' [bcj] (Bowern, 2012, p. 509)

c. *Jiidara arra ngalamanajiy irrola.

jiidara	arra	nga-la-ma-na=jiy	irrola
bewitch	NEG	1-IRR-put-REM.PST=3MIN.IO	spear

'Intended: I didn't bewitch your spear.' [bcj] (Bowern, 2012, p. 509)

3.5.4 Summary

This section looked at different cross-linguistic and language-dependent tests for identifying LVCs. As shown, these tests tend to be language-dependent, as they rely on specific features that are not universal to all languages.

3.6 Summary

In this chapter, I provided an overview of the literature on light verb constructions. This included sorting through the terminology surrounding these constructions (e.g. the possible ways of referring to LVCs and similar but different constructions) as well as different morphological, syntactic, and semantic properties. I also explored some grammar frameworks that have been used to represent LVCs and gave possible ways of testing for LVCs. Additionally, throughout this chapter, I described the ways in which the literature informed my analysis of LVCs and how I implemented them for the library.

Of the range of variation described in this chapter, my library will account for the following subset. The LVC library accounts for LVCs where the verb and coverb are free words. Currently, the library accounts for noun and verb coverbs, although other parts of speech are possible (e.g. adjectives or adverbs). For the dependency between light verb and coverb, I handle this by allowing coverbs to indicate which light verbs they can combine with. I model the dependencies between the coverb and its complements but not any dependencies between the coverb and the arguments of the LVC (e.g. subject or direct object). My library accounts for both regular and bleached light verbs but, for bleached light verbs, I only handle them combining with verb coverbs (not noun coverbs). The light verbs that I model introduce their own event, unless they are fully bleached. Additionally, the coverb is semantically contentful and introduces an event (in verb coverbs) or an instance (in noun coverbs). In the next chapter, I will describe my analysis and implementation in the LVC library.

Chapter 4
ANALYSIS AND IMPLEMENTATION

In this chapter, I discuss my analysis and implementation, but first, I will give an overview of my analysis. In an LVC, I analyze the light verb as the head daughter and the coverb (or the constituent it heads) as the non-head daughter. They combine using one of two new phrase structure rules that I defined, which are specialized variants of head-complement rules. The details on how these rules work will be discussed in section 4.2.2. In general, a goal of my analysis was to have the light verb combine with the coverb (or the constituent it heads) first, before the light verb combines with any other arguments (with one exception which will be discussed with the Persian example below). The reason behind this is that, with multiword expressions (e.g. LVCs), there is a dependency between the different parts of the expression, which my analysis accounts for.

In the next sections, I will go into more detail on the mechanics of my analysis and how it is implemented into the Grammar Matrix customization system. First, in order to show the range of variation possible in the LVC library, I provide some example sentences from my illustrative languages. I also include the desired syntactic and semantic representation for each of these sentences. Next, I will explain how the LVC feature works (in section 4.2.1). This feature is used to control which light verbs combine with which coverbs, as well as to make sure that the LVC phrase structure rules are only used to combine light verbs with their corresponding coverbs. I will then go into more detail about the phrase structure rules (in section 4.2.2) and lexical types (in sections 4.2.3 to 4.2.5) that allow sentences like the examples listed here to parse. Then, I will explain how the LVC-related types fit into the lexical type hierarchies, specifically in relation to my implementation within the Grammar Matrix customization system (in figure 4.9). Lastly, I will explain how the LVC library is implemented in the front-end Grammar Matrix questionnaire (in section 4.3).

4.1 Example Parses and Semantic Representations

The sentence from Bardi shown in (34) contains the transitive light verb *-nya-* "to catch", which combines with the noun coverb *liyan* "heart".

(34) Aamba nimalnga liyan innyana.

aamba-nim	nimal-nga	liyan	i-n-nya-na
man-ERG	nose-INS	heart	3-TR-catch-PST

'The man breathed through his nose.' [bcj] (Bowern, 2012, p. 440)

The desired tree and MRS for this sentence is shown in figure 4.1.[1] In the tree, each word in the sentence goes through a number of lexical rules (both inflecting and not), as indicated by the strings of Ns and Vs. For example, the subject *aamba* "man" goes through a lexical rule that assigns ergative case to it by adding the *-nim* suffix. Since the coverb comes before the light verb in Bardi, the *comp-head-lvc* rule is used to form the LVC *liyan innyana* "breathed" (this is how the NP for *liyan* and the V for *innyana* combine into a V in the tree). In the MRS, the relation for the light verb (*_catch_v_lv_rel*) introduces the event *e2*. The coverb (*_heart_n_rel*) is identified with the ARG2 of the light verb (indicated by the *x15* instance). The object is *nimalnga* "through the nose", which has a the *_nose_n_rel* relation in the MRS and is identified with the ARG3 of the light verb (indicated by the *x10* instance). The subject (*aambanim* "man") is identified with the ARG1 of the light verb, which is the same as it would be for a heavy verb. In my analysis for unbleached light verbs, the light verb introduces an event and its ARG2 is always identified with the coverb. The object, which in a construction with a heavy verb would be identified with the ARG2 of the verb, is instead identified with the ARG3 of the light verb in LVCs.

[1] This sentence results in two parses, one of which is incorrect. The reason behind this will be discussed in section 5.1.1.

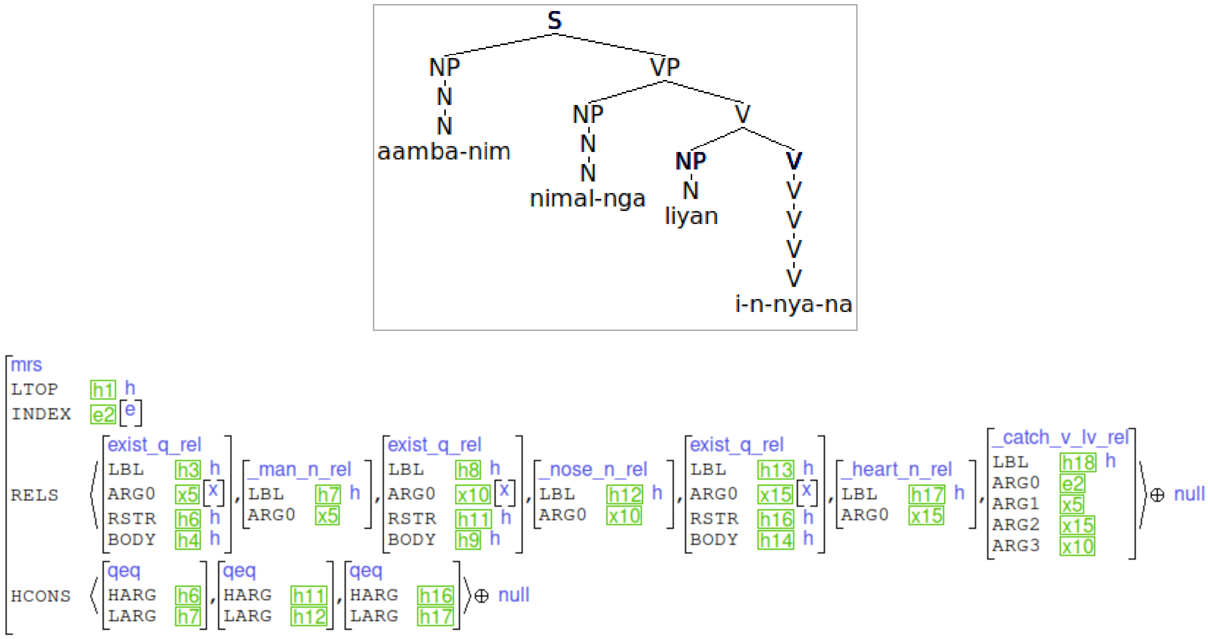


Figure 4.1: Parsing *Aamba nimalnga liyan innyana*

The sentence from English shown in (35) contains the intransitive light verb *take*, which combines with the noun coverb *shower*.

(35) Jordan takes a shower.

Jordan	take-s	a	shower
Jordan	take.PRS-3SG	DET.INDF	shower

'Jordan takes a shower.' [eng]

The desired tree and MRS for this sentence is shown in figure 4.2.[2] The main point of interest of this sentence is that the coverb takes a dependent, which is the determiner *a*. In the tree, the determiner and the coverb combine into an NP. That NP combines with the V for *takes* using the *head-comp-lvc* rule (since the coverb comes after the light verb in English)

[2]This sentence could also be interpreted as a construction with a heavy verb. Since the point is to show my analysis of light verbs, I do not discuss that interpretation here. See section 5.1.2 for that discussion.

in order to form the VP. Since the determiner is part of the coverb's constituent, there aren't any issues with having a word between the light verb and the coverb. However, in some languages, coverbs are not allowed to take dependents. For these languages, I modified the lexical types for coverbs to block them from taking dependents (this will be explained in section 4.2.3). As shown in the MRS, the coverb (along with its determiner) is identified with the ARG2 of the light verb and the subject is identified with the ARG1 of the light verb, as with the Bardi example.

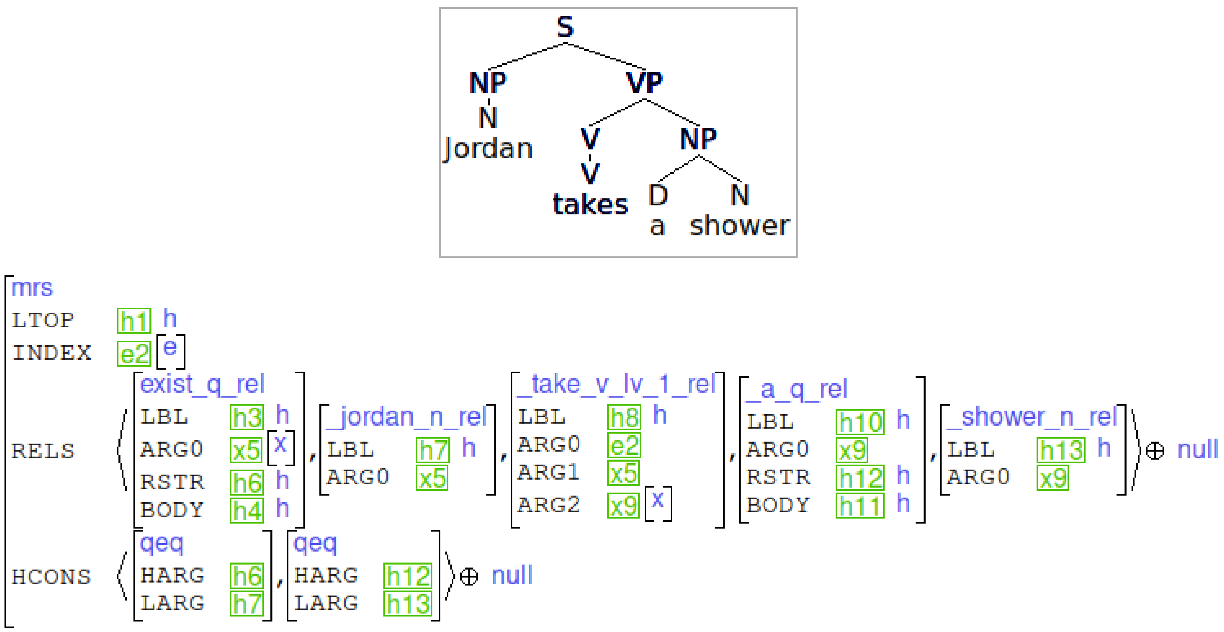


Figure 4.2: Parsing *Jordan takes a shower*

The sentence from Japanese shown in (36) contains the bleached intransitive light verb *shi* "to do", which combines with the verb coverb *benkyou* "study".[3]

[3]I analyze *benkyou* "study" as an intransitive coverb (taking only an ARG1 semantically). However, this might not be accurate for Japanese.

(36) Hanako ga benkyou shita.

Hanako	ga	benkyou	shi-ta
Hanako	NOM	study	do-PST

'Hanako studied.' [jpn][4]

The desired tree and MRS for this sentence is shown in figure 4.3. In the tree, the *comp-head-lvc* rule is used to combine the light verb and coverb (since the coverb comes before the light verb). Looking at the tree, it doesn't look like there is much difference between it and the unbleached examples shown for Bardi and English. This is because the differences are semantic, which is clear when looking at the MRS. There is no relation for the bleached light verb, since it doesn't provide anything to the sentence semantically. This means that the arguments that, for an unbleached light verb, are identified with the light verb are instead identified with the coverb. The subject *Hanako* (*_hanako_n_rel*) is identified with the ARG1 of the coverb (*_study_v_rel*).[5] Additionally, the coverb introduces an event (*e2*) and the light verb does not (as it does not contribute a predication and is not on the RELS list). In order to account for the semantic differences between bleached and unbleached light verbs, I use different lexical types for bleached light verbs and the coverbs that combine with them.

The sentence from Persian shown in (37) contains the light verb *zad* "to hit", which combines with the noun coverb *dast* "hand".

(37) Omid dast be golhā zanad.

Omid	dast	be	gol-hā	zan-ad
Omid	hand	to	flower-PL	hit-3SG

'Omid touches the flowers.' [per] (Godard & Samvelian, 2021, p. 470)

[4] This example comes from the AGGREGATION data testsuite for Japanese, which is not publicly accessible.

[5] In cases with unbleached light verbs and verb coverbs, the subject is identified with the ARG1 of the light verb and the coverb.

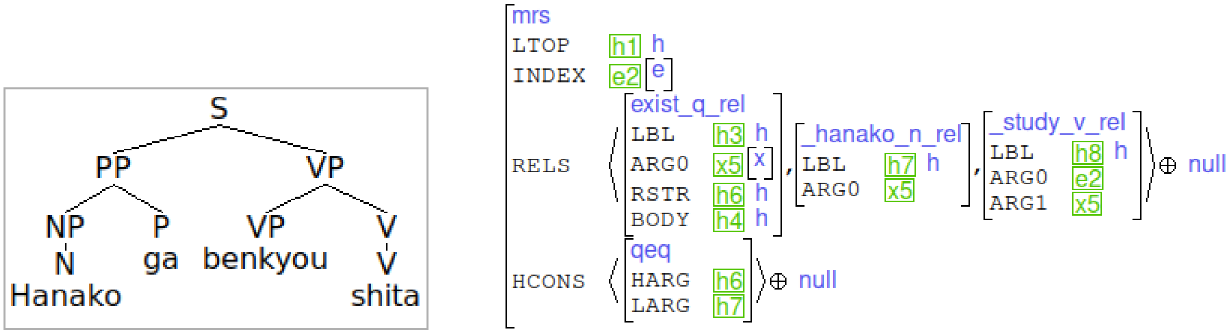


Figure 4.3: Parsing *Hanako ga benkyou shita*

The correct tree and MRS for this sentence is shown in figure 4.4. The main point of interest of this sentence is that an element (that is not a dependent of the coverb) is allowed to intervene between the light verb and coverb. In this example, the object *be golhā* "the flowers" is between the light verb and coverb. As I mentioned earlier, a part of my analysis included having verb and coverb (or the consitutent it heads) combine first, which is done to avoid extraneous ambiguity. However, that is not possible in this example as the intervening object is a complement of the light verb, not the coverb. Instead, V for the light verb and the PP for the object combine first to form the V *be golhā zanad*.[6] Then, this V combines with the NP for the coverb using the *comp-head-lvc* rule (since the coverb comes before the light verb), to form the LVC (with the object). As opposed to the bleached light verb case shown above in Japanese, this difference is syntactic and not semantic. The MRS for this sentence is not any different from one for a sentence where the elements appear in a different order (e.g. one where the intervening element is not between the coverb and the light verb). The coverb is still identified with the ARG2 of the light verb and the object is still identified with the ARG3 of the light verb. Some languages do not allow arguments of the light verb to intervene between the light verb and coverb. In order to block these kinds of constructions from parsing in such languages, I modified the *head-comp-lvc* and *comp-head-lvc* rules (this

[6]This is done using the *comp-head-2* rule.

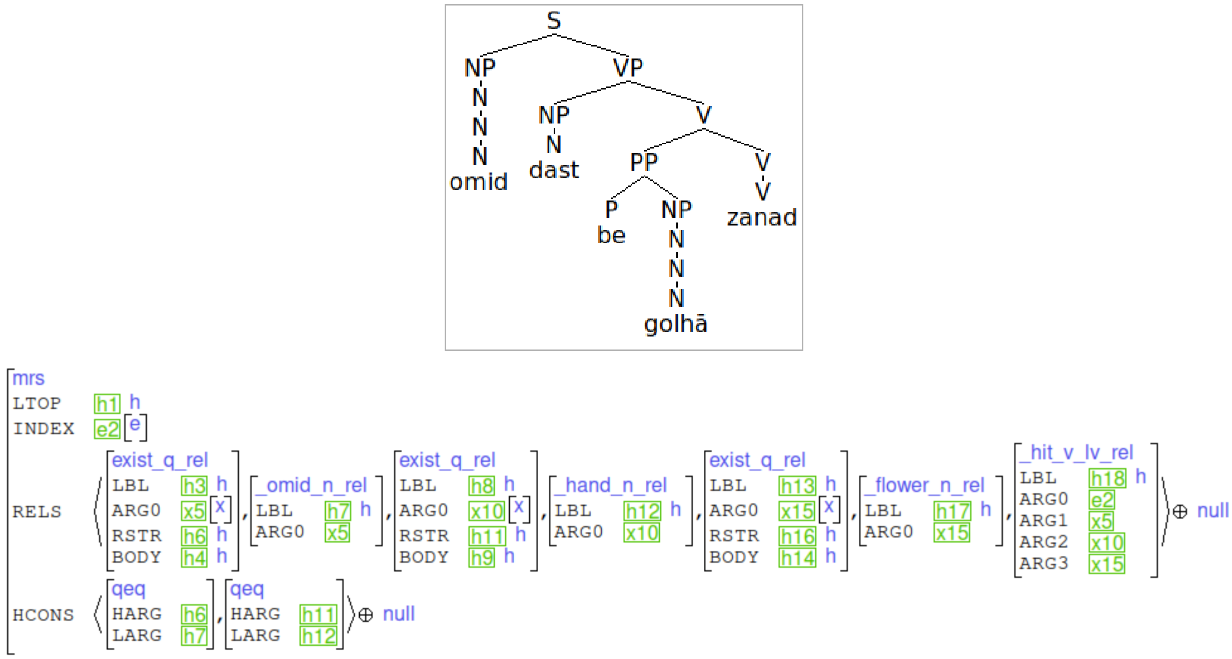


Figure 4.4: Parsing *Omid dast be golhā zanad*

will be explained in section 4.2.2). However, for languages like Persian that allow intervening elements, no changes are made to the LVC-specific head-complement rules.

4.2 Analysis and Implementation in the Grammar Matrix Code

In this section, I provide my analysis for how light verb constructions are incorporated into grammars in the LVC library. I also explain my implementation of this analysis, citing specific functions and files. These files are part of the Grammar Matrix codebase, which can be found on GitHub (at `https://github.com/delph-in/matrix/tree/trunk`).[7] The files mentioned in this section are either in `gmcs/` or `gmcs/linglib/`. I begin my explaining what the LVC feature is and how it is used in the LVC library. Next, I explain the phrasal types

[7]The version of the Grammar Matrix customization system that corresponds to the analysis and implementation described here can be found at `https://github.com/delph-in/matrix/tree/77033002bfada2b94e7499ae905955f815754066`.

that I added, why the existing ones were not sufficient, as well as changes that I made to the existing ones. Then, I explain my analysis and implementation of lexical types and entries for light verbs and coverbs as well as the changes I made to the customization of the lexical type hierarchies.

4.2.1 The LVC *Feature*

Analysis

One of the first things I did was find a way to connect light verbs and coverbs and allow for only certain combinations of them (as specified by the user). For this, I defined a new feature — the LVC feature — which is specified on the HEAD of the coverb and allows a coverb to indicate which light verb(s) it can combine with. For example, in the Bardi example from earlier with the LVC *liyan innyana* "breathed", the lexical type that the coverb *liyan* "heart" inherits from specifies a value of *lv-nya-tr-noun-trans* for its LVC feature. This value corresponds to the transitive light verb *-nya-* "to catch" that can combine with noun coverbs. The lexical type that the light verb inherits from specifies which value for LVC is allowed for its coverb complement.[8] The values for the LVC feature are drawn from the subhierarchy shown in figure 4.5.

In this hierarchy is the *lvc* type, which inherits from the initial **top** type. All other LVC feature types inherit from *lvc*. There are two types that inherit directly from *lvc* — *lv-none* and *lv-all* — which, in addition to the *lvc* type, are included in all grammars with LVCs. The *lv-none* type is used either for lexical types that are not coverbs or to prevent coverbs from being used in certain phrasal types. The *lv-all* type is used for lexical types that are coverbs and can be used to specify that a coverb accepts any light verb. If a coverb has *lv-all* as the value for its LVC feature, it can combine with any light verb. More specific types can also be created that inherit from *lv-all*, which allow a coverb to combine with only some light verbs. There are two types that inherit from *lv-all*, which are added depending on whether

[8] This will be described in more detail in section 4.2.3.

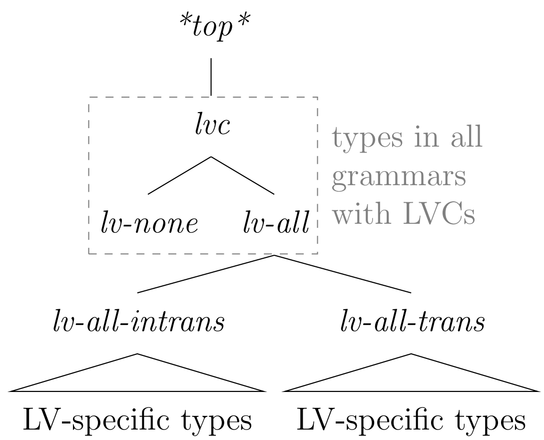


Figure 4.5: LVC Type Hierarchy

a language has intransitive and/or transitive light verbs. If the language allows intransitive light verbs, the *lv-all-intrans* type will be added and, if the language allows transitive light verbs, the *lv-all-trans* type will be added. After that, depending on the coverbs defined in the Lexicon subpage and which light verbs they select for, additional types will be added. The *lv-nya-tr-noun-trans* from earlier is an example of one of these types, which would inherit from the *lv-all-trans* type.

In addition to being used to constrain which light verbs can combine with which coverbs, the LVC feature is also used to indicate whether an element is a coverb or not. This is used primarily by different phrase structure rules, where some rules require specific elements to be coverbs and other rules require specific elements to not be coverbs.[9] As mentioned above, elements with the *lv-none* type are not coverbs and those with the *lv-all* type (or one of the subtypes of *lv-all*) are coverbs.

Implementation

The LVC feature is added to the language-specific TDL file in `light_verb_constructions.py`, using the `customize_light_verb()` function. This function specifies which head types

[9]This will be explained in section 4.2.2.

the LVC feature is appropriate for. Currently, since only noun and verb coverbs are supported by the LVC library, the disjunctive type will be `+nv` (except in the case where a language uses case-marking adpositions, in which case the type will be `+nvp`). The hierarchy is initialized in the same file (using the `init_light_verb_hierarchy()` function) with the default types as well as a type for each light verb. The format for the light verb types follows the schema in (38).

(38) lv-[{lv-name}]-[intrans/trans][10]

This hierarchy could be modified later on when creating the coverb lexical entries in `lexical_items.py`. Depending on the combination of light verbs that a coverb might select for, a new type could be created. The `get_type_covering()` function that is part of the `TDLHierarchy` class is then used to fit that type into the hierarchy.

4.2.2 LVC Phrasal Types

Analysis

In my analysis, the relationship between a light verb and a coverb is that of a head and its complement (i.e. the coverb is the complement of the light verb).[11] In Matrix-produced grammars, there exist phrasal types (also known as phrase structure rules) that combine heads and with their complements. Two of these are *head-comp* and *comp-head*, where the order of "head" and "comp" correlate to the order of those two constituents. Not all languages will have both of these rules, as this depends on the order of heads and complements for each language. For example, languages with SOV word order would only have the *comp-head* rule since the object (complement) is to the left of the verb (head). There are also the *head-comp-2* and *comp-head-2* rules, which are used in languages with verb-second or free

[10] {lv-name} refers to the name the user assigns to a particular light verb. Square brackets [ ] indicate required elements with the possible values for the element separated by a slash /. In the other schemas provided in this chapter, parentheses ( ) will indicate optional elements.

[11] This is not true of all published analyses, especially for ones that treat LVCs as syntactically monoclausal and sharing the same head.

word order, as complements can occur in positions that the other head-complement rules cannot model.

Initially, I tried to use the existing head-complement rules to combine light verbs and coverbs. However, the order of the light verb and coverb does not always match the order of heads and their complements in all languages. For example, although Bardi has free word order, this does not apply to the light verb and coverb. The coverb must always precede the light verb. Using the existing head-complement rules, ungrammatical sentences where the coverb comes after the light verb parses. However, constraining the existing rules would prevent some grammatical sentences from parsing. My solution was to create two new phrase structure rules that correlate specifically to the order of the light verb and coverb, regardless of a language's word order. These two rules are *head-comp-lvc*, which is used in languages where the coverb appears after the light verb, and *comp-head-lvc*, which is used in languages where the coverb appears before the light verb. If a language allows coverbs to appear before and after light verbs, both rules will be added. Both of these rules specify that the first element on the COMPS list of the head daughter is [LVC *lv-all*]. This constrains this element to be a coverb (since only coverbs will have an LVC value that is or inherits from *lv-all*). The *head-comp-lvc* rule inherits from the *head-comp-phrase-lvc* type, which itself inherits from *basic-head-1st-comp-phrase* and either *head-initial* or *head-initial-head-nexus* (the latter is only used if the language has free or verb-second word order). Its AVM can be found in (39).

(39) $\begin{bmatrix} \textit{head-comp-phrase-lvc} \\ \text{HEAD-DTR}\,|\,\text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CAT}\,|\,\text{VAL}\,|\,\text{COMPS} \quad \left\langle \begin{bmatrix}\text{LOCAL}\,|\,\text{CAT}\,|\,\text{HEAD}\,|\,\text{LVC} \quad \textit{lv-all}\end{bmatrix} \right\rangle \end{bmatrix}$

In the English example from earlier with the LVC *takes a shower*, the *head-comp-lvc* rule is used to form the LVC (see figure 4.6). In the grammar used to parse this sentence, the coverb *shower* is [LVC *lv-all*]. When it combines with the determiner *a*, it is the head daughter so the value for this feature will be carried up.[12] The light verb is then able to combine with the coverb using the *head-comp-lvc* rule, since the first element on the comps

[12]This is done using the *head-spec* rule.

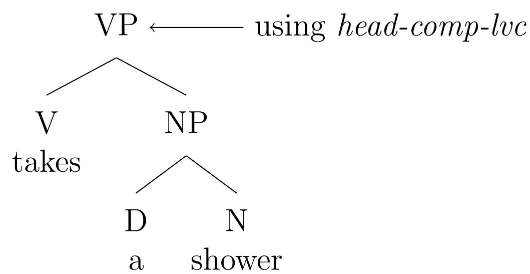


Figure 4.6: Tree for English LVC *takes a shower*

list of the head daughter, which would be the coverb, is [LVC *lv-all*] (this would also work if the coverb has a value for LVC that is a subtype of *lv-all*).

The *comp-head-lvc* rule inherits from the *comp-head-phrase-lvc* type, which itself inherits from *basic-head-1st-comp-phrase* and either *head-final* or *head-final-lvc* (the latter is only used in languages with free word order). The *head-final-lvc* type inherits from the *head-final* type and adds the [ATTACH *lmod*] feature. This allows the coverb to combine with the light verb first (i.e. before combining with other elements). The AVMs for the *head-final-lvc* and *comp-head-phrase-lvc* types can be found in (40) and (41), respectively.

(40) $\begin{bmatrix} \textit{head-final-lvc} \\ \text{SYNSEM} \mid \text{ATTACH} \quad \textit{lmod} \end{bmatrix}$

(41) $\begin{bmatrix} \textit{comp-head-phrase-lvc} \\ \text{HEAD-DTR} \mid \text{SYNSEM} \mid \text{LOCAL} \mid \text{CAT} \mid \text{VAL} \mid \text{COMPS} \quad \left\langle \text{LOCAL} \mid \text{CAT} \mid \text{HEAD} \mid \text{LVC} \quad \textit{lv-all} \right\rangle \end{bmatrix}$

In addition to defining *head-final-lvc*, in order to prevent objects from combining with the coverb before the coverb has combined with the light verb (specifically in cases where the word order is subject, object, coverb, and then light verb), my analysis involves a change to an existing type: *decl-head-subj-phrase*. Figure 4.7a shows the correct way for the elements to combine and figure 4.7b shows the incorrect way. This involves requiring *lv-none* for the LVC feature on the head daughter. This type is inherited by the *head-subj* and *subj-head* rules. Since the coverb, and not the light verb, carries the specific LVC feature types, when the coverb and light verb combine, the head daughter takes on the light verb's LVC type

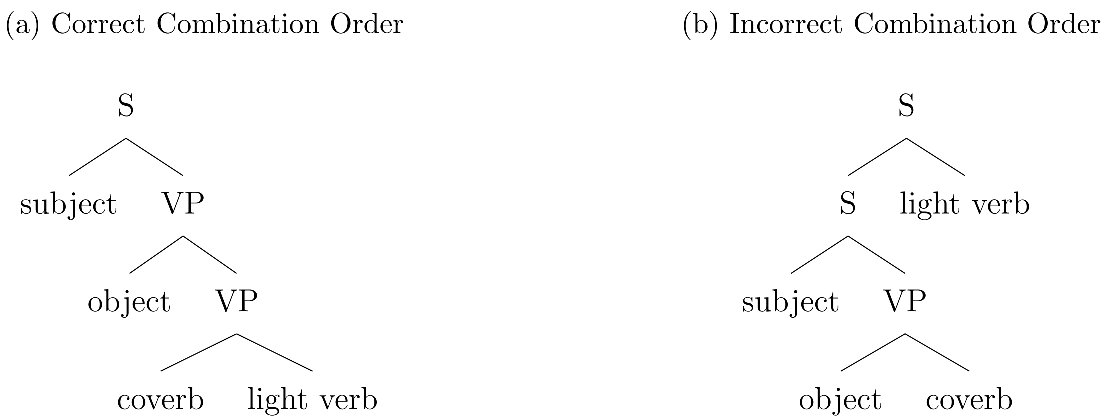


Figure 4.7: Sentence Combination Orders

(which will be underspecified). If the coverb combined with the object first, it could carry the coverb's LVC type up and then combine with the subject.

As mentioned previously, there are existing rules that are used to combine heads and their complements (*head-comp*, *comp-head*, *head-comp-2*, and *comp-head-2*). The addition of *head-comp-lvc* and *comp-head-lvc* creates new avenues for light verbs and coverbs to combine but does not prevent the other head-complement rules from being used to combine light verbs and coverbs. To do this, the [LVC *lv-none*] feature is required in the non-head daughter for each of the non-LVC head-complement rules (dependent on which rules a given language uses). This disallows a coverb from being used as the non-head daughter in any of these rules.

Additionally, some languages allow any element to intervene between the light verb and coverb while other languages are more strict. In languages that are more strict, they might allow elements to intervene if they are part of the constituent that the coverb heads[13] or might not allow any element to intervene. For languages that don't allow any element to intervene (including when they are part of the constituent that the coverb heads), the [LIGHT

[13]This will be explained for noun coverbs in section 4.2.4 and for verb coverbs in section 4.2.5.

+] feature[14] is required on the head daughter of *head-comp-lvc* and/or *comp-head-lvc*. This only applies to languages without auxiliaries (since this feature would prevent auxiliaries from combining with light verbs).

Implementation

The rules and changes mentioned above are executed in the `create_lvc_phrase_types()` function in `light_verb_constructions.py`. In addition to adding the *head-comp-phrase-lvc* and *comp-head-phrase-lvc* types to the specific language TDL file, they are also added to the rules file (as *head-comp-lvc* and *comp-head-lvc*, respectively) in order to make them available.

4.2.3 Light Verbs

Analysis

My analysis accounts for three different kinds of light verb properties: bleached vs. unbleached, intransitive vs. transitive, and those that combine with noun coverbs vs. those that combine with verb coverbs. A light verb will have an attribute for each of these properties. For example, a light verb can be unbleached, intransitive, and combine with noun coverbs. Currently, my analysis only allows for bleached light verbs to combine with verb coverbs. This is because the semantic information in these constructions comes from the coverb, including the event of the predication. Since nouns cannot contribute an event, the coverb must be a verb.[15] I defined six different lexical types that a light verb could inherit from, each a different combination of the attributes possible for each property. A summary of these types with their corresponding properties can be found in table 4.1.

The original lexical types that I created had a lot of redundancy. For example, the

[14] This feature is used when handling phenomena that involve distinguishing words (or word-like small phrases) from phrases (Abeillé, Godard & Müller, 2003).

[15] Future work could look into possible ways of allowing bleached light verbs to work with noun coverbs. For example, taking a noun coverb and denominalizing it so its lexical category is now a verb would allow for an event to be provided.

Type name	Bleached?	Intransitive or Transitive?	Noun or Verb Coverb?
intrans-noun-lv-lex	no	intrans	noun
intrans-verb-lv-lex	no	intrans	verb
trans-noun-lv-lex	no	trans	noun
trans-verb-lv-lex	no	trans	verb
bleached-intrans-verb-lv-lex	yes	intrans	verb
bleached-trans-verb-lv-lex	yes	trans	verb

Table 4.1: Light Verb Feature Structure Combinations

intrans-noun-lv-lex and *trans-noun-lv-lex* types both required that the coverb be a noun, which involved both syntactic and semantic features. To reduce these kinds of redundancy, I factored out parts of the feature structures based on the properties. This resulted in multiple lexical types, which act as supertypes of the six types shown in table 4.1. Most of the syntactic structure of light verbs comes from the *verb-lex* or *main-verb-lex* (if auxiliaries are present in the language) types that are created for verbs (the light verb types inherit from one of these two types). The types that I created introduce mainly semantic information which, for bleached and unbleached light verbs, is very different. Due to this, there wasn't much redundancy to factor out for the other two properties between bleached light verb types and their unbleached counterparts. For example, bleached light verbs that combine with verb coverbs involve very different semantics from their unbleached counterparts. As mentioned earlier with the Japanese example, the light verb does not contribute anything semantic to the construction and, instead, the coverb does that. Therefore, I decided to create lexical types for the intransitive, transitive, and verb coverb subtypes different from the unbleached ones. This results in two subhierarchies: one for unbleached light verbs which inherit from *lv-lex* and one for bleached light verbs which inherit from *bleached-lv-lex*. I was able to factor out some of the redundancy between the transitive and intransitive lexical entries for bleached and unbleached verbs. These are factored out into *basic-intrans-lv-lex* (inherited by *intrans-lv-lex* and *bleached-intrans-lv-lex*) and *basic-trans-lv-lex* (inherited by *trans-lv-lex* and *bleached-trans-lv-lex*). The full hierarchy for the LVC-related types are shown in figure 4.8.

The *lv-lex* lexical type is inherited by all unbleached light verbs. It contains features that apply to all unbleached light verbs and sets up a majority of the semantics for the construction. *lv-lex* inherits either from *verb-lex* or *main-verb-lex* (which, as mentioned previously, depends on the presence of auxiliaries) and its AVM is shown in (42). In my analysis, the coverb is the first complement of the light verb. In order to model this, the *lv-lex* type identifies the first element of the COMPS list with the second element of the ARG-ST list (marked with [1]). Next, it identifies the LTOPs of the light verb and coverb with one another (marked with [3]). For verb coverbs, this results in their LBL being shared with the LBL of the light verb.[16] Last, this type connects the ARG1 of the light verb with the INDEX (or ARG0) of the subject and with the XARG of the the coverb (marked with [4]). This accounts for some of the semantics discussed for the Bardi example above, particularly with respect to the coverb and light verb relations. This is what identifies the ARG0 of the coverb with the ARG2 of the light verb.

[16]This only happens with verb coverbs because the LTOP of an NP does not point to anything. Therefore, noun coverbs will have a different handle than the light verb.

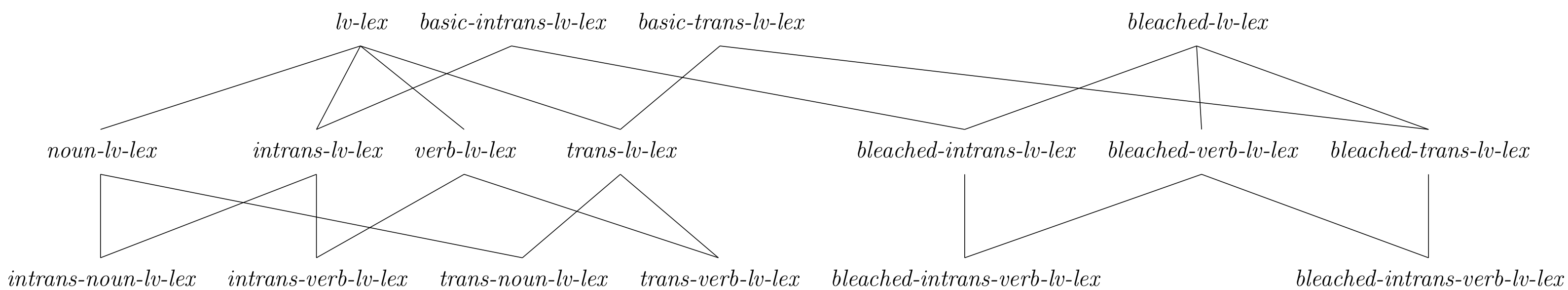


Figure 4.8: Hierarchy for LVC-related Types

(42)
$$\begin{bmatrix} \textit{lv-lex} & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT}\,|\,\text{VAL}\,|\,\text{COMPS} & \left\langle \boxed{1}, \ldots \right\rangle \\ \text{CONT}\,|\,\text{HOOK} & \begin{bmatrix} \text{CLAUSE-KEY} & \boxed{2} \\ \text{LTOP} & \boxed{3} \end{bmatrix} \end{bmatrix} \\ \text{LKEYS}\,|\,\text{KEYREL}\,|\,\text{ARG1} \ \ \boxed{4} & \end{bmatrix} \\ \text{ARG-ST} & \left\langle \begin{array}{l} \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \textit{cat-sat} \\ \text{CONT}\,|\,\text{HOOK} & \begin{bmatrix} \text{INDEX} & \boxed{4}\ \textit{ref-ind} \\ \text{ICONS-KEY}\,|\,\text{IARG1} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \\ \boxed{1} \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \textit{cat-sat} \\ \text{CONT}\,|\,\text{HOOK} & \begin{bmatrix} \text{ICONS-KEY}\,|\,\text{IARG1} & \boxed{2} \\ \text{XARG} & \boxed{4} \\ \text{LTOP} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}, \ldots \end{array} \right\rangle \end{bmatrix}$$

Next, unbleached intransitive light verbs inherit from *intrans-lv-lex* which inherits from *basic-intrans-lv-lex*. The *basic-intrans-lv-lex* type inherits from *non-mod-lex-item* ant its AVM for is shown in (43). The only thing this type does is make sure that only one element is on the comps list (the coverb). The *intrans-lv-lex* type does not contribute any additional information and therefore does not have an AVM associated with it. The reason there are two types here is to coordinate with how light verbs are added to the lexical type hierarchies in the Grammar Matrix customization system. This makes sure that mophotactic types are properly added to the correct light verb-related types, if necessary (this will be discussed more in section 4.2.6).

(43)
$$\begin{bmatrix} \textit{basic-intrans-lv-lex} & \\ \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CAT}\,|\,\text{VAL}\,|\,\text{COMPS} & \left\langle [\ ] \right\rangle \end{bmatrix}$$

In order to account for unbleached transitive light verbs, I define two lexical types that parallel the intransitive ones: *basic-trans-lv-lex* and *trans-lv-lex*. Similarly, *trans-lv-lex* does not contribute any additional information. As mentioned in section 3.2, complements of the light verb can, in some languages, be analyzed as the complement of the coverb. In my analysis, this is not possible and complements of the light verb can only be realized as such. Therefore, constraints on the object are indicated on the light verb's lexical type. The *basic-trans-lv-lex* type inherits from *non-mod-lex-item*, *non-local-none-no-hcons*, and *basic-icons-lex-item* and its AVM is shown in (44). The main purpose of this type is to provide information about the object in the sentence. The object must be specifier empty, which will make sure that it is a noun phrase (NP). Additionally, the ARG3 of the light verb is identified with the INDEX of the object (marked with $\boxed{1}$).[17] The light verb also has the feature [LIGHT +], which is part of what prevents the light verb from combining with anything else in the syntax before the coverb and reduces ambiguity.

(44)
$$\begin{bmatrix} \textit{basic-trans-lv-lex} & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT}|\text{VAL}|\text{COMPS} & \left\langle [\,], \begin{bmatrix} \text{OPT} & - \\ \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \textit{cat-sat} & \\ \text{VAL}|\text{SPR} & \langle\,\rangle \end{bmatrix} \\ \text{CONT}|\text{HOOK} & \begin{bmatrix} \text{INDEX} & \boxed{1}\ \textit{ref-ind} \\ \text{ICONS-KEY}|\text{IARG1} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \\ \text{CONT}|\text{HOOK}|\text{CLAUSE-KEY} & \boxed{2} \end{bmatrix} \\ \text{LKEYS}|\text{KEYREL}|\text{ARG3} & \boxed{1} \\ \text{LIGHT} & + \end{bmatrix} \end{bmatrix}$$

Now that unbleached light verbs can select for the correct number of arguments, they need to select for the correct type of coverb (noun or verb). The *noun-lv-lex* type is used for unbleached light verbs that combine with noun coverbs. It inherits from *lv-lex* and its AVM is shown in (45). This type identifies the ARG2 of the light verb with the INDEX of the coverb, which has a type of *ref-ind* (marked with $\boxed{2}$). Nominal elements are able to have type *ref-ind* while verbal ones cannot, which will prevent a light verb inheriting from this type from combining with verb coverbs.

[17] It was discovered after the implementation stage that this is not completely correct. This will work for unbleached light verbs but not for bleached ones. In bleached light verbs, the INDEX of the object should be identified with the ARG2 of the coverb (this could be done by having *bleached-trans-lv-lex* do argument composition or by having the object directly combine with the coverb). Identifying the INDEX of the object with the ARG3 of the light verb should be moved from *basic-trans-lv-lex* to *trans-lv-lex*.

(45)
$$\begin{bmatrix} \textit{noun-lv-lex} & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} \mid \text{CAT} \mid \text{VAL} \mid \text{COMPS} & \left\langle \boxed{1}, \ldots \right\rangle \\ \text{LKEYS} \mid \text{KEYREL} \mid \text{ARG2} & \boxed{2} \end{bmatrix} \\ \text{ARG-ST} & \left\langle [\,], \boxed{1} \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \textit{cat-sat} & \\ \text{VAL} \mid \text{SPR} & \langle\,\rangle \end{bmatrix} \\ \text{CONT} \mid \text{HOOK} \mid \text{INDEX} & \boxed{2}\ \textit{ref-ind} \end{bmatrix} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}$$

The *verb-lv-lex* type is used for unbleached light verbs that combine with verb coverbs. It inherits from `lv-lex` and its AVM is shown in (46). The ARG2 of the light verb is identified with the INDEX of the coverb, which has the *event* type (marked with $\boxed{2}$). Contrary to nominal elements, only verbal ones can have type *event*. This will prevent a light verb inheriting from this type from combining with verb coverbs.

(46)
$$\begin{bmatrix} \textit{verb-lv-lex} & \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} \mid \text{CAT} \mid \text{VAL} \mid \text{COMPS} & \left\langle \boxed{1}, \ldots \right\rangle \\ \text{LKEYS} \mid \text{KEYREL} \mid \text{ARG2} & \boxed{2} \end{bmatrix} \\ \text{ARG-ST} & \left\langle [\,], \boxed{1} \begin{bmatrix} \text{LOCAL} \mid \text{CONT} \mid \text{HOOK} \mid \text{INDEX} & \boxed{2}\ \textit{event} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}$$

With these types, sentences with unbleached light verbs can be modeled. This includes the Bardi and English sentences from the beginning of this chapter. In the Bardi sentence, the light verb *innyanana* "catch" would inherit from *trans-noun-lv-lex*, which would inherit from *noun-lv-lex* and *trans-lv-lex.* In turn, *noun-lv-lex* would inherit from *lv-lex* and *trans-lv-lex* would inherit from *lv-lex* and *basic-trans-lv-lex.* In the English sentence, the light verb *takes* would inherit from *intrans-noun-lv-lex* (for the types this would inherit from, refer to figure 4.8).

In order to model bleached light verbs, I created the *bleached-lv-lex* type, which is inherited by all bleached light verbs. Similar to *lv-lex*, *bleached-lv-lex* contains features that apply to all bleached light verbs and sets up a majority of the semantics for the construction. It inherits from *verb-lex* and its AVM is shown in (47). For my analysis of a bleached light verb, in addition to specifying that the coverb is the first complement of the light verb (marked with $\boxed{2}$), it also specifies that the subject is the first element of the ARG-ST list (marked with $\boxed{1}$).[18] As for the semantics, in my analysis, bleached light verbs should not be contributing anything meaningful in the semantics, as that is the coverb's job. Therefore, instead of identifying ARG1 of the light verb with the INDEX of the subject, *bleached-lv-lex* identifies the ARG1 of the coverb with the INDEX of the subject (marked with $\boxed{4}$). It also identifies the entire HOOK of the light verb with the HOOK of the coverb (marked with $\boxed{3}$). Additionally, my analysis of bleached light verbs relies on the existence of *main-verb-lex*. This type is created when auxiliaries are present in a language as a subtype of *verb-lex*. When auxiliaries exist, many of the features and constraints in *verb-lex* get moved to *main-verb-lex*. Bleached light verbs rely on the *verb-lex* type with fewer features and constraints. For languages that don't have auxiliaries but do have bleached light verbs, the LVC library will provide *main-verb-lex*. In order to make sure that bleached light verbs select for the correct number of arguments, I use the *bleached-intrans-lv-lex* and *bleached-trans-lv-lex* types, which inherit from *basic-intrans-lv-lex* and *basic-trans-lv-lex*, respectively. Neither of these types contribute any additional information and exist for the same reasons given for the existence of *intrans-lv-lex* and *trans-lv-lex*.

(47)
$$\begin{bmatrix} \textit{bleached-lv-lex} & \\ \text{SYNSEM} \mid \text{LOCAL} & \begin{bmatrix} \text{CAT} \mid \text{VAL} & \begin{bmatrix} \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2}, \ldots \rangle \end{bmatrix} \\ \text{CONT} \mid \text{HOOK} & \boxed{3} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1} \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \textit{cat-sat} \\ \text{CONT} \mid \text{HOOK} \mid \text{INDEX} & \boxed{4}\ \textit{ref-ind} \end{bmatrix} \end{bmatrix}, \boxed{2} \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CAT} & \textit{cat-sat} \\ \text{CONT} \mid \text{HOOK} & \boxed{3} \end{bmatrix} \\ \text{LKEYS} \mid \text{KEYREL} \mid \text{ARG1} & \boxed{4} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}$$

[18] This is because the *verb-lex* type it inherits from does not always identify the subject with the first element of the ARG-ST list. When auxiliaries are present in a language, this functionality is moved to *main-verb-lex*, which *bleached-lv-lex* does not inherit from.

As mentioned previously, my analysis only allows for bleached light verbs to combine with verb coverbs. The `bleached-verb-lv-lex` type inherits from `bleached-lv-lex` and its AVM is shown in (48). This type specifies that the INDEX of the coverb has the *event* type (i.e. requiring the coverb to be a verb).

(48)
$$\begin{bmatrix} \textit{bleached-verb-lv-lex} & \\ \text{SYNSEM}\,|\,\text{LOCAL}\,|\,\text{CAT}\,|\,\text{VAL}\,|\,\text{COMPS} & \left\langle \boxed{1}, \ldots \right\rangle \\ \text{ARG-ST} & \left\langle [\;], \boxed{1} \begin{bmatrix} \text{LOCAL}\,|\,\text{CONT}\,|\,\text{HOOK}\,|\,\text{INDEX} & \textit{event} \end{bmatrix}, \ldots \right\rangle \end{bmatrix}$$

With these types, sentences with bleached light verbs can be modeled. This includes the Japanese sentence from the beginning of this chapter. In that sentence, the light verb *shita* "do" would inherit from *bleached-intrans-verb-lv-lex* (for the types this would inherit from, refer to figure 4.8.

Now that all of these types have been defined, the six types from table 4.1 can be created. Then, for each light verb that a language has, a type can be created that inherits from one of these six types. This type will specify a value of LVC on its first complement (which would be the coverb). This restricts which coverbs the light verb can combine with. The specific light verb type can additionally contain information about other features of this light verb (e.g. case, TAM, inflection flags, etc.). For example, in the Bardi example from the beginning of the chapter, the light verb *innyana* "catch" specifies that the coverb must have [LVC *lv-nya-tr-noun-trans*], that the subject must have ergative case, and that the object must be one of four cases (absolutive, allative, locative, or instrumental).

There are also additional features that can be added to these types in order to constrain for certain language-dependent functionality. For *lv-lex* and *bleached-lv-lex*, they specify the head type for the subject, which is dependent on whether a language uses case-marked adpositions (e.g. *+np*, *adp*, *noun*). For *noun-lv-lex*, *verb-lv-lex*, and *bleached-verb-lv-lex*, there are some languages that allow coverbs to take on dependents similar to their non-coverb counterparts and some that don't. For the languages that don't, certain features need to be constrained. This involves adding [LIGHT +] and [MODIFIED *notmod*] to the coverb (as shown in the AVM in (49)).

(49)
$$\left[\text{SYNSEM}\ \left[\begin{array}{ll}\text{LOCAL}\,|\,\text{CAT}\,|\,\text{VAL}\,|\,\text{COMPS} & \left\langle \boxed{1}, \ldots \right\rangle \\ \text{ARG-ST} & \left\langle [\ ], \boxed{1}\left[\begin{array}{ll}\text{LIGHT} & + \\ \text{MODIFIED} & \textit{notmod}\end{array}\right]\right\rangle\end{array}\right]\right]$$

Whereas the additions and changes mentioned above are to types relating specifically to light verbs, a change also had to be made to the `transitive-verb-lex` type. In order to prevent transitive verbs from treating a coverb as an object, they require the second element on their ARG-ST list (i.e. the coverb) to be type [LVC *lv-none*]. I made a similar change to the *case-marking-adp-lex* type. In order to prevent adpositions from being incorrectly used in the *head-comp-lvc* and *comp-head-lvc* rules, they are made type [LVC *lv-none*].[19]

As for light verb lexical entries, they consist of up to two features: the STEM and (optionally) a PRED value. Bleached light verbs do not contribute any semantic meaning to the construction and therefore have no predication. These entries inherit from one of the specific light verb lexical types (the ones that include information about the LVC type associated with this light verb).

Implementation

The TDL code for all of the lexical types mentioned above is stored in constants in `light_verb_constructions.py`. However, the types are added to the language TDL file in the `create_lv_lex_types()` function in `lexical_items.py`. These types are added if the user has selected that noun and/or verb coverbs in the language. However, depending on whether the user selected that intransitive and/or transitive light verbs are possible in the language, only the corresponding types will be added. Adding the head type for the subject (as mentioned above) is also done in this function. The `customize_light_verbs()` function

[19] It was discovered during testing that this is not the best solution as adpositions can combine with coverbs, and the resulting adposition phrases can't be [LVC *lv-none*] so that they can still combine with light verbs.

(also in `lexical_items.py`) calls this function and makes the change to the *transitive-verb-lex* type (also mentioned above). `create_lv_lex_types()` additionally creates the individual lexical types for each light verb that the user defines. A type is created for each light verb, which follows the schema in (50).

(50) [{lv-name}](-bleached)-[intrans/trans]-[noun/verb]-lv-lex

For example, a transitive light verb that allows noun coverbs with the name *lv1* would have the type *lv1-trans-noun-lv-lex* and a bleached intransitive light verb that allows verb coverbs with the name *lv2* would have the type *lv2-bleached-intrans-verb-lv-lex*. The LVC type for this light verb (that a coverb can select for) is also added. An example lexical type for the Bardi intransitive light verb *nya* "to catch" (which allows verb coverbs, requires absolutive case on its subject, and has the name *nya-it*) is shown in (51). This type inherits from *intrans-verb-lv-lex*.

(51)
$$\begin{bmatrix} \textit{nya-it-intrans-verb-lv-lex} & \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{CAT} \mid \text{VAL} & \begin{bmatrix} \text{COMPS} & \left\langle \begin{bmatrix} \text{LOCAL} \mid \text{CAT} \mid \text{HEAD} \mid \text{LVC} & \textit{lv-nya-it-intrans} \end{bmatrix} \right\rangle \\ \text{SUBJ} & \left\langle \begin{bmatrix} \text{LOCAL} \mid \text{CAT} \mid \text{HEAD} \mid \text{CASE} & \textit{abs} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Whether or not coverbs take on normal dependents associated with their head type is dependent on the user's choices when filling out the questionnaire. If they select that noun or verb coverbs take noun-normal or verb-normal dependents, no additional features are added. Conversely, if they select that these coverbs don't take on these dependents, the features specified in (49) are added to the respective coverb types (*noun-lv-lex* for noun coverbs, *verb-lv-lex* for verb coverbs, and *bleached-verb-lv-lex* for verb coverbs with bleached light verbs). This is done in the `add_lvc_phrase()` function in `light_verb_constructions.py`. This function gets called as a result of the `customize_lvc()` function being called inside `customize.py`. Additionally, for languages that bleached light verbs but not auxiliaries, the `customize_auxiliaries()` function in `lexical_items.py` is called in order to get the correct lexical types added to the TDL files.

Light verb lexical entries are added to the lexicon in the `create_lv_lex_entries()` function in `lexical_items.py`. An example of an unbleached light verb lexical entry can be found in (52). This is for the Persian light verb *dād* "to give", which inherits from the *dadan-past-trans-noun-lv-lex* type. An example of a bleached light verb can be found in (53). This is for the Japanese light verb *shi* "to do", which inherits from *shi-it-bleached-intrans-verb-lv-lex*.

(52) $\begin{bmatrix} \textit{dād} & \\ \textsc{stem} & \left\langle \textit{"dād"} \right\rangle \\ \textsc{synsem}\,|\,\textsc{lkeys}\,|\,\textsc{keyrel}\,|\,\textsc{pred} & \textit{"\_give\_v\_lv\_rel"} \end{bmatrix}$

(53) $\begin{bmatrix} \textit{shi\_1} & \\ \textsc{stem} & \left\langle \textit{"shi"} \right\rangle \end{bmatrix}$

4.2.4 Noun Coverbs

Analysis

In my analysis, there are a few similarities between noun coverbs and regular nouns. Both require that the specifier — the first element on the ARG-ST list — is [HEAD *det*] and that the COMPS list is empty.[20] Additionally, noun coverbs will get the same value as regular nouns for OPT on the first element in their SPR list (either + or −) (this is to account for optional, required, and impossible determiners). However, whereas noun coverbs don't say anything about the subject,[21] their regular noun counterparts are SUBJ[22] empty. Additionally, noun coverbs specify that the MOD list is empty. Noun coverbs inherit from the *coverb-noun-lex* type, which inherits from *basic-noun-lex*. The AVM for *coverb-noun-lex*can

[20]This is not true for all languages, as specifying that the COMPS is empty blocks the noun from taking complements. Future work could further investigate complement-taking noun coverbs.

[21]The coverb does not say anything about its subject. This is to allow for the possibility of identifying the subject of the light verb with the subject of the coverb. As mentioned in the literature review chapter, there is an analysis of coverbs that allows it to constrain the subject. This is one avenue with which that could be possible. Fully implementing this is left to future work.

[22]I discovered after the implementation phrase that noun coverbs should also be SPEC empty. With the amount of similarities between noun coverbs and regular coverbs, future work could look into factoring these similarities out into a common supertype that both *coverb-noun-lex* and *noun-lex* inherit from.

be found in (54), which does not include the OPT feature.

(54)
$$\begin{bmatrix} \textit{coverb-noun-lex} & \\ \text{SYNSEM} \mid \text{LOCAL} \mid \text{CAT} & \begin{bmatrix} \text{HEAD} \mid \text{MOD} & \langle\ \rangle \\ \text{VAL} & \begin{bmatrix} \text{COMPS} & \langle\ \rangle \\ \text{SPR} & \left\langle \boxed{1} \begin{bmatrix} \text{LOCAL} \mid \text{CAT} \mid \text{HEAD} & \textit{det} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \text{ARG-ST} & \left\langle \boxed{1}, \ldots \right\rangle \end{bmatrix}$$

In the Grammar Matrix, multiple lexical entries can inherit from the same lexical type. In my analysis, groups of coverbs can inherit from a single lexical type, which share which light verbs they can combine with. This specific type inherits from *coverb-noun-lex* and specifies the LVC type for that group of coverbs. If the coverbs that inherit from this lexical type can also be regular nouns, both the specific coverb type and the specific noun type inherit from an intermediate type that contains any shared features. Noun coverb lexical entries consist of a STEM and a PRED value. They inherit from one of the specific coverb types. Additionally, the regular noun lexical type *noun-lex* is given the [LVC *lv-none*] feature to disallow its use as a coverb.

Implementation

The noun coverb type is stored as a constant in `light_verb_constructions.py` but is added to the language TDL file in the `add_initial_noun_coverb_lex_types()` function in `lexical_items.py`. The value for OPT is also added here as is making *noun-lex lv-none*. For each coverb group, a lexical type is created that inherits from *coverb-noun-lex* (and optionally an intermediate type as mentioned above). It follows the schema in (55).

(55) [{coverb-name}]-coverb-noun-lex

For example, for a coverb with the name *noun1*, its type would be *noun1-coverb-noun-lex*. Additionally, for each light verb that can combine with a given coverb, a lexical type is created that inherits from the coverb-specific lexical type. It follows the schema in (56).

(56) [{coverb-name}]-[{lv-type}]-coverb-noun-lex

For example, for a Bardi coverb with the name *noun10* and that has a LVC value of *lv-nya-tr-noun-trans*,[23] the type would be *noun10-lv-nya-tr-noun-trans-coverb-noun-lex*. This type, which inherits from *noun10-coverb-noun-lex*, is shown in (57). Noun coverb lexical entries are added to the lexicon in the `customize_nouns()` function in `lexical_items.py`. An example lexical entry for the Bardi noun coverb *liyan* "heart", which inherits from the *noun10-lv-nya-tr-noun-trans-coverb-noun-lex* type, can be found in (58).

(57) [*noun10-lv-nya-tr-noun-trans-coverb-noun-lex*
SYNSEM | LOCAL | CAT | HEAD | LVC *lv-nya-tr-noun-trans*]

(58) [*liyan*
STEM ⟨ *"liyan"* ⟩
SYNSEM | LKEYS | KEYREL | PRED *"_heart_n_rel"*]

For each noun lexical type that the user defines when filling out the questionnaire, they have the option of specifying it as being only or optionally a coverb. For noun lexical types that are only coverbs, only lexical types and entries associated with coverbs will be created. For noun lexical types that are optionally coverbs, lexical types and entries associated with regular nouns are additionally created.

4.2.5 Verb Coverbs

Analysis

There are two main verb coverb types, one for intransitive verb coverbs and one for transitive verb coverbs. There isn't a relationship between the transitivity of the coverb and the light verb (i.e. a transitive light verb can combine with an intransitive coverb). The *coverb-intrans-verb-lex* type inherits from *intransitive-verb-lex* and the *coverb-trans-verb-lex*

[23] This is for the Bardi transitive light verb *nya* "to catch" that allows noun coverbs.

type[24] inherits from *transitive-verb-lex*. No additional features are added to either type. Similar to noun coverbs, a specific type is created for each group of coverbs that inherits from *coverb-verb-lex*. This type specifies the LVC type that corresponds to the light verbs that the coverbs in the group can combine with. An intermediate type that both the verb coverb and the regular verb inherit from is created for any coverb that can also be a regular verb. Verb coverb lexical entries consist of a STEM and a PRED value. They inherit from one of the specific coverb types.

Implementation

The verb coverb type is stored as a constant in `light_verb_constructions.py` but is added to the language TDL file in the `add_initial_verb_coverb_lex_types()` function in `lexical_items.py`. For each coverb, a lexical type is created that inherits from *coverb-verb-lex* (and optionally an intermediate type as mentioned above). It follows the schema in (59).

(59) [{coverb-name}]-coverb-verb-lex

For example, for a coverb with the name *verb1*, its type would be *verb1-coverb-verb-lex*. Additionally, for each light verb that can combine with a given coverb, a lexical type is created that inherits from the coverb-specific lexical type. It follows the schema in (60).

(60) [{coverb-name}]-[{lv-type}]-coverb-verb-lex

For example, for a Japanese coverb with the name *verb3* and that has a LVC value of *lv-shi-it-intrans*,[25] the type would be *verb3-lv-shi-it-intrans-coverb-verb-lex*. This type, which inherits from verb3-coverb-verb-lex, is shown in (61). Verb coverb lexical entries are added to the lexicon in the `customize_verbs()` function in `lexical_items.py`. An example lexical

[24]Ideally, coverbs should be able to take arguments, which is the reason this lexical type exists. However, I did not fully test the lexical type for transitive coverbs. Further work would need to be done in order to confirm the functionality of this type.

[25]This is for the Japanese bleached intransitive light verb *shi* "to do" that allows verb coverbs.

entry for the Japanese verb coverb *touchaku* "to arrive", which inherits from the *verb3-lv-shi-it-intrans-coverb-verb-lex* type, can be found in (62).

(61) $\begin{bmatrix} \textit{verb3-lv-shi-it-intrans-coverb-verb-lex} & \\ \textsc{synsem}\,|\,\textsc{local}\,|\,\textsc{cat}\,|\,\textsc{head}\,|\,\textsc{lvc} & \textit{lv-shi-it-intrans} \end{bmatrix}$

(62) $\begin{bmatrix} \textit{touchaku} & \\ \textsc{stem} & \left\langle \text{``\textit{touchaku}''} \right\rangle \\ \textsc{synsem}\,|\,\textsc{lkeys}\,|\,\textsc{keyrel}\,|\,\textsc{pred} & \text{``\textit{\_arrive\_v\_rel}''} \end{bmatrix}$

Similar to noun coverbs, for each verb lexical type that the user defines when filling out the questionnaire, they have the option of specifying it as being only or optionally a coverb. For verb lexical types that are only coverbs, only lexical types and entries associated with coverbs will be created. For verb lexical types that are optionally coverbs, lexical types and entries associated with regular verbs are additionally created.

4.2.6 Changes to the Customization of Lexical Type Hierarchies

In order for morphology to work correctly with light verbs and coverbs, changes had to be made to the lexical type hierarchies. These changes are dependent on whether auxiliaries are present in the language or not, as the presence of auxiliaries adds the main verb *main-verb-lex* type. Normally, the intransitive verb *intransitive-verb-lex* and transitive verb *transitive-verb-lex* types inherit from *verb-ex* but, when auxiliaries are present, they inherit from *main-verb-lex*, which in turn inherits from *verb-lex* (compare figures 4.9a and 4.9b). When light verbs are present in a language, the *lv-lex* type is added as inheriting from *verb-lex* (or *main-verb-lex* if auxiliaries are present). Depending on whether the language allows intransitive light verbs, transitive light verbs, or both, the *intrans-lv-lex* and/or *trans-lv-lex* types are added as inheriting from *lv-lex*. This is shown in figures 4.9c and 4.9d. Additionally, if bleached light verbs are present in the language, the *bleached-lv-lex* type is added and inherits from *verb-lex*. Again, depending on the transitivity allowed for light verbs, the *bleached-intrans-lv-lex* and/or *bleached-trans-lv-lex* types are added inheriting from *bleached-lv-lex*. If only bleached light verbs are possible in a language, only *bleached-lv-lex* and its subtypes are added (shown in red in figure 4.9d). The *lv-lex* type and its subtypes (shown in blue in figure 4.9d) are not added. Coverbs are also added to the lexical type hierarchies. There are two kinds of verb coverbs, intransitive (*coverb-intrans-verb-lex*) and transitive (*coverb-trans-verb-lex*), which inherit from *intransitive-verb-lex* and *transitive-verb-lex*, respectively. Noun coverbs (*coverb-noun-lex*) inherit from the *noun-lex* type. These are shown in figures 4.9e and 4.9f. The lexical type hierarchies are created in the `lexical_type_hierarchy()` function in `lexicon.py`. Some additional work had to be done in order for coverbs to have the correct names so that they can be used in the lexical type hierarchies. This is done in the `fix_coverb_pc_inputs()` function in `morphotactics.py`.

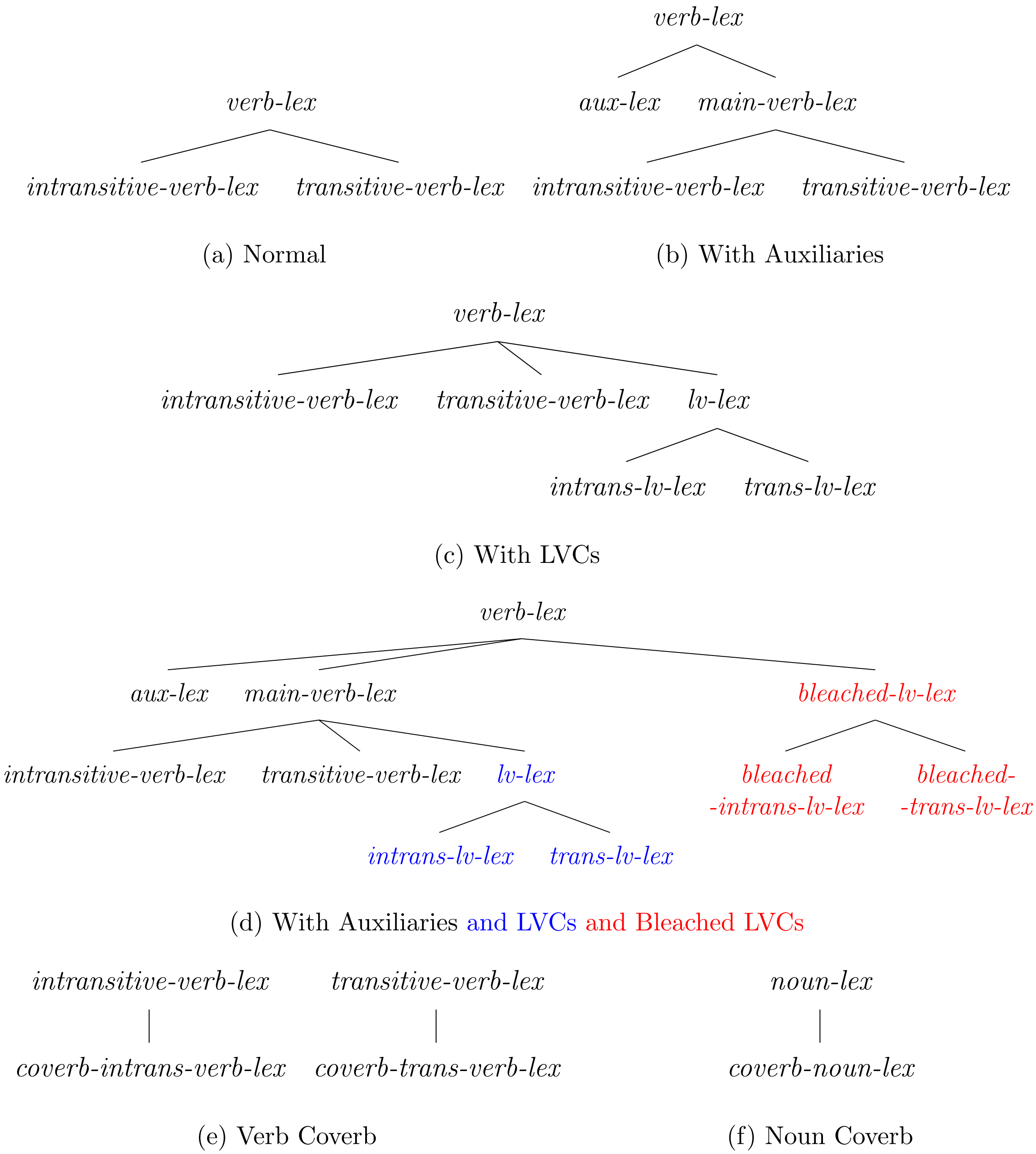
4.2.7 Summary

In this section, I explained the analysis behind adding light verb constructions to the LinGO Grammar Matrix customization system. I also provided my implementation of this analysis, giving reasoning for the decisions that I made. This included the addition of the LVC feature, new phrasal types as well as changes to existing ones, lexical types and entries for light verbs and coverbs, as well as changes that were made to the customization of lexical type hierarchies. In the next section, I will explain the front-end part of the LVC library.

4.3 Questionnaire

The questionnaire is the front-end for the Matrix customization system. Based on the answers to the questionnaire, the customization system generates a grammar that follows their answers in the questionnaire (this was the focus of the last section). In this section, I explain the changes I made to the Matrix questionnaire (for both the LVC and Lexicon subpages) and give an overview as to how each choice influences customization. The parts of the questionnaire relevant to the LVC library follow the range of variation for LVCs described at the end of chapter 3 and at the beginning of this chapter. I also explain how validation works for the questionnaire and describe the validation checks for the LVC library.

Figure 4.9: Lexical Type Hierarchies



(a) Normal

(b) With Auxiliaries

(c) With LVCs

(d) With Auxiliaries and LVCs and Bleached LVCs

(e) Verb Coverb

(f) Noun Coverb

4.3.1 LVC Subpage

In order to allow users to model LVCs in their language, I provide a subpage of the questionnaire specifically for LVCs. A screenshot of this subpage can be found in figure 4.10. Users can describe the kinds of coverbs that are possible in the language they are modeling: noun and/or verb coverbs. Selecting either one of these options (or both) will trigger the functions in the customization system that add LVC functionality to a grammar. This includes adding the LVC feature (see section 4.2.1), adding LVC phrasal types (see section 4.2.2), adding light verb lexical types (see section 4.2.3), and making changes to the customization of the lexical type hierarchies (see section 4.2.6). If noun coverbs are selected, their corresponding lexical types are added (see section 4.2.4. If verb coverbs are selected, their corresponding lexical types are added (see section 4.2.5). An additional question will be made visible for each coverb type selected, which allows the user to describe whether that coverb type takes normal dependents associated with their head type. Users can also describe the order of the coverb and light verb in their language. This will affect which phrase structure rules get added (see section 4.2.2). In the next question, the user is able to describe whether or not the coverb (or the constituent it heads) has to be immediately adjacent to the light verb. This is to determine whether or not to allow elements like subjects or objects to intervene between the coverb and light verb (see section 4.2.2). In the next question, the user can describe what valence options are possible for light verbs in their langauge. Depending on whether intransitive light verbs (meaning the light verb takes just the coverb as an argument), transitive light verbs (meaning the light verb takes both the coverb and an additional complement as arguments), or both are allowed, their respective lexical types are added (see sections 4.2.3 and 4.2.6). Lastly, the user can describe bleached verbs for their language (if they are possible). If they are possible, an additional question is made visible, where the user can state whether or not all light verbs are bleached. If bleached light verbs are possible, types associated with them are added (see sections 4.2.3 and 4.2.5). If all light verbs are bleached, unbleached light verb and coverb lexical types are not added. The choices that the user makes on this subpage is added to the language choices file. An example of what that looks like for the choices made in figure 4.10 can be found in figure 4.11.

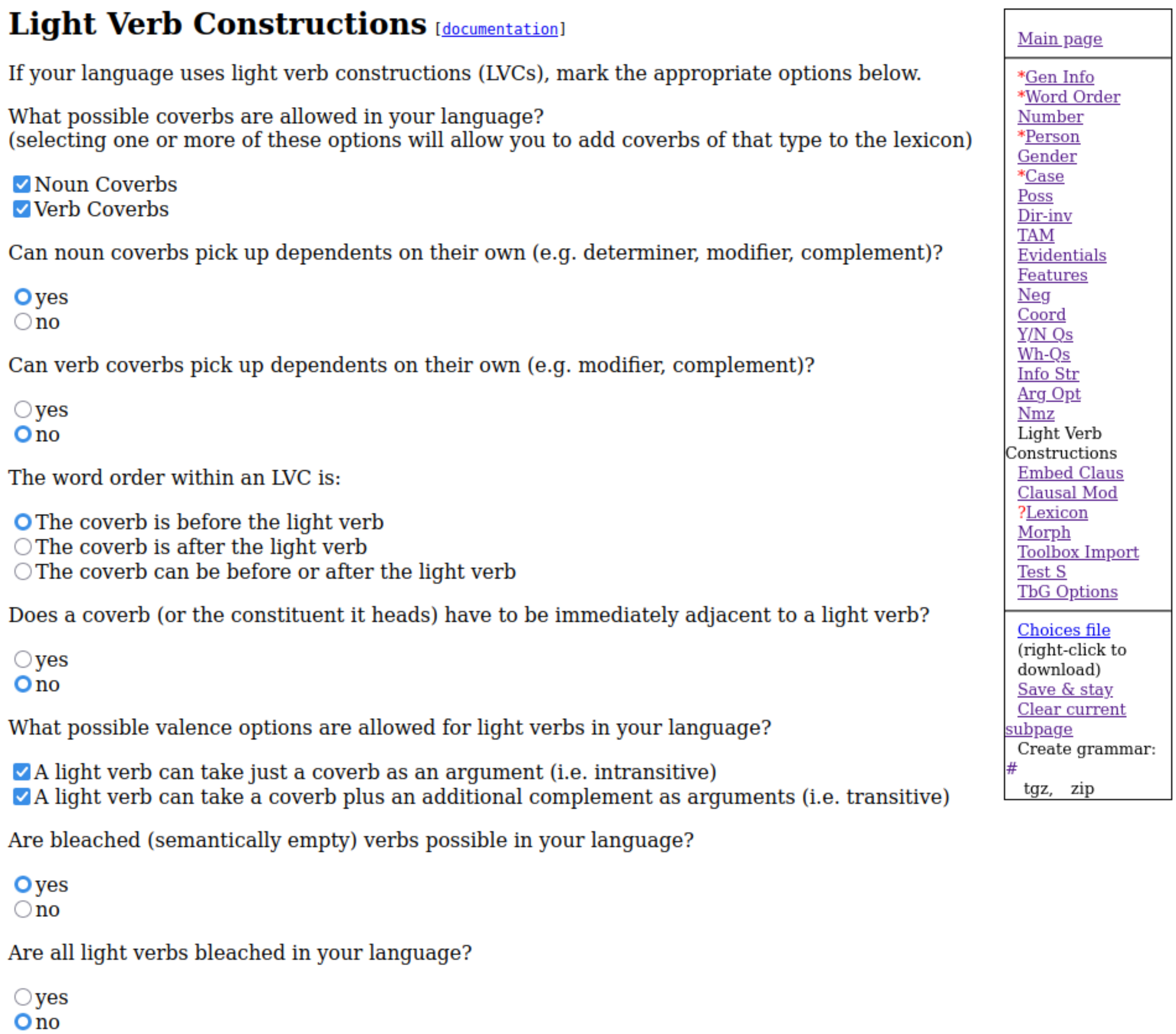


Figure 4.10: LVC Subpage

```
section=lvc
coverb-n=on
coverb-v=on
lvc-noun-cv-dep=yes
lvc-verb-cv-dep=no
lvc-word-order=cv-lv
lvc-adjacent=no
lvc-it=on
lvc-tr=on
lvc-bleached=yes
lvc-all-bleached=no
```

Figure 4.11: LVC Choices

4.3.2 Changes to the Lexicon Subpage

In addition to making choices about LVCs for the language, the user must also add coverbs and light verbs to the lexicon. This required making some changes to the Lexicon subpage. I decided to separate light verbs from heavy verbs when entering them into the Lexicon subpage. This is because there are some major differences between what the user can describe for a light verb as opposed to a heavy verb. For example, the mechanics for implementing the argument structure for a heavy verb are not the same as doing so for a light verb. Additionally, a lot of functionality that occurs when creating verb lexical types and entries is not the same as when creating light verb lexical types and entries. For these reasons, I implemented a new section for adding light verbs is at the bottom of the Lexicon subpage. A screenshot of that section can be found in figure 4.12. For each light verb that the user adds, they must specify a type name (this must be unique), a valence type (whether the light verb is intransitive or transitive), a coverb type (whether the light verb takes noun

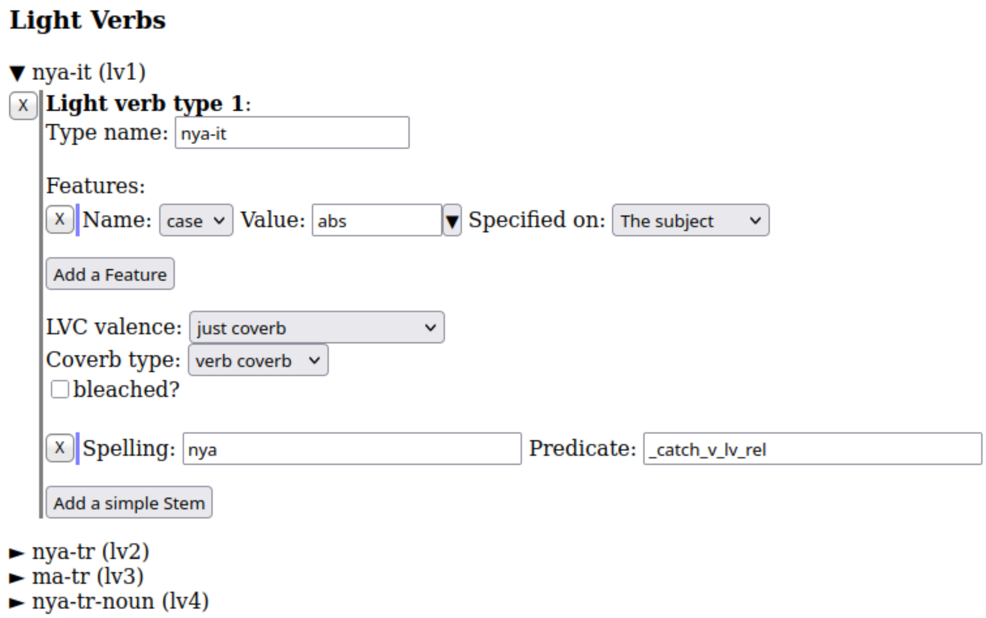
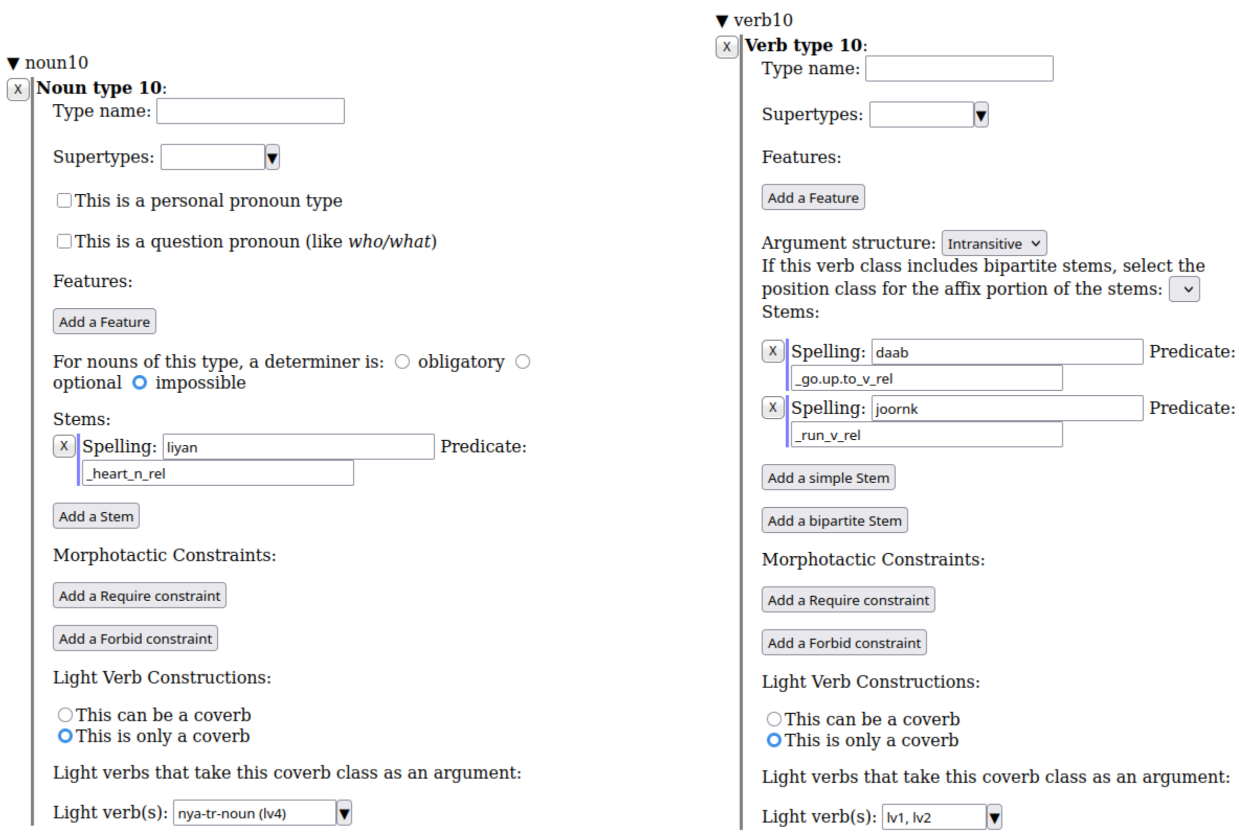


Figure 4.12: Lexcion Subpage: Light Verbs

or verb coverbs), whether the light verb is bleached or not, and at least one spelling for the light verb. The name that user gives to the light verb type, along with the valence, coverb type, is used when creating the LVC type name. If the light verb is bleached, the predicate field must be left blank. Otherwise, a predicate value must be provided. In addition to these required aspects, the user can optionally specify features of the light verb (as with most lexical types on this subpage).

Additionally, I debated on how to fit coverbs into the Lexicon subpage. Originally, I was going to have them entered separately, similar to the light verbs. However, I wanted to take advantage of the similarities between the coverbs and their corresponding non-coverb counterparts. Indicating that noun and/or verb coverbs are possible on the Lexcion subpage will make options visible for the noun lexical types and/or verb lexical types on the Lexicon subpage. These options allow the user to describe whether each noun or verb lexical type that they create is only a coverb or optionally a coverb (the options should be left blank if it is not a coverb). Selecting one of these options opens up a dropdown menu where the user can select which light verbs can combine with this coverb type. The combination of light verbs that the coverb lexical type can combine with affects how the type hierarchy that the LVC feature selects from is built. For example, if a coverb can combine with one light verb, a type will be created for that light verb and will be specified on the coverb. If a coverb can combine with two light verbs, a disjunctive type will be created for that light verb (and will also be specified on the coverb). For types that are optionally coverbs, two lexical types are created: one for the regular noun or verb and one for the coverb. Otherwise, just a coverb type is created. Images of what these parts of the Lexcion subpage look like for Bardi are shown in figure 4.13.



(a) Noun Coverbs (b) Verb Coverbs

Figure 4.13: Lexicon Subpage: Coverbs

4.3.3 Validation

Validation is performed on the questionnaire in an attempt to ensure that the grammars output by the customization system compile with the processing engines that use them. In other words, the grammars are well-formed and have (relatively) predictable behavior. The validation code checks for incompatible answers and unanswered questions. Validation is triggered whenever the user navigates between subpages or when the "Save & stay" button is clicked. There are two kinds of validation output: warnings and errors. A red question mark is used to indicate warnings and a red asterisk is used to indicate errors. With warnings, a grammar can still be generated by the customization system. With errors, however, the buttons to generate a grammar are grayed out. The errors must be fixed in order to generate the grammar. Hovering over a red question mark or red asterisk will provide information about the warning or error, which guides the user towards fixing it.

Validation for the LVC library is split into two parts: validation for the LVC subpage (found in the `validate_lvc()` function in `validate.py`) and validation for the Lexicon subpage (found in the `validate_lexicon()` function in `lexicon.py`). All of the validation checks for the LVC library result in errors, not warnings. The first level of validation on the LVC subpage involves making sure that, if either of the options for the first question (which asks if noun and/or verb coverbs are possible), the other questions on the subpage are filled out. The next level makes sure that at least one light verb and coverb are defined in the lexicon. If both noun and verb coverbs are possible, one of each is required. If bleached light verbs are possible, at least one light verb needs to indicate that it is bleached. Since my analysis only allows verb coverbs to combine with bleached light verbs, an additional check is made to make sure that the verb coverb option is selected when bleached light verbs are possible. On the Lexicon subpage, validation checks are performed on each noun coverb, verb coverb, and light verb type that is defined. For coverbs, this involves making sure that light verbs that the coverb type can combine with have been specified. A check is also performed to make sure that the light verb listed can combine with this type of coverb. For example, if a light verb specifies that it take noun coverbs, an error will appear if the user tries to specify it as a possible light verb for a verb coverb. For light verbs, validation involves making sure that all necessary fields have been filled out. The type name is checked to make sure it is unique in order to prevent errors resulting from conflicting type names. Additionally, a check is made to make sure that the options selected for valence, coverb type, and whether or not the light verb is bleached match the options selected on the LVC subpage. Lastly, if the user indicated, on the LVC subpage, that all light verbs are bleached, a check is made to make sure that all light verbs have the bleached option selected.

4.3.4 Summary

In this section, I discussed what changes and additions had to be made to the Matrix Questionnaire for the LVC library. This included adding a new subpage and making changes to the Lexicon subpage. Additionally, I added validation tests to prevent users from making or missing choices that would lead to errors or a prevent a grammar from compiling.

4.4 Summary

The focus of this chapter was my analysis and implementation for the LVC library. This included the customization system back-end and the front-facing Matrix questionnaire. I provided analyses for the lexical types and entries for light verbs and coverbs as well as the lexical rules that allow them to combine. I also discussed changes to other types that had to be made in order for the analysis to work. For each part of the analysis, I provided an explanation for how it was implemented in the back-end customization system code. Lastly, I explained the changes I made to the questionnaire and the types of validation checks I added. In the next chapter, I will explain how I tested this analysis and implementation.

Chapter 5

EVALUATION

In this chapter, I explain how I evaluated the light verb construction (LVC) library. Evaluation is conducted in order to test how the analysis and implementation perform with respect to coverage, well-formed MRSes, and generalization to languages not considered during development. In this process, languages are selected and, for each language, a choices file and testsuite are created. There are three different groups of languages that are used: illustrative, pseudo, and held-out. Both illustrative and held-out languages are real ones while pseudo-languages (section 5.2) are fake ones created to test certain combinations of choices. Illustrative languages (section 5.1) are used during the analysis and implementation while held-out languages (section 5.3) are only used after implementation is complete. I also performed an error analysis, where I document the bugs and errors that I encountered after implementation was complete (section 5.4). Lastly, a the overall results for each language is given in section 5.5.

The testsuites for each language contain both positive (grammatical) and negative (ungrammatical) examples. In order to verify that basic phenomena were input into the Matrix customization system correctly, same basic sentences (both positive and negative) are included in the testsuite. These will not be discussed at length for most languages to allow for more focus on the LVC examples. The examples used in testsuites are not always the original examples given by the linguists whose work the testsuites are based on. In order to showcase specific features or properties of LVCs, sentences might be simplified or reconstructed. However, the original work from which the example comes from is still cited.

5.1 Illustrative Languages

I used illustrative languages in the process of building my analysis and implementing said analysis. The goal with selecting illustrative languages is to find a large distribution across the phenomena being implemented in order to build a more well-rounded library. In my case, I tried to find languages with different kinds of LVCs, for example ones with different kinds of light verbs and coverbs and ones that have a range of properties and rules for what is and isn't allowed. I selected four languages: Bardi (bcj, Nyulnuylan), English (eng, Indo-European), Japanese (jpn, Japonic), and Persian (per, Indo-European). The reasons why each language was selected, other than the fact that each of them has LVCs, will be explained in the sections below. Additionally, I will give the results on how the LVC library performed on the testsuites for each language.

5.1.1 Bardi [bcj]

Bardi is an Australian Aboriginal language in the Nyulnyulan family spoken in north-western Australia (mostly at the tip of the Dampier Peninsula and on nearby islands). In 2021, the Australian Bureau of Statistics released the results of a cultural diversity census, which showed that 384 people speak Bardi (which is less than .01% of Australia's 25,422,788 population).[1] Bardi is an ergative-absolutive language with free word order. It is a highly agglutinative language where much of the affixation occurs on the verb (e.g. person and number of subjects and objects, tense, aspect, and mood) (Bowern, 2012).

One of the primary methods of forming predicates in Bardi is with LVCs, which is why I originally selected this language. I used Bardi a lot in my initial analysis of LVCs, especially with creating the lexical entries for light verbs and coverbs and with the rules that allow them to combine. Bardi allows both intransitive and transitive light verbs and can form

[1] This data can be found in Table 5 of the "Data table for Cultural diversity summary" download at `https://www.abs.gov.au/statistics/people/people-and-communities/cultural-diversity-census/2021`.

LVCs with noun, verb, and adjective coverbs[2] (neither noun nor verb coverbs can pick up any dependents). Examples of basic LVCs in Bardi can be found in (63a) and (63b) below. In (63a), the light verb *-nya-* "catch" combines with the verb coverb *joornk* to form the intransitive LVC "run away (quickly)". In (63b), the light verb *-nya* "catch" combines with the noun coverb *liyan* "heart" to form the transitive LVC "breathe", which takes *nimal* "through the nose" as an object. Although Bardi has free word order, coverbs must appear before the light verb. Therefore, the example in (63c) (where the verb coverb *daab* "go up to" appears after the light verb *-nya-* "catch") is ungrammatical. Additionally, the coverb (and the constituent it heads) must be immediately adjacent to the light verb.[3]

(63) a. Aamba joornk inyana.

aamba	joornk	i-nya-na
man.ABS	run	3-catch-REM.PST

'The man ran away (quickly).' [bcj] (Bowern, 2012, p. 511)

b. Aamba nimalnga liyan innyana.

aamba-nim	nimal-nga	liyan	i-n-nya-na
man-ERG	nose-INS	heart	3-TR-catch-PST

'The man breathed through his nose.' [bcj] (Bowern, 2012, p. 440)

c. *Aambanim innyana daab garrin daab.

aamba-nim	i-n-nya-na	daab	garrin
man-ERG	3-TR-catch-REC.PST	go.up.to	hill

'Intended: The man went up the hill.' [bcj] (Bowern, 2012, p. 227)

The testsuite I created for Bardi has ten sentences: seven positive and three negative. For the negative examples in the testsuite, the first is to make sure that features specified on the light verb (in this example features about the case of the subject) work properly. The

[2]Adjective coverbs are not covered in the current version of the LVC library.

[3]I did not include a sentence for this in the testsuite. However, I tested two sentences independently, one where the subject intervened between the light verb and coverb and one where the object did. Neither of these sentences parsed, which is the expected behavior.

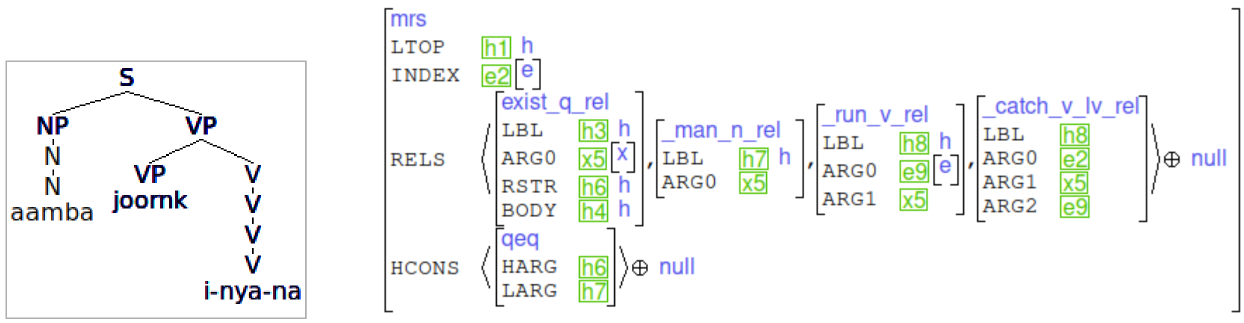


Figure 5.1: Parsing *Aamba joornk inyana*

other two negative examples involve constructions where the light verb precedes the coverb, which is not allowed in Bardi (as seen in (63c) above).

As for the positive examples, the first two sentences in the testsuite are for basic Bardi intransitive and transitive sentences. The next four sentences use verb coverbs in either intransitive or transitive constructions with different word orders. (63a) above is one of these sentences. It consists of a subject *aamba* "man", a verb coverb *joornk* "run", and an intransitive light verb *nya* "to catch". The combined LVC has the meaning "to run away". When parsed, it has one parse, which is represented by the tree and MRS shown in figure 5.1.[4] The MRS shows that the coverb and the light verb share a subject (represented by *x5* which is the ARG0 of *_man_n_rel*), indicated by the *x5* instance as the ARG1 of both *_run_v_rel* and *_catch_v_lv_rel*. Additionally, the coverb and light verb share the value for the LBL feature (indicated by the *h8* handle). Conversely, the coverb and light verb introduce their own events (*e9* and *e2*, respectively).

The next sentence uses a noun coverb in a transitive LVC, which is (63b) above. This sentence was shown at the beginning of chapter 4, including the parse tree and MRS. This sentence consists of a subject *aamba* "man", an object *nimal* "nose", a coverb *liyan* "heart",

[4]The tree and MRS representation show the output of the LKB when parsing (63a) with a grammar for Bardi produced by the customization system. In the tree, the strings of Ns and Vs represent lexical rules being applied. Such structures will be displayed throughout this section in order to illustrate aspects of my implementation.

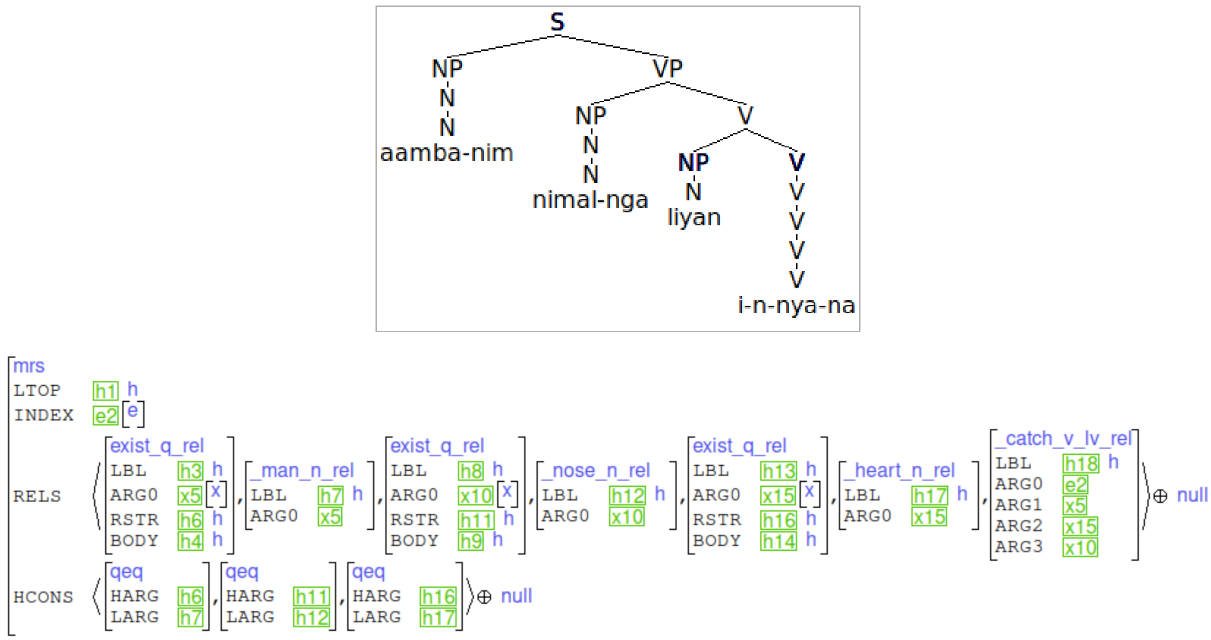


Figure 5.2: Parsing *Aamba nimalnga liyan innyana*

and a transitive light verb *nya* "to catch". The combined LVC has the meaning[5] "to breathe". As mentioned in chapter 4, this sentence results in two parses. Both have the same MRS but, in one of the parses, the light verb extraneously goes through the *dir-obj-no-drop-lex* lexical rule.[6] The correct tree and the MRS for this sentence is redisplayed in figure 5.2. The MRS shows that, unlike with the verb coverb, the noun coverb does not share a subject with the light verb and has no event (since nouns don't have subjects or events). The ARG1 of *_catch_v_lv_rel* is the subject (*x5*), the ARG2 is the coverb (*x15*), and the ARG3 is the object (*x10*).

[5]The LVC library does not support idiomatic meanings for LVCs. Future work could support a lookup table that finds the meaning of the full LVC based on the coverb and light verb combination.

[6]This ambiguity is outside the scope of my thesis, but it likely comes from a bug in the Bardi choices file I created and not from the LVC library.

5.1.2 English [eng]

English is an Indo-European language (specifically the Germanic branch) spoken in many countries around the world. As a first language, English is spoken primarily in North America, the United Kingdom, Australia, and New Zealand. Ethnologue estimates around 1.5 billion speakers around the world as of 2025.[7] English has SVO word order and, although there are some vestiges of a nominative-accusative case system, case is not actively marked on nouns (with the exception of personal pronouns).

Unlike Bardi, English does not rely heavily on LVCs to form most predicates. However, English LVCs allow for more internal modification than other languages, which is the reason why I selected it when building my analysis. In English, noun coverbs can pick up dependents (e.g. determiners, modifiers, and complements) on their own.[8] The coverb appears after the light verb and the constituent it heads must be immediately adjacent to the light verb, which can be intransitive or transitive. Some examples of LVCs in English are shown in (64).

(64) a. Jordan takes a shower.

Jordan	take-s	a	shower
Jordan	take.PRS-3SG	DET.INDF	shower

‘Jordan takes a shower.’ [eng]

b. Jordan takes a big survey of the participants.

Jordan	take-s	a	big	survey	of the	participant-s
Jordan	take.PRS-3SG	DET.INDF	big	survey	of DET.DEF	participant-PL

‘Jordan takes a big survey of the participants.’ [eng]

c. I have taken a bath.

I	have	take-n	a	bath
1SG	have.PRS	take-PST.PTCP	DET.INDF	bath

[7] This data can be found at `https://www.ethnologue.com/insights/ethnologue200/`.

[8] The current analysis only allows for coverbs to pick up determiners and modifiers. The Matrix customization system in general does not currently support complement-taking nouns.

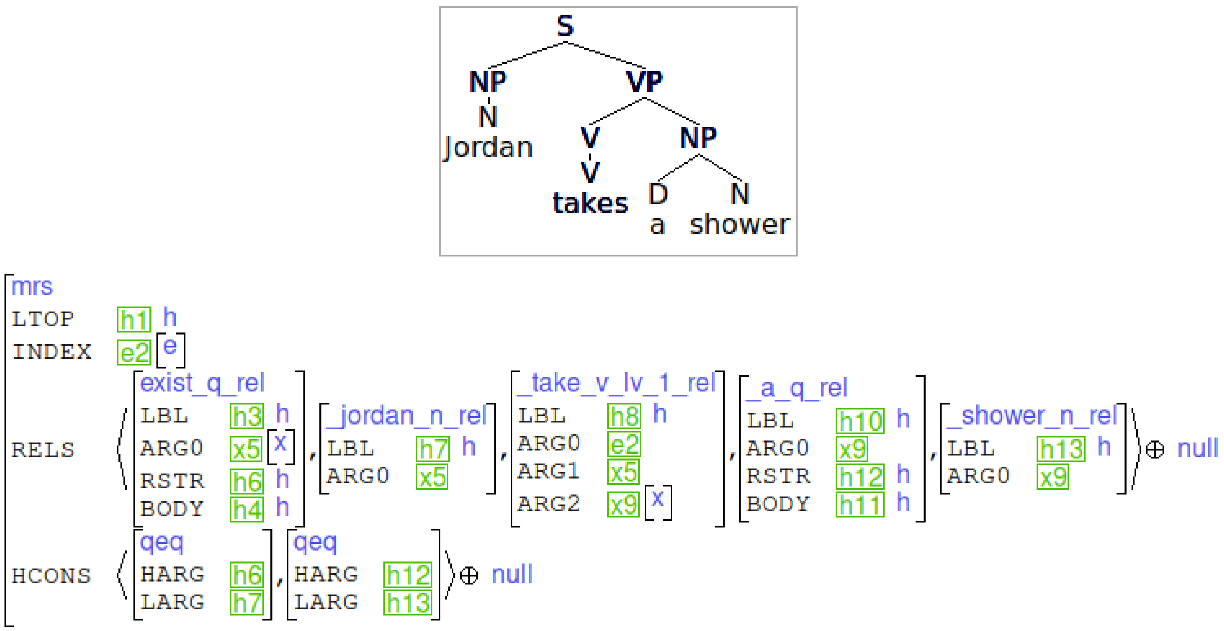
'I have taken a bath.' [eng]

The testsuite that I created for English has nine sentences: seven positive and two negative. The negative examples include LVCs with the coverb appearing before the light verb (one uses an intransitive light verb and the other uses a transitive one). The first two positive examples are for basic intransitive and transitive sentences. The other positive examples include basic LVCs as well as LVCs with prepositional phrases (which act as complements), adjectives, and auxiliaries. In (64a), the sentence consists of a subject *Jordan*, a light verb *take*, and a coverb *shower*. When parsed, it results in two distinct parses, both of which are correct. The first parse is the LVC, where the meaning of the LVC refers to the act of showering, while the second uses the heavy form of the verb, where the meaning refers to the act of physically picking up and taking a shower. Both trees look the same, however in the LVC one, the V and NP combine using *head-comp-lvc* (instead of *head-comp*). Additionally, *takes* is a light verb and *shower* is a coverb. This is more apparent when comparing the MRSes, where the relation for the light verb is *_take_v_lv_1_rel*, as opposed to *_take_v_rel* for the heavy verb. The tree and MRSes can be found in figure 5.3.

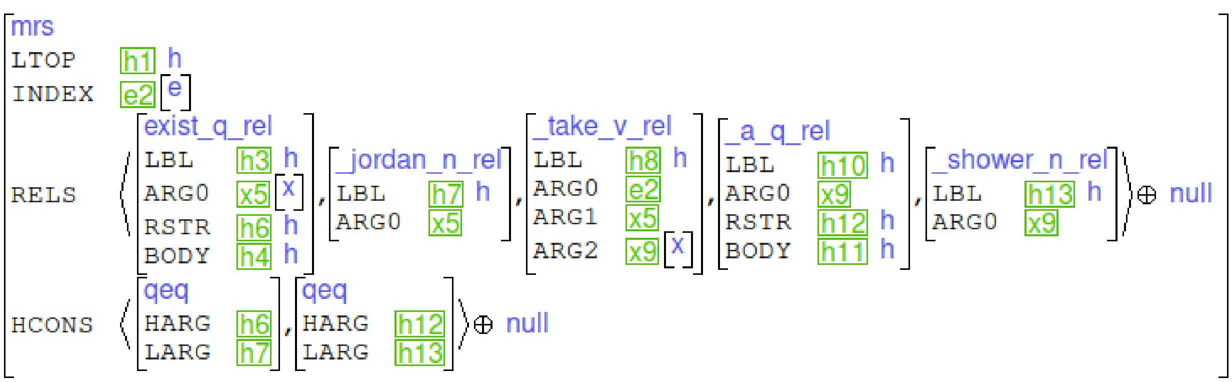
In (64b), there is an adjective (*big*) modifying the coverb (*survey*) and a prepositional phrase (PP) complement (*of the participants*). This sentence also includes the subject *Jordan* and the light verb *to take*. When parsed, it has four parses. In the first parse, the PP attaches to the S *Jordan takes a big survey* and the LBL on the preposition (P) *of* is the same as the LBL on the light verb *take*. In the second parse, the PP attaches to the N *survey* and the LBL on the P is the same as the LBL on the N. In the third parse, the PP attaches to N *big survey* and the LBL on the P is the same as the LBL on the N. In the fourth parse, the PP attaches to the VP *takes a big survey* and the LBL on the P is the same as the LBL on the light verb. In English, these parses are acceptable, since PPs and adjectives can attach in multiple places.[9]

(64c) shows an LVC with an auxiliary *have*. This sentence also includes the subject *I*,

[9]For languages where these types of ambiguity are not grammatical, further work would have to be done to constrain lexical types and/or rules. However, these constraints are outside the scope of this library.



(a) MRS using Light Verb *takes*



(b) MRS using Heavy Verb *takes*

Figure 5.3: Parsing *Jordan takes a shower*

the light verb *to take* and the coverb *bath*. When parsed, it has four parses, two of which are correct. In the two correct parses, one uses an LVC while the other uses a heavy verb. The two incorrect parses stem from ambiguity with the constraints on the auxiliary. In these incorrect parses, the auxiliary is allowed to combine with the light verb using HEAD-COMP-LVC rule when it should only be allowed to use the HEAD-COMP rule.[10] The MRS for the correct LVC parse is almost identical to the LVC parse in figure 5.3, with the only difference being the coverb relation (*_bath_n_rel* versus *_shower_n_rel*).

5.1.3 Japanese [jpn]

Japanese is in the Japonic language family and is spoken in Japan. As of the 2020 Population Census, there are 126,146,099 people in Japan, almost all of whom speak Japanese.[11] Japanese is a verb-final language with a nominative-accusative case system. Cases are indicated with various particle markers, which have been interpreted as suffixes or postpositions by Japanese linguists. Both methods are possible with the Matrix Customization System, however, I have made the choice to interpret them as postpositions.

Japanese only has one light verb *suru* "to do", which is bleached. It is this property of the light verb that led me to choose Japanese as an illustrative language. It has coverbs that are often treated as a hybrid category, which are called verbal nouns in the literature on the language. However, given the options available in the LVC library (that bleached light verbs can only combine with verb coverbs), I treat these verbal nouns as verb coverbs. These coverbs cannot pick up dependents on their own. The coverb (or the constituent it heads) must be immediately adjacent to the light verb, which can form intransitive or transitive constructions. The coverb must appear before the light verb in the construction. Some examples of Japanese LVCs are shown in (65).[12]

---

[10]The fix for this is described in section 5.4.

[11]This data can be found in Table 1-1 at `https://www.e-stat.go.jp/en/stat-search/files?page=1&layout=datalist&toukei=00200521&tstat=000001136464&cycle=0&year=20200&month=24101210&tclass1=000001136466`. This does not account for Japanese speakers elsewhere in the world.

[12]In Japanese, coverbs can also be marked for case. I did not include examples with where coverbs

(65) a. Hanako ga benkyou shita.

Hanako ga benkyou shi-ta
Hanako NOM study do-PST

'Hanako studied.' [jpn][13]

b. Taroo ga kurma de touchaku shita.

Taroo ga kurma de touchaku shi-ta
Taroo NOM car INS arrive do-PST

'Taroo arrived by car.' [jpn][14]

c. *Taroo ga hoeru to yakusoku Jirou ni shita.

Taroo ga hoeru to yakusoku Jirou ni shi-ta
Taroo NOM bark COMP promise Jirou DAT do-PST

'Intended: Taroo promised Jirou to bark.' [jpn]

The testsuite that I created for Japanese has nine sentences: six positive and three negative. The first two negative examples contain the incorrect order of light verb and coverb (one uses an intransitive light verb and the other uses a transitive one). The last negative example is seen in (65c). Here, the object *Jirou* is intervening between the coverb and the light verb, which Japanese does not allow. This sentence does not parse, which is correct. As for the positive examples, the first two are simple intransitive and transitive sentences. The other four sentences contain intransitive and transitive LVCs. One of these sentences is seen in (65a). When parsed, it produces one parse. The tree and MRS for this parse can be found in figure 5.4. This sentence consists of the subject *Hanako*, the verb coverb *benkyou* "to study", and the bleached light verb *shita* "to do". Since the light verb is bleached, it does not contribute a relation. Instead, the coverb contributes a relation and

take case-marking adpositions in my testsuite. During testing, I discovered an issue with case-marking adpositions for coverbs in Daasanach and subsequently learned they did not work in Japanese either. This is due to an extra constraint on the case-marking adposition lexical type, which is discussed in section 5.3.2.

[13]This example comes from the AGG data testsuite for Japanese, which is not publicly accessible.

[14]This example and (65c) come from the MRS testsuite at `https://github.com/delph-in/docs/wiki/MatrixMrsTestSuiteJa`.

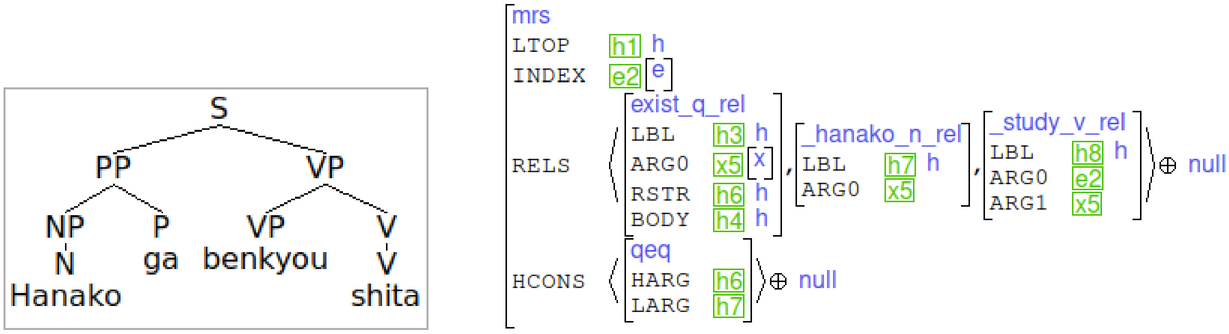


Figure 5.4: Parsing *Hanako ga benkyou shita*

is linked semantically to the subject, which is indicated by the `x5` instance as the ARG1 of *_study_v_rel*.[15] Additionally, the INDEX for the whole MRS points to the event that is the coverb's ARG0. This contrasts with an unbleached LVC (see figure 5.1) where the coverb introduces its own event.

(65b) is an example of an intransitive LVC with a postpositional phrase (PP) complement. It consists of a subject *Taroo*, a PP *kurma de* "by car", a coverb *touchaku* "to arrive", and the bleached light verb *shita* "to do". When parsed, this sentence results in one parse, which can be seen in the tree and MRS in figure 5.5. Like with the last sentence, the light verb contributes no predication in favor of the coverb holding the information about the predicate. Additionally, the preposition shares its LBL with the coverb (indicated by the `h13`).

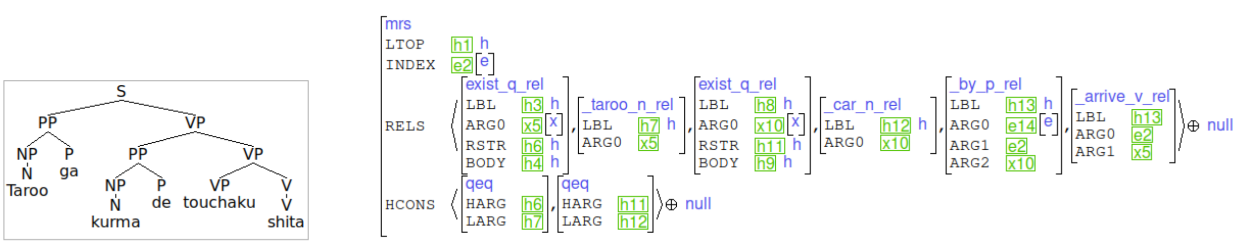


Figure 5.5: Parsing *Taroo ga kurma de touchaku shita*

[15]Technically, *benkyou* "study" is a transitive coverb, as it can take an ARG2 (which represents what is being "studied"). However, for the purposes of my testsuite, I treat it as an intransitive coverb.

5.1.4 Persian [per]

Persian (Farsi)[16] is in the Indo-Iranian branch of the Indo-European language family and is spoken in many parts of Europe and Asia but primarily in countries in the Middle East, Central Asia, and Russia. Due to how widespread of a language Persian is, it is hard to determine how many people speak it. Estimates put the number of speakers at around 170 million (60 million speak Persian as a native language and 110 million speak it as a second language) (Windfuhr, 2009). Persian is described as an SOV language with a nominative-accusative case system. However, Persian uses case markers for both direct and indirect objects and displays flexible word order.

Persian LVCs are interesting because they allow elements like the subject and object to intervene between the coverb and light verb. This corresponds to the Matrix LVC subpage option where the coverb and the constituent it heads does not need to be immediately adjacent to the light verb. To allow for this option, the word order was entered into the Matrix as free. Persian only allows noun coverbs which, furthermore, cannot have dependents. Despite generally free word order, the order of the light verb and coverb is fixed and the coverb must appear before the light verb. Additionally, both intransitive and transitive light verbs are allowed in Persian. Some examples of Persian LVCs can be found in (66).

(66) a. Omid sili zad.

Omid	sili	zad
Omid	slap	hit.PST

'Omid gave slaps. (lit. Omid slapped.)' [per] (Godard & Samvelian, 2021, p. 471)

b. Maryam bāqčerā āb dād.

Maryam	bāqče=rā	āb	dād
Maryam	garden=ACC	water	give.PST

[16] Following Ali Jahanshiri and the Academy of Persian Langauge and Literature, I use Persian (rather than Farsi) to refer to this language as the preferred and established exonym. The page on Ali Jahanshiri's website describing this is no longer publicly available but it was saved by the Internet Archive's Wayback Machine at `https://web.archive.org/web/20230813161409/https://www.jahanshiri.ir/fa/en/persian-variants`.

'Maryam watered the garden.' [per] (Godard & Samvelian, 2021, p. 472)

c. Omid dast be golhā zanad.

Omid	dast	be	gol-hā	zan-ad
Omid	hand	to	flower-PL	hit-3SG

'Omid touches the flowers.' [per] (Godard & Samvelian, 2021, p. 470)

The testsuite that I created for Persian contains eight sentences: six positive and two negative. The two negative examples contain the incorrect order of light verb and coverb. The first two positive examples are for simple intransitive and transitive sentences. The next is for an intransitive sentence, shown in (66a). The last three are for transitive sentences. One of them is shown in (66b) above. Parsing this sentence results in one parse, the tree and MRS for which can be found in figure 5.6. This sentence consists of a subject *Maryam*, an object *bāqčerā* "garden", a noun coverb *āb* "water", and the light verb *dād* "to give". The subject, object, and coverb are all arguments of the light verb, as is shown in the *_give_v_lv_rel* relation (the ARG1, ARG2, and ARG3).

Another transitive LVC sentence is shown in (66c), which contains the object *golhā* "flowers" intervening between the coverb *dast* "hand" and the light verb *zanad* "to hit". When parsed, this sentence results in one parse, which is shown in the tree and MRS in figure 5.7. In the tree, the PP object *be golhā* "the flowers" combines with the light verb using the *comp-head-2* rule,[17] and the resulting V combines with the coverb using the *comp-head-lvc* rule. As shown in the *_hit_v_rel* relation in the MRS, the coverb is still the ARG2 of the light verb (represented by the *x10*).

[17] As mentioned in chapter 3, *comp-head-2* is used in languages with verb-second or free word order to account for complements that can occur in positions that the other head-complement rules cannot model. More specifically, *head-comp-2* allows a head to pick up the second element on its COMPS list when the first element has not yet been realized.

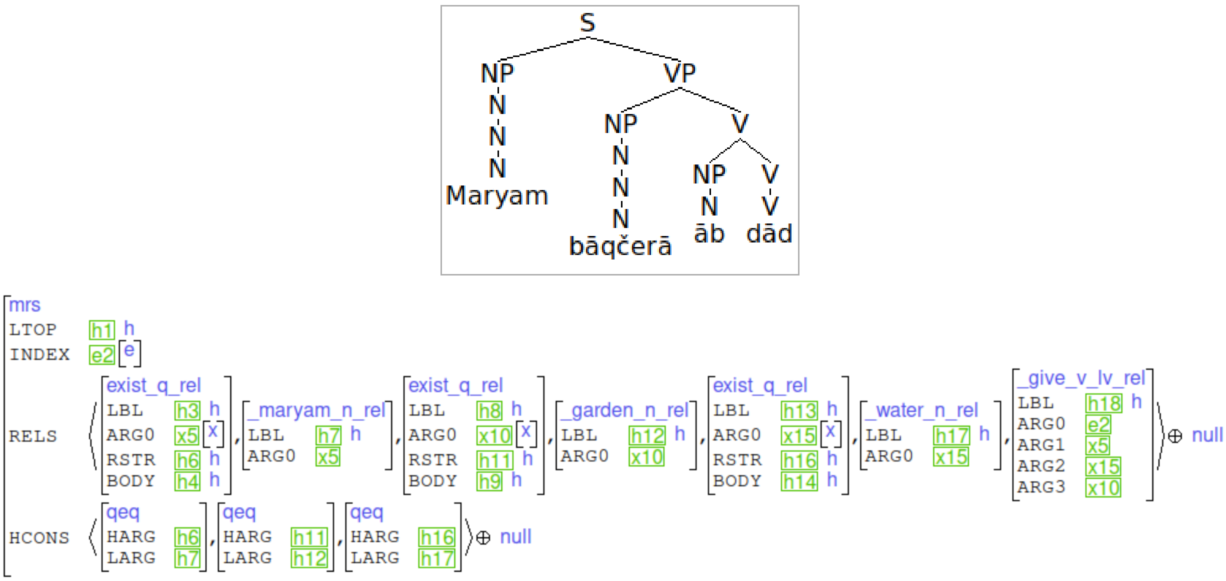


Figure 5.6: Parsing *Maryam bāqčerā āb dād*

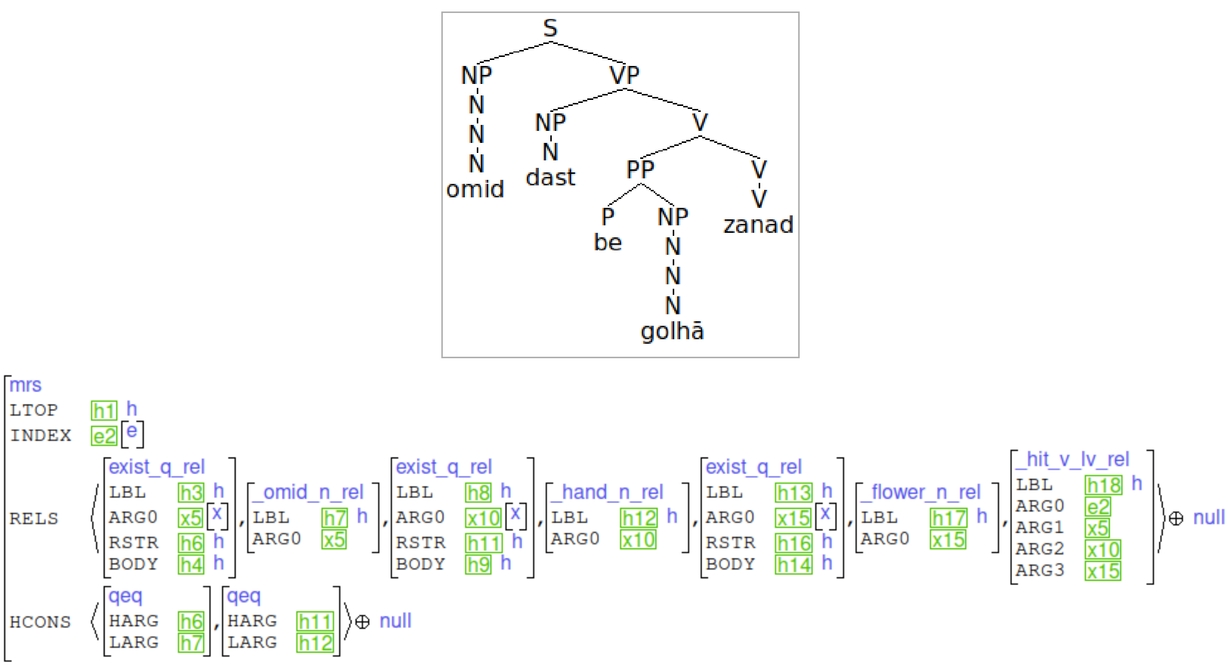


Figure 5.7: Parsing *Omid dast be golhā zanad*

5.1.5 Summary

In this section, I described the four illustrative languages that I used (Bardi, English, Japanese, and Persian) and how the LVC library performed on the testsuites for each language (with regard to coverage, overgeneration, parse ambiguity, and well-formed MRSes). Since these languages were used during implementation, issues with coverage or overgeneration were fixed. However, some issues relating to parse ambiguity and well-formed MRSes were not fixed either because they were outside the scope of this thesis or due to oversights during implementation.

5.2 Pseudo Languages

Pseudo languages are fake languages used to test combinations of choices that might not have come up naturally in the illustrative languages. The goal in doing this is to catch bugs and implementation mistakes that are not predictable with the set of illustrative languages. I came up with three pseudo languages, named Pseudo 1, Pseudo 2, and Pseudo 3.

5.2.1 Pseudo 1 [pso1]

Pseudo 1 has VOS word order and a nominative-accusative case system. Additionally, it allows determiners (which appear after the noun) and adverbs. This pseudo language allows both noun and verb coverbs, both of which can pick up dependents. The coverb must appear before the light verb and must be immediately adjacent to it. Additionally, only intransitive light verbs are possible in this language.

The main purpose of this pseudo language is to test how coverbs interact with picking up dependents, specifically determiners and modifiers. Additionally, it was also used to make sure that intransitive light verbs cannot be used to build transitive constructions. Some examples of the sentences in the testsuite are given in (67).[18]

[18]Full IGT are not provided for the sentences in the pseudo language testsuites as they are not real sentences with meaning. For reference, *coverb1* refers to a noun coverb and *coverb2* refers to a verb coverb. Additionally, *noun1* and *noun2* are basic nouns that optionally allow determiners and *lv1* is an

(67) a. coverb1 det1 lv1 noun1 det1

b. coverb2 adverb1 lv2 noun1

c. *coverb1 lv1 noun1 noun2 det1

d. *det1 coverb1 lv1 noun1 det1

In (67a), a determiner attaches to the noun coverb. When parsed, this determiner is correctly used as the specifier for the coverb. In (67b), an adverb is between the verb coverb and the light verb. When parsed, it results in two trees: one where the adverb is modifying the coverb and one where it is modifying the light verb. Since the adverb was not constrained as to how it can attach, both parses are correct. The MRSes for both sentences are identical and they shouldn't be. The ARG1 of *manner_nonsp_rel* should point towards the ARG0 of the element its modifying. For the parse where the adverb attaches to the coverb, the ARG1 should point to *e6* (the ARG0 of the coverb). For the parse where the adverb attaches to the light verb, the ARG1 should point to *e2* (the ARG0 of the light verb). Currently, the ARG1 of *manner_nonsp_rel* is *e2*, indicating that the adverb is modifying the light verb in both parses. This is due to a bug in the adverb library[19] that is unrelated to the LVC library, and is therefore out of scope for this thesis. The trees and MRS for this sentence can be found in figure 5.8.

In (67c), the transitive sentence does not parse, which is correct. In (67d), the determiner is before the noun it is modifying, which is not allowed in this language. This sentence does not parse and shows that head-specifier rules are being properly applied for specifiers of noun coverbs.

intransitive light verb.

[19]The lexical type for adverbs identifies the ARG1 of *manner_nonsp_rel* with the value for CLAUSE-KEY on the verb it is modifying, which it shouldn't. Since coverbs and light verbs share the value for CLAUSE-KEY, the ARG1 of *manner_nonsp_rel* is the same in both parses.

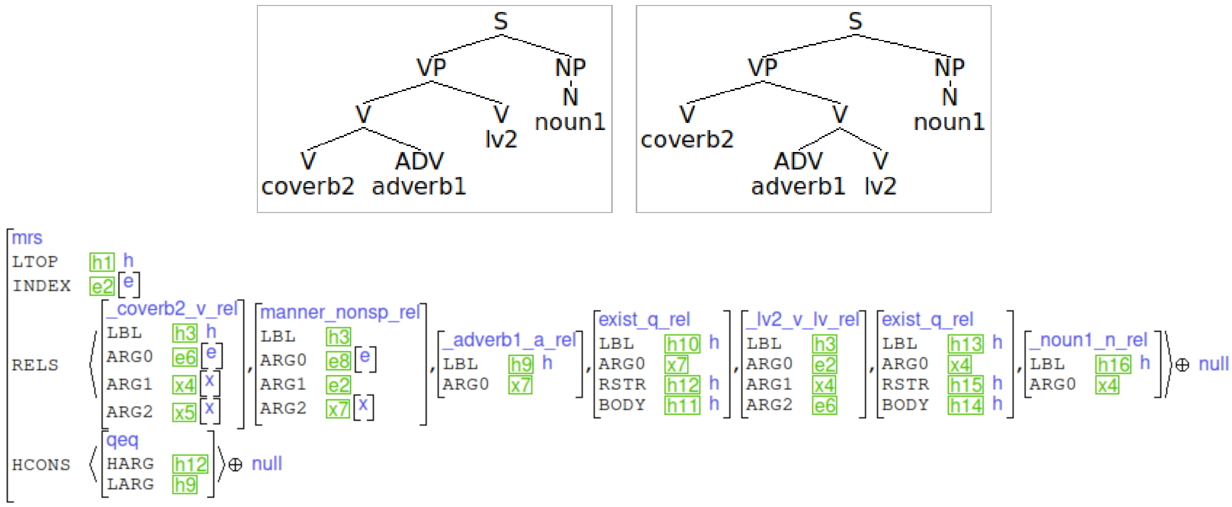


Figure 5.8: Parsing *coverb2 adverb1 lv2 noun1*

5.2.2 Pseudo 2 [pso2]

Pseudo 2 has a word order where the verb occurs in the second position (verb-second) and a nominative-accusative case system. Similar to Pseudo 1, it also allows determiners, which appear after the noun, and adverbs. However, Pseudo 2 only allows verb coverbs (which can pick up dependents). This language allows both orders of coverb and light verb. Additionally, Pseudo 2 allows elements (that are not part of the constituent that the coverb heads) to intervene between the coverb and light verb. Only transitive light verbs are allowed, either bleached or not.

The main purpose of this pseudo language is to test the interactions of allowing both orders of coverbs and light verbs, as this is a choice that did not come up naturally in any of the illustrative languages. It is also used to test having both bleached and unbleached light verbs as well as to make sure that transitive light verbs cannot be used to build intransitive constructions. Lastly, it tests the use of adverbs in a language with close to free word order. Some examples of the sentences in the testsuite are give in (68).[20]

---

[20]For reference, *lv2* refers to a transitive light verb, *lv3* refers to a bleached transitive light verb, and

(68) a. noun1 adverb1 coverb2 lv2 noun2

b. noun1 lv2 noun2 coverb2

c. noun1 det1 coverb3 lv3 noun2 det1

d. *noun1 det1 coverb2 lv2

In (68a), the sentence results in 10 parses. This ambiguity stems from a combination of free word order and non-constrained adverbs. Both *noun1* and *noun2* can act as a subject or object due to this language having free word order. The adverb can attach to just the coverb, the LVC, or the LVC with *noun2*. There isn't anything necessarily ungrammatical about this ambiguity, but the grammar for a real language will most likely want to further constrain adverbs and other lexical types. There is some ambiguity that stems from free word order that is possibly not desired. These parses allow the light verb to combine with *noun2* before combining with the coverb. This is happening because this language allows items to intervene between the coverb and the light verb, which results in less constraints on the light verb. Further work would have to be done to constrain light verbs enough to eliminate these parses (if unwanted) but still allow elements to intervene.

In (68b), the sentence allows *noun2* (as either an object or subject) to intervene between the coverb and the light verb. When parsed, this sentence results in two parses (one with *noun2* as the object and the other with it as the subject), which is expected. In (68c), the sentence uses a bleached light verb. This sentence results in four parses, which are due to the same ambiguity mentioned previously. Lastly, the negative sentence in (68d) does not parse as there is no object and this language only allows transitive sentences.

5.2.3 Pseudo 3 [pso3]

Pseudo 3 also has verb-second word order and a nominative-accusative case system. Although it does allow determiners like the other two pseudo languages, the determiners must appear before the noun. Additionally, instead of allowing adverbs, Pseudo 3 allows

coverb3 refers to a verb coverb.

adjectives and prepositions. This language uses noun coverbs, which can pick up dependents and must appear after the light verb. Only intransitive light verbs are allowed and elements (that are not part of the constituent that the coverbs heads) are allowed to intervene between the light verb and the coverb.

The main purpose of this pseudo language is to test different kinds of dependents (determiners and modifiers) for noun coverbs. Some examples of the sentences in the testsuite are given in (69).

(69) a. noun1 lv1 prep1 noun2 coverb1

b. noun1 lv1 adj1 coverb1

In (69a), there is a prepositional phrase modifying the coverb. In (69b), there is an adjective modifying the coverb. When parsed, each of these sentences result in a single parse. This shows that the coverb can correctly be modified.

5.2.4 Summary

In this section, I described how the LVC library performed on testsuites for the three pseudo languages that I created for this thesis (with regard to coverage, overgeneration, and parse ambiguity). These languages were used during implementation in order to highlight questionnaire choice combinations that did not come up in the illustrative languages. For Pseudo 1 and Pseudo 3, this was to test how coverbs interact with picking up determiners and different kinds of modifiers. For Pseudo 2, this was to test allowing coverbs to appear before and after the light verb as well as test having both bleached and unbleached light verbs. Similar to the illustrative languages, issues with coverage or overgeneration were fixed but not all issues with parse ambiguity were (as they were outside the scope of this thesis).

5.3 Held-Out Languages

Held-out languages are used to test the performance of an implementation in generalizing to languages that were not considered during development. For these languages, I looked at whether or not sentences parsed correctly and if the MRSes for them matched the intended representations. More specifically, I built grammars for each language on a frozen state of library development, meaning that I did not make any changes to my implementation as I tested the coverage of each grammar.[21] I selected three languages in order to test the coverage of my library: Ch'ol, Daasanach, and Korafe-Yegha. These languages were selected primarily based on language family - i.e. I looked for languages in families that the illustrative languages did not belong to. This increases the likelihood of coming across phenomena that might not have been encountered during the implementation phase in order to get a better assessment on the overall performance of the library.

5.3.1 Ch'ol [ctu]

Ch'ol (Chol) is in the Mayan language family and is spoken in southeastern Mexico (primarily in the states of Chiapas, Tabasco, and Campeche). As of the 2020 Population and Housing census performed by the Mexican National Institute of Statistics, Geography and Computing (INEGI), there are 254,715 Ch'ol speakers in Mexico. The majority of this population is in Chiapas, which has 210,771 Ch'ol speakers (approximately 14.4% of Chiapas' 1,459,648 population).[22] Ch'ol is a split-ergative language with VOS word order. It relies on two sets of person markers that are marked on the verb (Set A for agent-like arguments of transitive verbs; and Set B for patient-like arguments of transitive verbs and sole arguments of intransitive verbs[23]) and are dependent on aspect (mainly distinguishing perfective from imperfective) (Coon, 2017).

Ch'ol has one light verb *cha'l* "to do" which is used in three different contexts (Vázquez Álvarez, 2011). The first is with the so-called "verbal nouns," which can have nominal or

[21] The version of the Grammar Matrix customization system that was used to create the grammars for these languages can be found at `https://github.com/delph-in/matrix/tree/77033002bfada2b94e7499ae905955f815754066`.

[22] This data can be found under the Basic questionnaire → National/State → Population file at `https://en.www.inegi.org.mx/programas/ccpv/2020/#tabular_data`.

[23] These are glossed as A and B, which is the same as the Daasanach glossing. All A and B glosses in this section refer to the Cho'ol Set A and Set B and are described on the Abbreviations page.

verbal readings. With nominal readings, verbal nouns can be used as arguments of predicates. With verbal readings, verbal nouns are noun coverbs that require a light verb in order to form a predicate (70a). For example, the verbal noun *soñ* has a verbal reading of "to dance" (when used with the light verb *cha'l*) and a nominal reading of "radio" (which can be used as a subject or object in a sentence). The second and third contexts are with compound verbs (70b) and antipassive absolutive verbs[24] (70c), respectively, both of which require *cha'l* in order to form grammatical sentences. Ch'ol has both noun (the verbal nouns) and verb (the compound and antipassive absolutive verbs) coverbs. Noun coverbs can pick up dependents while verb coverbs cannot. Light verbs occur before the coverb and the coverb (and the constituent it heads) must be immediately adjacent to the light verb. Ch'ol only allows intransitive light verbs, in this case only *cha'l*, which I am analyzing as having both bleached and unbleached readings. When *cha'l* is being used with the compound verbs or antipassive absolutive verbs (i.e. verb coverbs), it isn't contributing any meaning to the construction and is therefore bleached. When it is being used with the verbal nouns (i.e. noun coverbs), it is contributing some meaning to the construction and is therefore unbleached.[25]

(70) a. Tyi icha'le alas jiñ alob.

tyi	i-cha'l-e	alas	jiñ	alob
PFV	A3-do-DTV	game	DET.DEF	child

'The child played.' [ctu] (Coon, 2017, p. 658)

[24]The antipassive absolutive constructions could also be interpreted as auxiliary constructions. However, they were not described as such in Vázquez Álvarez (2011) and, following that description, I made the choice to model them as LVCs. Their use here contrasts with the definition of LVCs that I presented in chapter 3 but this shows the flexibility of the Grammar Matrix customization system in allowing metamodeling choices.

[25]Whether *cha'l* is contributing meaning in the cases where it combines with noun coverb is debatable. However, since the LVC library does not currently model bleached light verbs with noun coverbs, I chose to treat them as unbleached light verbs. These constructions are of particular interest for evaluating my LVC library because coverbs can also act as regular nouns in the language. I wanted to take advantage of the option in the questionnaire that allows a user to specify a noun as being both a coverb and a regular noun.

b. Tyi kcha'le päk'bu'ul.

tyi	k-cha'l-e	päk'bu'ul
PFV	A1-do-DTV	plant.bean

'I planted beans.' [ctu] (Vázquez Álvarez, 2011, p. 221)

c. Tyi kcha'le chonoñel.

tyi	k-cha'l-e	chon-oñ-el
PFV	A1-do-DTV	sell-ANTIP-NF

'I sold.' [ctu] (Vázquez Álvarez, 2011, p. 109)

The testsuite I created for Ch'ol has fifteen sentences: ten positive and five negative. I included some basic examples that do not use LVCs in order to verify that the basic phenomena were input into the Matrix customization system correctly. The first sentence I want to explore is (70a), which produces four parses. This sentence should only have one parse and highlighted two areas of my implementation that were not constrained enough. The first is the result of a choice I made not to implement the option to include require and forbid morphotactic constraints for light verbs, which is something that is an option for most lexical types that can be defined in the Lexicon subpage and allows the user to specify which morphotactic position classes the lexical type is allowed or not allowed to go through. If this had been implemented, it would have allowed me to forbid on of the morphotactic position classes for the light verb.[26] The second issue involves auxiliaries. In the extraneous parse, the perfective auxiliary *tyi* is allowed to combine with the LVC *icha'le k'ay* "played" using the *head-comp-lvc* rule when it should only be allowed to use the *head-comp* rule. This issue is further discussed with English in section 5.1.2 and how it is fixed is discussed in section 5.4.

[26]In the extraneous parse, the intransitive light verb goes through an optional non-inflecting lexical rule that should only be used on transitive verbs (*b3-obj-lex*). Lexical rules can be indicated as optional or required. If a lexical rule is required, all lexical types (that the lexical rule can apply to) must go through that rule. In this case, all transitive verbs must go through *b3-obj-lex* but intransitive verbs shouldn't. In order to get the right behavior, the require and forbid constraints on lexical types can be used. I can force transitive verbs to go through this rule by requiring it and I can prevent intransitive verbs from going through this rule by forbidding it.

(70c) should parse but doesn't, which highlights an issue with my implementation. Currently, inflecting lexical rules are unable to be used with all coverbs due to a unification failure with the expected type for the coverb. As I mentioned in section 4.2.6, I had to do some additional work in order for coverb types to have the correct names so that they can be used in the lexical type hierarchies. There is a bug in the function that fixes these names where it does not output the correct type names for coverbs that combine with bleached light verbs. Since the lexical rules rely on the lexical type hierarchies being properly formed (with the correct names), coverbs that combine with bleached light verbs are unable to go through lexical rules. I was able to confirm this because if if *chonoñel* "to sell" is entered into the lexicon as a full word, the sentence parses.

Ch'ol allows verbal noun coverbs to be modified by a determiner or adjective but not by a demonstrative (Vázquez Álvarez, 2011, p. 105). I included sentences for each of these three cases in my testsuite (71).

(71) a. Tyi kcha'le li ajñel.

tyi	k-cha'l-e	li	ajñel
PFV	A1-do-DTV	DET.DEF	run

'I ran.' [ctu] (Vázquez Álvarez, 2011, p. 105)

b. Yi kcha'le lekojbä ajñel.

yi	k-cha'l-e	lekoj=bä	ajñel
PFV	A1-do-DTV	strange=.REL	run

'I ran strangely.' [ctu] (Vázquez Álvarez, 2011, p. 105)

c. *Tyi kcha'le ili ajñel.

tyi	k-cha'l-e	ili	ajñel
PFV	A1-do-DTV	DEM	run

'I did this run.' [ctu] (Vázquez Álvarez, 2011, p. 105)

(71a) and (71b) each produce four parses. For each sentence, three of the parses are ambiguous for the same reasons as (70a). (71c) doesn't parse, which is correct. In order to do this, I created an additional feature COVERB[27] which can have values of + or -. When entering coverbs into the lexicon, they were made [COVERB +]. Demonstratives in the lexicon are made [COVERB -], which means that they won't be able to combine with any coverbs.

5.3.2 Daasanach [dsh]

Daasanach is a Afro-Asiatic language in the Cushitic branch and is spoken by the Daasanach people in Ethiopia. The most recent census was conducted in 2007, which reported that 0.1% of Ethiopia's 68,722,265 population spoke Daasanach as their primary language (approximately 68,722 people).[28] Daasanach is an SOV language with a subjective-absolutive case system.[29] It relies on two different verb forms – A-form and B-form – which are distinguished by a vocal endings for verb paradigms and/or a suffix, infix, or stem change.[30] Both of these verb forms are used with light verbs and heavy verbs.

Daasanach only allows noun coverbs, which can pick up dependents, and the constituent they head must appear immediately before the light verb, which can be intransitive or transitive. Daasanach has one light verb, which means "to hit, beat"[31] and has an A-form of *ɗoi* and a B-form of *ɗoyyi*.

---

[27]This was done using the Other Features subpage of the Grammar Matrix questionnaire and not by changing my implementation or by editing the TDL files.

[28]This data can be found under the Data and Resources section at `https://data.humdata.org/dataset/ethiopia-languages`.

[29]This is entered into the Matrix as a nominative-accusative case system, since the subjective is used for S and A while absolutive is used for O. S refers to the subject of intransitive verbs, A refers to the agent of transitive verbs, and O refers to the object of transitive verbs, as per Dixon (1994).

[30]These are glossed as A and B, which is the same as the Ch'ol glossing. All A and B glosses in this section refer to the Daasanach A-form and B-form, which are described on the Abbreviations page.

[31]It is not clear from the resources that I found on Daasanach whether this light verb is bleached or not. I treat it as unbleached but it could be argued that this is a bleached light verb. Another option for the LVCs in this language would be to treat them as similar to the verbal nouns in Japanese, which are entered as verb coverbs that combine with a bleached light verb.

(72) a. Yáa ʔél ɗoi.

yáa ʔél ɗoi
1SG.SBJV cheek hit.PRF.A

'I chewed.' [dsh] (Tosco, 2001, p. 495)

b. Luoc náan kár kí ɗoi.

luoc náan kár kí ɗoi
lion spear fence with hit.PRF.A

'The lion fled with the spear (in its body).' [dsh] (Tosco, 2001, p. 510)

c. Min buoyyu ʔaar ɗoyyi.

min buoyyu ʔaar ɗoyyi
woman cry.PRF.B song hit.PRF.B

'The woman cried and sang this song.' [dsh] (Tosco, 2001, p. 295)

The testsuite that I created for Daasanach has thirteen sentences: nine positive and four negative. The four negative examples contain sentences with incorrect word order (both incorrect order of subject, object, and verb as well as incorrect order of coverb and light verb). As for the positive examples, the first four are for basic intransitive and transitive sentences, the last of which uses the heavy verb version of *ɗoyyi* "to hit" and therefore should not form a light verb construction. This sentence can be found in (73).[32] When parsed, it produces one parse that uses the *noun1-noun-lex* type for *ʰík* (a regular noun type) and the *trans-b-form-drop-verb-lex* type for *ɗoyyi* (a heavy verb type), and they are combined with *comp-head*, not *comp-head-lvc*. This is the expected behavior.

(73) Ye ʰík ɗoyyi.

ye ʰík ɗoyyi
1SG.OBJ hiccough hit.PRF.B

'I have the hiccoughs. (lit. hiccough hit me)' [dsh] (Tosco, 2001, p. 495)

[32] The resources that I had for Daasanach were limited and I was unable to find a good example involving a heavy verb use of *ɗoyyi*. Therefore, I chose this sentence to stand in as a non-LVC use case.

The testsuite also contains three intransitive LVC sentences, one of which is shown in (72a) above. When parsed, it produces one parse. The tree and the MRS are well-formed. There are also two transitive LVC sentences in the testsuite, one of which is shown in (72b) above. This sentence does not parse, which highlights an issue with my implementation. In this sentence, *kí́* is a case-marking adposition, which combines with the coverb and heads the resulting prepositional phrase. Case-marking adpositions are [LVC *lv-none*], which bars them from combining with a light verb via *comp-head-lvc*. This was originally added to prevent coverbs from combining with adpositions in examples where they shouldn't. However, it is now clear that this is not the correct way to implement this. Removing this feature from case-marking adpositions allows this sentence to parse.

Another of the intransitive LVC sentences is shown in (72c) above. This sentence includes coordination of the heavy verb *buoyyu* "to cry" and the LVC *ʔaar ɗoyyi* "to sing". Coordination is not a phenomenon that was tested during the implementation stage so this sentence was included in the Daasanach testsuite to see what would happen if it parsed. Surprisingly, it results in one parse, which has a well-formed tree and an almost well-formed MRS. The issue is with one of the handles in the coordination relation. This relation should have the handle of *_cry_v_rel* as its left handle L-HNDL and the handle of *_hit_v_lv_rel* as its right handle R-HNDL. Although the light verb's index (*e11*) is correctly in the coordination relation, its handle (*h18*) is not. This can be seen in figure 5.9. This also occurs in cases of coordination of two heavy verbs, indicating that this is not an issue with the LVC library and is instead an issue with the coordination library.

5.3.3 Korafe-Yegha [kpr]

Korafe-Yegha (Korafe) is a Nuclear Trans New Guinea language spoken primarily in the Oro province of Papua New Guinea. It is spoken by approximately 3000 people, as per a 1990 census (Farr, 2001). It has verb-final word order with no case-marking system. Korafe relies on a verb suffix paradigm where the suffix indicates tense, person and number agreement on the subject, and a speech-act value. This suffix is also used to distinguish between finite verbs

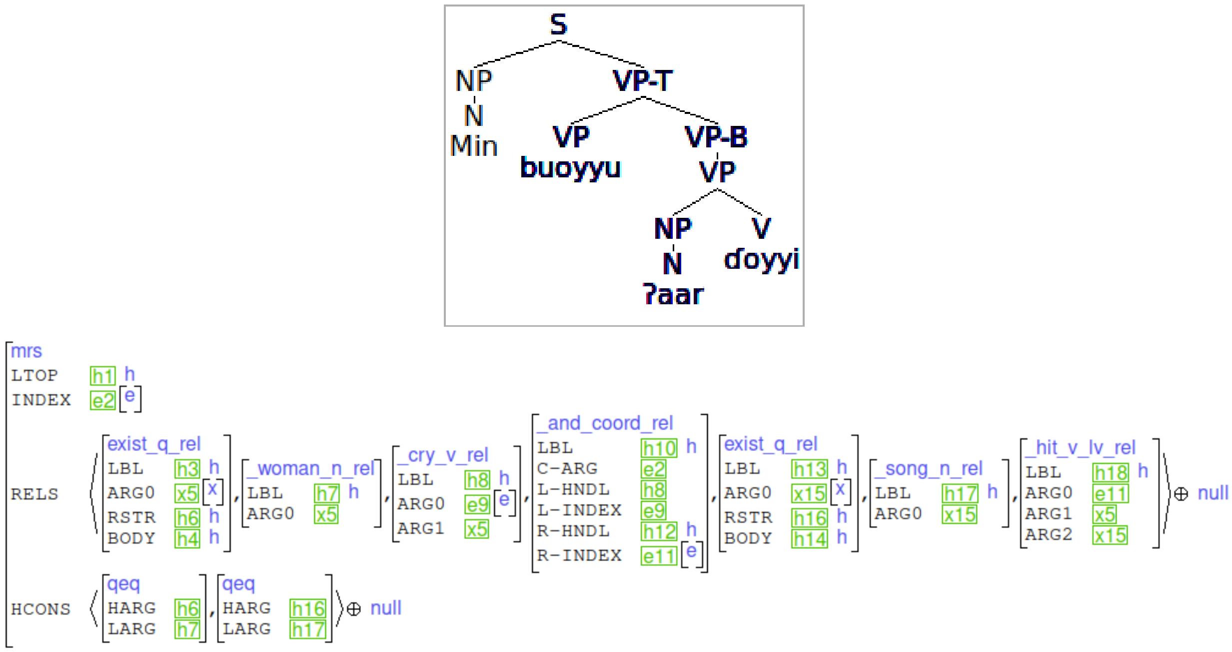


Figure 5.9: Parsing *Min buoyyu ʔaar ɗoyyi*

(FIN) and assertions or questions (AQ). Additionally, the imperfect marker can optionally appear as a prefix or a suffix.

LVCs are common in Korafe, with a range of semantic compositionality (meaning some LVCs are compositional and others are idiomatic). Korafe LVCs use noun coverbs that cannot pick up dependents and appear before the light verb. Both intransitive and transitive light verbs are possible. The coverb and the constituent that it heads does not need to be directly adjacent to the light verb. In order to allow sentences with this property to parse, I had to make the word order free. This led to some overgeneration, which will be discussed below. Some examples of Korafe LVCs can be found in section 5.3.3.

(74) a. Na dubomema ereena.

na dubomema er-e-ena
1SG neck.pain IPFV-do.I-PRS.1SG.FIN

'I am upset over this. (lit. I am doing neck pain)' [kpr] (Farr, 2001, p. 128)

b. Enda andagho emutari.

enda andagho e-mutari
ground clear.brush do-HEST.1PL.AQ

'We cleared the ground.' [kpr] (Farr, 2001, p. 322)

c. Gagara kae nine jighmutasi.

gagara kae ni-ne jigh-mutasi
girl poison 2SG-ACT hold.I-HEST.2SG.AQ

'You practiced sorcery on the girl.' [kpr] (Farr, 2001, p. 66)

d. *Na ni dubo beká ererurena.

na ni dubo beká ere-rur-ena
1SG 2SG neck true IPFV-get.II-PRS.1SG.FIN

'Intended: I love you truly.' [kpr] (Farr, 2001, p. 65)

The testsuite that I created for Korafe has twelve sentences: eight positive and four negative. This includes two simple positive sentences and two simple negative sentences (meaning that they don't use LVCs). As for LVC examples, (74a) contains an intransitive LVC and, when parsed, results in two parses. The light verb used here can be both an intransitive and a transitive light verb, so it has two lexical entries. One of the parses uses the intransitive entry while the other uses the transitive one. In the transitive case, it treats the pronoun *na* as an object and treats the subject as dropped. This parse is not correct and, in order to block it, more work would have to be done in order to restrict the specific light verb and coverb combination to be intransitive. (74b) shows a sentence that does drop the subject (and is correctly parsed).

(74c) is an example of a sentence where an element (that is not part of the coverb's constituent) is in between the coverb and the light verb. In this case, it is the object. This sentence has one parse, which has a well-formed tree and MRS. However, the only way to get this sentence to parse was to allow for overgeneration in other parts of the testsuite. As I mentioned above, I specified (in the questionnaire) that Korafe has free word order (despite Korafe actually having verb-final word order). In order to allow elements to intervene between a light verb and coverb, my implementation relies on rules introduced by the word order library for languages with free or verb-second word order. I wanted to model sentences in Korafe that had intervening elements, therefore I had to specify free word order. In the two basic negative sentences in the testsuite, the verb is in the first or second position. These sentences are not grammatical but still parse (due to the free word order specification). This highlights a limitation with my implementation of this library. Currently, manual editing of the grammar would need to be done in order to block the parses of the ungrammatical strings but still allow (74c).

Lastly, Korafe does not allow coverbs to pick up dependents. (74d) is an example of a sentence with an adjective modifying the coverb and should not parse. However, it does parse. In my implementation, in order to not allow dependents on the coverb, the light verb's lexical type requires [LIGHT +] and [MODIFIED *notmod*] on the coverb. If the coverb has been modified, this should prevent it from combining the the light verb. However, after the coverb combines with the adjective, it goes through the *bare-np* rule. When going through this rule, the values for LIGHT and MODIFIED are not passed up. This means that there is nothing blocking the light verb from combining with the coverb that has picked up dependents. This can be fixed by specifying which features on the head daughter should be passed up in the *bare-np* rule. In this case, the LIGHT feature should not be passed up, since this would lead to other issues, but the MODIFIED should be.

5.3.4 Summary

In this section, I described the three held-out languages (Ch'ol, Daaasanach, and Korafe-Yegha) I chose in order to test the LVC library's ability to generalize to languages that I did not consider during implementation. I looked at coverage, overgeneration, parse ambiguity, and MRSes. There were some issues with coverage for Ch'ol and Daasanach, which resulted from lexical rules not working for all coverbs and coverbs not being able to take case-marking adpositions. The overgeneration in Korafe was a result of a choice I made to model the language as having free word order in order to improve coverage (as this allowed sentences where elements intervened between the light verb and coverb to parse). There was some parse ambiguity and partially incorrect MRSes, which was due to some implementation oversights or, in some cases, issues with other libraries.

5.4 Error Analysis

As mentioned previously, my implementation was tested on a frozen state of development. During this period, I came across some bugs in my implementation, some of which I was able to fix after testing on all languages was complete. Although these improvements are not considered in any of the results, it is still important to document them. A brief summary of each bug and its fix (if applicable) is given below.

I discovered extraneous parses in LVCs with auxiliaries due to missing constraints on the auxiliary (in English and Ch'ol). This was fixed by adding [LVC *lv-none*] to the first element on the *aux-lex* type's COMPS list. Since the *head-comp-lvc* and *comp-head-lvc* rules require the first element on the COMPS list of the head daughter (which would be the auxiliary) to be *lv-all*, this change bars the auxiliary from combining with the light verb with either LVC rule.

Another bug that I discovered involves the HCONS list in the MRS. This bug is not discussed above as it did not affect the metrics being tested (coverage, overgeneration, and ambiguity) for those sentences. The *basic-intrans-lv-lex* lexical type was missing two supertypes that it should have been inheriting from. These two supertypes are *non-local-none-no-hcons* and *basic-icons-lex-item*. The *basic-trans-lv-lex* correctly inherits from both of these types, which connects certain specifier and noun relations using the HARG and LARG features. This bug only affected the MRSes of some sentences, which is why it was originally missed.

A bug that does not affect languages with LVCs, but rather those without, was also discovered during testing. During implementation, I only tested my implementation on grammars with LVCs. When I was working on the held-out languages, I first built grammars that did not have LVCs (in order to make make sure that I correctly specified the basic functionality of the language). However, an error occurred when I tried to generate the grammar. I found that the source of this bug was in a function I had defined and was calling. This function relied on the existence of coverbs in the lexicon. I changed this function to only be called when coverbs do exist in the lexicon and the grammars without LVCs were able to be generated.

For languages where coverbs can take case-marking adpositions (e.g. Japanese and Daasanach), there is a bug in my implementation where coverbs are unable to combine with these adpositions. As mentioned above for Daasanach, this resulted from over-constraining adpositions by making them [LVC *lv-none*]. Removing this constraint fixes the bug.

In Korafe, I discovered a bug where certain properties specified on the coverb are not passed up after the coverb combines with an adjective (or other dependents where the coverb is no longer the head). Once the coverb combines with the adjective, it goes through the *bare-np* rule before combining with the light verb. When going through this rule, the value on the *modified* feature is lost. This can be fixed by changing the *bare-np* rule to pass up the value for this feature.

There were also bugs that I found that I document here for future work. One of these is the example in Bardi where one of the parses extraneously goes through the *dir-obj-no-drop-lex*. As mentioned for Ch'ol, I discovered a bug in my implementation where I do not allow light verbs to require or forbid morphotactic position classes. This results in ambiguity as some incorrect parses cannot be blocked without this functionality. Additionally, sentences in Ch'ol highlighted an issue where coverbs that combine with bleached light verbs cannot go through lexical rules. In the coordinated example from Daasanach, the MRS for the parse was not completely correct, which was due to a bug in the coordination library. For Korafe, a sentence had an extra parse stemming from the transitivity of the light verb and coverb combination not being constrained enough. Lastly, a choice in my implementation for Korafe was to specify it as a language with free word order (in order to get sentences with elements intervening between the light verb and coverb to parse). This resulted in overgeneration in basic intransitive and transitive sentences. In these sentence, the word order is ungrammatical but they were still able to be parsed. This is a limitation of my library that, currently, would require manual editing of the grammar to fix. More generally speaking, these bugs would required further work in order to fix, which I am leaving to future work.

5.5 Overall Results

Table 5.1 contains the results for each language.[33] The metrics that are used for evaluation are the number of positive sentences, coverage, the number of negative sentences, overgeneration, and ambiguity. **Coverage** is the percentage of positive sentences that have a reading (i.e. parse). **Overgeneration** is the percentage of negative sentences that have a reading. The ideal goal is 100% coverage (meaning that all positive sentences have at least one parse) and 0% overgeneration (meaning that no negative sentences have a parse). Parse **ambiguity** is the average number of parses among the positive sentences that parsed. The ideal value would be 1.0, since that would mean one parse per sentence, however there are cases where a language might have multiple correct parses for a sentence.

5.6 Summary

In this chapter, I explained the evaluation process for the LVC library. This process uses sets of illustrative, pseudo, and held-out languages in order to test the performance and coverage of the library. For each language, I used specific sentences in order to highlight certain aspects of my analysis and implementation. I also provided summary statistics on how the grammars (that were output from the Matrix after the addition of the LVC library) performed on the testsuites built for each language. The results of this evaluation show that, for the implementation and pseudo languages, coverage was high, overgeneration was low, and parse ambiguity was fairly low. The metrics for coverage and overgeneration are expected, as these languages were used during development and I made sure the implementation could

[33]These are the results from before any changes were made to fix the bugs and errors described in section 5.4.

Language	Positive	Coverage	Negative	Overgeneration	Ambiguity
Bardi [bcj]	7	100%	3	0%	1.86
English [eng]	7	100%	2	0%	2.86
Japanese [jpn]	6	100%	3	0%	1.0
Persian [per]	6	100%	2	0%	1.0
Pseudo 1 [pso1]	9	100%	5	0%	1.44
Pseudo 2 [pso2]	11	100%	2	0%	3.91
Pseudo 3 [pso3]	10	100%	2	0%	1.80
Ch'ol [ctu]	10	90%	5	0%	3.11
Daasanach [dsh]	9	77.8%	4	0%	1.0
Korafe-Yegha [kpr]	8	100%	4	75%	2.13

Table 5.1: Overall Results

account for the different kinds of sentences in these languages. The values for parse ambiguity are also expected, as there are some kinds of ambiguity that are not incorrect as well as some that were a result of issues outside the scope of my thesis. The results also show that the LVC library, although not perfect, can generalize to languages not used during development. Coverage was 100% for two of these languages and the overgeneration in Korafe-Yegha was due to an implementation choice in order to improve coverage. Parse ambiguity for these languages was on-par with the languages used during development. Overall, the results show that the LVC library can model the properties of LVCs that I set out to model.

Chapter 6

CONCLUSION

In this thesis, I have described the development of the light verb construction (LVC) library as it was added to the LinGO Grammar Matrix Customization System. Chapters 2 and 3 gave background for the tools and frameworks that were used throughout the project and for what is said in the literature about light verb constructions. Chapter 4 focused on my analysis for the library and how I implemented it (including in the Matrix questionnaire as well as in the customization system code). In chapter 5, I described the evaluation process for this project, using illustrative, pseudo, and held-out languages in order to test my implementation of the LVC library, and finally provided summary statistics of the results.

The LVC library relies on my definition of a light verb construction, as a type of complex predicate comprised of a light verb and a coverb, where most (but not necessarily all) of the lexical meaning of the combined complex predicate comes from the coverb. In my analysis, I account for noun and verb coverbs, which are free words that can optionally take dependents. Additionally, light verbs can be unbleached, meaning they are "light" and contribute some information (e.g. an event) to the semantics, or bleached, meaning they are semantically empty. In cases of unbleached light verbs, they do not contribute an event in my analysis. Instead, the coverb contributes an event. Currently, my analysis only accounts for bleached light verbs which occur with verb coverbs. The LVC library allows for the coverb to appear before and/or after the light verb and, depending on the specifications of a language, and optionally allows for intervening elements between the coverb and light verb (currently only for languages with verb-second or free word order).

The additions made to the Matrix have expanded it, making it more useful to its users. Although not every language uses light verb constructions, many do rely on them to form predicates. With the addition of the LVC library, users will be able to model an additional phenomena in their languages and be able to generate grammars with more coverage.

Throughout the process of developing this library, I learned about the importance of creating well-rounded testsuites and testing interactions with other libraries. When building a testsuite for a language, I did my best to include the different types of LVCs that are representative of the phenomena for that language. I also made sure that the options possible via the questionnaire were well-represented across the testsuites. However, in hindsight after testing the held-out languages, I realized that with my focus on specific aspects of each illustrative and pseudo languages, other aspects were not given as much attention. The different libraries that are part of the Grammar Matrix are more interconnected than might be expected. The introduction of a new feature (in the case of this project, the LVC feature) opens the door for much ambiguity if not constrained properly in the features used throughout the Matrix project. Re-using already existing features, as opposed to creating new ones, allows for a more concise analysis, for example like the use of the LIGHT feature in various parts of my analysis. Similarly, I did my best to keep coverb lexical types as close to their corresponding lexical types (ordinary nouns and verbs) as possible, instead of creating coverb lexical types completely from scratch. Doing so allows for more coverage of the different properties of coverbs cross-linguistically. As described in the literature, there are many similarities between coverbs and ordinary nouns/verbs for many languages, which is something I was trying to model in my library. Lastly, I learned about the importance of good documentation in a project that has been working on by numerous grammar engineers over the past twenty years. In the parts of the customization system with good documentation, it was much easier to become familiar with the implementation, which allowed me to work on my implementation faster. Since not all parts were as well-documented, my goal was to include enough documentation for my contributions to make the implementation process easier for future people that work on this project.

When working on a project of this scale, there are always more things to implement that have to be set aside as out of scope. For example, only noun and verb coverbs are currently implemented, however, other coverbs are possible (e.g. adjectives or adverbs). Future work could explore allowing these other coverbs. At the moment, only verb coverbs can combine with bleached light verbs. This is not representative of LVCs cross-linguistically, so future work could look into allowing other kinds of coverbs. I also discovered some bugs while working with the held-out languages. These were described in sections 5.3 and 5.4 and need some additional work in order to fix. This includes the issue pointed out for Korafe-Yegha involving word order, where my implementation currently only accounts for elements intervening between the coverb and the light verb for languages with verb-second or free word order. This does not cover the full range of possibilities as languages with other word orders (like Korafe-Yegha) can also have intervening elements. Additionally, the current semantics for LVCs do not handle the range of idiomatic meanings possible and future work could incorporate the existing approaches to idioms in DELPH-IN grammars (e.g. Bond, Ho & Flickinger (2015)) into the LVC library. As with many of the other libraries, manual editing has to be done in order to account for the nuances of individual languages. However, as a whole, this library is a good starting point for grammar engineers to get coverage for sentences with LVCs.

BIBLIOGRAPHY

Farhat Abdullah, Tafseer Ahmed, Uzma Anjum, et al. 2021. Urdu conjunct predicates (N+V) inventory from Urdu Universal Dependency Corpus. *Corporum: Journal of Corpus Linguistics* 4(1). 15–29.

Anne Abeillé, Daniele Godard & Stefan Müller. 2003. The syntactic flexibility of French degree adverbs. In *Proceedings of the 10th International Conference on Head-Driven Phrase Structure Grammar, Michigan State University*, 26–46. Stanford. `http://cslipublications.stanford.edu/HPSG/4/`.

Josep Alba-Salas. 2002. *Light verb constructions in Romance: A syntactic analysis*. Cornell University dissertation.

Alex Alsina, Joan Bresnan & Peter Sells. 1997. *Complex predicates*. Stanford, CA.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson & Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation* 8(1). 10.1007/s11168-010-9070-1, 23–72. `http://dx.doi.org/10.1007/s11168-010-9070-1`.

Emily M. Bender & Dan Flickinger. 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.

Emily M. Bender, 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.

Olivier Bonami & Pollet Samvelian. 2010. Persian complex predicates: Lexeme formation by itself. In *Septiemes Décembrettes Morphology Conference*, vol. 3. Toulouse, France.

Francis Bond, Jia Qian Ho & Dan Flickinger. 2015. Feeling our way to an analysis of English possessed idioms. In Stefan M¸ller (ed.), *Proceedings of the 22nd International Conference on Head-Driven Phrase Structure Grammar, Nanyang Technological University (NTU)*, 61–74. Stanford, CA: CSLI Publications. https://doi.org/10.21248/hpsg.2015.4.

Claire Bowern. 2010. The typological implications of Bardi complex predicates. *Linguistic Typology* 14(1). 39–70. https://doi.org/doi:10.1515/lity.2010.002. https://doi.org/10.1515/lity.2010.002.

Claire Bowern. 2012. *A grammar of Bardi*. Berlin, Boston: De Gruyter Mouton. https://doi.org/doi:10.1515/9783110278187. https://doi.org/10.1515/9783110278187.

Miriam Butt. 2010. The light verb jungle: Still hacking away. *Complex predicates in cross-linguistic perspective*. 48–78.

Jessica Coon. 2017. Ch'ol. In *The Mayan languages*, 648–684. Routledge.

Ann Copestake. 2002. *Implementing typed feature structure grammars*. Vol. 110. CSLI publications Stanford.

Ann Copestake, Dan Flickinger, Carl Pollard & Ivan A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation* 3. 281–332.

Joshua Crowgey. 2012. *The syntactic exponence of sentential negation: A model for the LinGO Grammar Matrix*. University of Washington MA thesis.

Berthold Crysmann & Woodley Packard. 2012. Towards efficient HPSG generation for German, a non-configurational language. In *Proceedings of COLING 2012*, 695–710.

R. M. W. Dixon. 1994. *Ergativity*. Cambridge: Cambridge University Press.

Scott Drellishak. 2009. *Widespread but not universal: Improving the typological coverage of the Grammar Matrix*. University of Washington dissertation.

Scott Drellishak & Emily M. Bender. 2005. A coordination module for a crosslinguistic grammar resource. In Stefan Müller (ed.), *Proceedings of the 12th International Conference on*

Head-Driven Phrase Structure Grammar, University of Lisbon, 108–128. Stanford: CSLI Publications. `http://cslipublications.stanford.edu/HPSG/6/`.

Cynthia J. Farr. 2001. The interface between syntax and discourse in Korafe, a Papuan language of Papua New Guinea. *Amazonian Languages; Balkan Languages; Bilingual Dictionaries; Central African Republic; Endangered Languages* 4. 50.

Dan Flickinger, Emily M. Bender & Stephan Oepen. 2003. MRS in the LinGO Grammar Matrix: A Practical User's Guide. ms.

Danièle Godard & Pollet Samvelian. 2021. Complex predicates. *Head-Driven Phrase Structure Grammar: The handbook* 9. 419.

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

Jane Grimshaw & Armin Mester. 1988. Light verbs and θ-marking. *Linguistic Inquiry*. 205–232.

Maurice Gross. 1976. Sur quelques groupes nominaux complexes. *Méthodes en grammaire française*. 97–119.

Michael Haeger. 2017. *An evidentiality library for the LinGO Grammar Matrix*. University of Washington MA thesis.

Martin Haspelmath. 1993. *A grammar of Lezgian*. De Gruyter Mouton.

Martin Haspelmath. 2016. The serial verb construction: Comparative concept and cross-linguistic generalizations. *Language and linguistics* 17(3). 291–319.

Kristen Howell, Olga Zamaraeva & Emily M. Bender. 2018. Nominalized clauses in the Grammar Matrix. In Stefan Müller & Frank Richter (eds.), *Proceedings of the 25th International Conference on Head-Driven Phrase Structure Grammar, University of Tokyo*, 68–88. Stanford, CA: CSLI Publications. `http://cslipublications.stanford.edu/HPSG/2018/hpsg2018-hzb.pdf`.

Michio Isoda. 1991. The light verb constructions in Japanese. In M. Dobrin, L. Nichols & R.M. Rodriguez (eds.), *Papers from the 27th Regional Meeting of the Chicago Linguistic Society*, vol. 27, 261–275. Chicago, IL: Chicago Linguistic Society.

Jeong-Ryeol Kim. 1991. *A Lexical-Functional Grammar account of light verbs*. University of Hawai'i at Manoa.

Jeong-Ryeol Kim. 1993. Parsing light verb constructions in Lexical-Functional Grammar. 어학연구.

Jong-Bok Kim, Kyung-Sup Kim & Jaehyung Yang. 2007. Structural ambiguities in the light verb constructions: Lexical relatedness and divergence. *The Linguistic Association of Korea Journal* 15(2). 207–231.

Hans-Ulrich Krieger & Ulrich Schafer. 1994. TDL-A type description language for constraint-based grammars. In *COLING 1994 Volume 2: the 15th International Conference on Computational Linguistics*. https://aclanthology.org/C94-2144.

Yo Matsumoto. 1992. *On the wordhood of complex predicates in Japanese*. Stanford University.

Yo Matsumoto. 1996. A syntactic account of light verb phenomena in Japanese. *Journal of East Asian Linguistics* 5(2). 107–149.

William B. McGregor. 2002. Ergative and accusative patterning in Warrwa. *The nominative & accusative and their counterparts*. 285–317.

Karine Megerdoomian. 2004. A semantic template for light verb constructions. In *Proceedings of the First Workshop on Persian Language and Computers*, 25–26.

Elizabeth K. Nielsen. 2018. *Modeling adnominal possession in the LinGO Grammar Matrix*. University of Washington MA thesis.

Kelly O'Hara. 2008. *A morphotactic infrastructure for a grammar customization system*. University of Washington MA thesis.

Stephan Oepen. 2001. [incr tsdb()] — *Competence and Performance Laboratory. User Manual*. Technical Report. in preparation. Saarbrücken, Germany: Computational Linguistics, Saarland University.

Stephan Oepen & Ulrich Callmeier. 2000. Measure for measure: Parser cross-fertilization. Towards increased component comparability and exchange. In *Proceedings of the 6th International Parsing Technology Workshop*, 183–194. Trento, Italy.

Stephan Oepen & John Carroll. 2000a. Ambiguity packing in constraint-based parsing. Practical results. In *Proceedings of the 1st Conference of the North American Chapter of the ACL*, 162–169. Seattle, WA.

Stephan Oepen & John Carroll. 2000b. Performance profiling for parser engineering. *Natural Language Engineering* 6 (1) (Special Issue on Efficient Processing with HPSG). 81–97.

Stephan Oepen & Dan Flickinger. 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.

Stephan Oepen, Klaus Netter & Judith Klein. 1997. TSNLP — Test Suites for Natural Language Processing. In John Nerbonne (ed.), *Linguistic databases*, 13–36. Stanford, CA: CSLI Publications.

David Osgarby & Claire Bowern. 2023. Complex predication and serialization. *The Oxford guide to Australian languages*.

Barbara Partee. 1984. Compositionality. *Varieties of formal semantics* 3. 281–311.

Carl Pollard & Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Keren Ruditsky. 2024. *Action nominals in the Grammar Matrix*. University of Washington MA thesis.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake & Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Computational Linguistics and Intelligent Text Processing: Third International Conference*, 1–15.

Safiyyah Saleem. 2010. *Argument optionality: A new library for the Grammar Matrix customization system*. University of Washington MA thesis.

Agata Savary, Cherifa Ben Khelil, Carlos Ramisch, Voula Giouli, Verginica Barbu Mititelu, Najet Hadj Mohamed, Cvetana Krstev, Chaya Liebeskind, Hongzhi Xu, Sara Stymne, Tunga Güngör, Thomas Pickard, Bruno Guillaume, Eduard Bejček, Archna Bhatia, Marie Candito, Polona Gantar, Uxoa Iñurrieta, Albert Gatt, Jolanta Kovalevskaite, Timm Lichte, Nikola Ljubešić, Johanna Monti, Carla Parra Escartín, Mehrnoush Shamsfard, Ivelina Stoyanova, Veronika Vincze & Abigail Walsh. 2023. PARSEME corpus release 1.3. In Archna Bhatia, Kilian Evang, Marcos Garcia, Voula Giouli, Lifeng Han & Shiva Taslimipoor (eds.), *Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)*, 24–35. Dubrovnik, Croatia: Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.mwe-1.6. https://aclanthology.org/2023.mwe-1.6 (23 June, 2023).

Melanie Seiss, Miriam Butt & Tracy Holloway King. 2009. On the difference between auxiliaries, serial verbs and light verbs. In *Proceedings of the LFG09 Conference*, 501–519. Stanford, CA: CSLI Publications.

Nicholas Thieberger. 2006. *A grammar of South Efate: an Oceanic language of Vanuatu*. University of Hawaii Press.

Mauro Tosco. 2001. *The Dhaasanac language*. Vol. 17 (Kuschitische Sprachstudien). Köln: Rüdiger Köppe.

Juan Jesús Vázquez Álvarez. 2011. *A grammar of Chol, a Mayan language*. University of Texas dissertation.

Stephen Wilson. 1999. *Coverbs and complex predicates in Wagiman* (Stanford Monographs in Linguistics). Standford, CA: ERIC.

Gernot Windfuhr. 2009. *The Iranian languages*. Routledge.

Olga Zamaraeva, Chris Curtis, Guy Emerson, Antske Fokkens, Michael Wayne Goodman, Kristen Howell, T.J. Trimble & Emily M. Bender. 2022. 20 years of the Grammar Matrix: cross-linguistic hypothesis testing of increasingly complex interactions. *Journal of Language Modelling* 10(1). 49–137.