

# **Light Verb Constructions in the Matrix**

**Tara Wueger**

**University of Washington - Seattle**

**July 2, 2024**

# What is a Light Verb Construction?

- light verb construction (LVC): a type of complex predicate comprised of a light verb and a coverb, where most (but not necessarily all) of the lexical meaning comes from the coverb
- light verb: verb 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 full) lexical meaning to the construction
- coverb: noun, verb, or adjective co-occurring with the light verb

Bardi (N. Australian): *liyan* “heart”

*liyan* + *-(i)nya-* “catch” = “to breathe”

*liyan* + *-ga-* “carry” = “to carry a grudge”

*liyan* + *-ma-* “put” = “to want something”

English

*take* + *a bath* = “to bathe (oneself)”

*give* + *a bath* = “to bathe (someone)”

# General LVC TDL

# Add LVC Type/Features

```
+nvj :+ [ LVC lvc ].  
  
;;; LVCs  
lvc := *top*.  
lv-none := lvc.  
all-lv := lvc.  
joo := all-lv.  
ma := all-lv.  
∅ := all-lv.  
nya := all-lv.  
ar := all-lv.  
gal := all-lv.  
boo := all-lv.  
jiidi := all-lv.  
ga := all-lv.  
ni := all-lv.
```

- LVC
  - type on the coverb
- lv-none
  - not a coverb
- all-lv
  - is a coverb that can go with any light verb
  - other e.g. joo
    - is a coverb that can go with specific coverbs
    - allows for the creation of a light verb hierarchy if useful

# Add Coverb Lexical Types

```
; Verb Coverbs
coverb-verb-lex := intransitive-verb-lex &
[ INFLECTED.IT-ABS-VERB-FLAG na-or-- ].

coverb-verb-lex-all := coverb-verb-lex &
[ SYNSEM.LOCAL.CAT.HEAD.LVC all-lv ].

coverb-verb-lex-joo := coverb-verb-lex &
[ SYNSEM.LOCAL.CAT.HEAD.LVC joo ].

; Noun Coverbs
coverb-noun-lex := basic-noun-lex &
[ ARG-ST < #spr >,
SYNSEM.LOCAL.CAT [ HEAD.MOD < >,
VAL [ COMPS < >,
SPR < #spr & [ LOCAL.CAT.HEAD det,
OPT + ] > ] ] ].

coverb-noun-lex-all := coverb-noun-lex &
[ SYNSEM.LOCAL.CAT.HEAD.LVC all-lv ].
```

- coverb-verb-lex
  - **inheriting from** intransitive-verb-lex to prevent coverbs from taking objects as arguments
- coverb-noun-lex
  - **SPR seeking**
  - **COMPS empty**
  - **MOD empty**

# Add Coverbs to the Lexicon

```
; Verb Coverbs
daab := coverb-verb-lex-all &
[ STEM < "daab" >,
  SYNSEM.LKEYS.KEYREL.PRED "_go.up.to_v_rel" ].

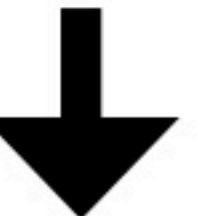
garrja := coverb-verb-lex-joo &
[ STEM < "garrja" >,
  SYNSEM.LKEYS.KEYREL.PRED "_sharp_v_rel" ].

; Noun Coverbs

liyan := coverb-noun-lex-all &
[ STEM < "liyan" >,
  SYNSEM.LKEYS.KEYREL.PRED "_heart_n_rel" ].
```

# Change Noun Lexical Type

```
noun-lex := basic-noun-lex & basic-non-wh-word-lex & non-local-none-lex-item & no-  
hcons-lex-item & non-mod-lex-item &  
[ ARG-ST < #spr >,  
  SYNSEM.LOCAL.CAT.VAL [ COMPS < >,  
    SUBJ < >,  
    SPEC < >,  
    SPR < #spr &  
      [ LOCAL.CAT.HEAD det,  
        OPT + ] > ] ].
```

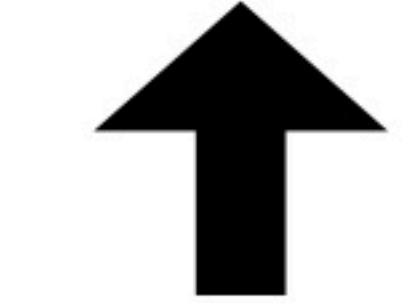


```
noun-lex := basic-noun-lex & basic-non-wh-word-lex & non-local-none-lex-item & no-  
hcons-lex-item & non-mod-lex-item &  
[ ARG-ST < #spr >,  
  SYNSEM.LOCAL.CAT [ HEAD.LVC lv-none,  
    VAL [ COMPS < >,  
      SUBJ < >,  
      SPEC < >,  
      SPR < #spr &  
        [ LOCAL.CAT.HEAD det,  
          OPT + ] > ] ].
```

- makes regular nouns (non-coverbs) LVC lv-none so they can't be used in LVC constructions as coverbs

# Add Light Verb Lexical Types

```
lv-it-lex := tr-min-rule-dtr &
[ SYNSEM.LOCAL.CAT.VAL.SUBJ.FIRST.LOCAL.CAT.HEAD.CASE abs,
  INFLECTED [ TR-ERG-ABS-VERB-FLAG na-or--,
    IT-ABS-VERB-FLAG + ] ].  
  
lv-tr-lex := non-local-none-no-hcons & basic-icons-lex-item & tr-aug-rule-dtr & tr-min-rule-dtr &
[ SYNSEM [ LOCAL [ CAT.VAL [ SUBJ.FIRST.LOCAL.CAT.HEAD.CASE erg,
  COMPS < [] , [ LOCAL [ CAT cat-sat & [ HEAD.CASE not-erg,
    VAL.SPR < > ],
    CONT.HOOK [ INDEX ref-ind & #ind3,
    ICONS-KEY.IARG1 #clause ] ] ] > ],
  CONT.HOOK.CLAUSE-KEY #clause ],
  LKEYS.KEYREL.ARG3 #ind3,
  LIGHT + ],
  INFLECTED [ IT-ABS-VERB-FLAG na-or--,
    TR-AUG-OR-TR-MIN-FLAG -,
    DIR-OBJ-FLAG -,
    TR-ERG-ABS-VERB-FLAG + ] ].
```

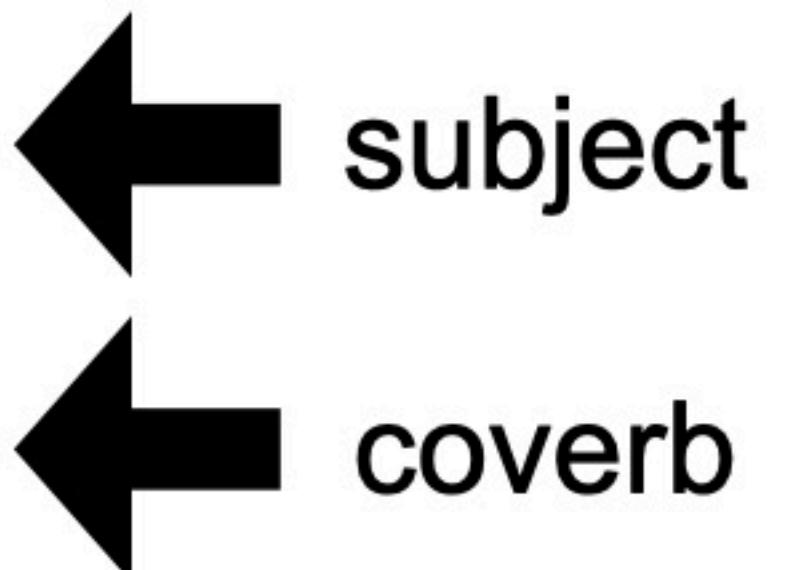


object

# Add Light Verb Lexical Types

```
lv-lex := verb-lex & aspect-rule-dtr & aug-rule-dtr & dir-obj-rule-dtr &  
pernum-rule-dtr & tense-rule-dtr &  
[ SYNSEM [ LOCAL [ CAT [ HEAD.LVC lv-none,  
VAL.COMPS.FIRST #comps ],  
CONT.HOOK [ CLAUSE-KEY #clause,  
LTOP #ltop ] ],  
LKEYS.KEYREL.ARG1 #ind1 ],  
ARG-ST [ FIRST [ LOCAL [ CAT cat-sat,  
CONT.HOOK [ INDEX ref-ind & #ind1,  
ICONS-KEY.IARG1 #clause ] ] ],  
REST.FIRST #comps & [ LOCAL [ CAT cat-sat,  
CONT.HOOK [ ICONS-KEY.IARG1 #clause,  
XARG #ind1,  
LTOP #ltop ] ] ] ],  
INFLECTED [ TENSE-FLAG -,  
PERNUM-FLAG - ] ].
```

- HEAD.LVC lv-none
- subject cannot be coverb



# Add Light Verb Lexical Types

```
lv-nouny-lex := lv-lex &
[ SYNSEM [ LOCAL.CAT.VAL.COMPS.FIRST #comps,
          LKEYS.KEYREL.ARG2 #ind2 ],
  ARG-ST.REST.FIRST #comps & [ LOCAL [ CAT cat-sat & [ VAL.SPR < > ],
                                     CONT.HOOK. INDEX ref-ind & #ind2 ] ] ].  
  
lv-verby-lex := lv-lex &
[ SYNSEM [ LOCAL.CAT.VAL.COMPS.FIRST #comps,
          LKEYS.KEYREL.ARG2 #ind2 ],
  ARG-ST.REST.FIRST #comps & [ LOCAL.CONT.HOOK. INDEX event & #ind2 ] ].  
  
lv-it-nouny-lex := lv-nouny-lex & lv-it-lex.  
  
lv-it-verby-lex := lv-verby-lex & lv-it-lex.  
  
lv-tr-nouny-lex := lv-nouny-lex & lv-tr-lex.  
  
lv-tr-verby-lex := lv-verby-lex & lv-tr-lex.
```

- **nouny coverbs**
  - don't share labels
  - to be implemented:
    - if quantifier, don't share
    - if no quantifier, share
- **INDEX is instance**
- **verby coverb**
  - share labels
  - **INDEX is event**

# Add Light Verb Lexical Types

## Verby Coverb

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

```
[ TOP: h1
INDEX: e2 [ e SF: PROP-OR-QUES E.TENSE: NON-FUT E.ASPECT: REM E.MOOD: REAL ]
RELS: <
  [ "exist_q_rel"
    LBL: h3
    ARG0: x5 [ x SPECI: BOOL COG-ST: COG-ST PNG.PERNUM: 3MIN ]
    RSTR: h6
    BODY: h4 ]
  [ "_man_n_rel"
    LBL: h7
    ARG0: x5 ]
  [ "exist_q_rel"
    LBL: h8
    ARG0: x10 [ x SPECI: BOOL COG-ST: COG-ST PNG.PERNUM: PERNUM ]
    RSTR: h11
    BODY: h9 ]
  [ "_hill_n_rel"
    LBL: h12
    ARG0: x10 ]
  [ "_go.up.to_v_rel"
    LBL: h13
    ARG0: e14 [ e SF: IFORCE E.TENSE: TENSE E.ASPECT: ASPECT E.MOOD: MOOD ]
    ARG1: x5 ]
  [ "_catch_v_lv_rel"
    LBL: h13
    ARG0: e2
    ARG1: x5
    ARG2: e14
    ARG3: x10 ] >
HCONS: < h6 qeq h7 h11 qeq h12 > ]
```

← coverb

← light verb

## Nouny Coverb

```
aamba-nim nimal-nга liyan      i-n-nya-na
man-ERG    nose-INS heart     3-TR-catch-PST
"The man breathed through his nose."
```

```
[ TOP: h1
INDEX: e2 [ e SF: PROP-OR-QUES E.TENSE: NON-FUT E.ASPECT: REM E.MOOD: REAL ]
RELS: <
  [ "exist_q_rel"
    LBL: h3
    ARG0: x5 [ x SPECI: BOOL COG-ST: COG-ST PNG.PERNUM: 3MIN ]
    RSTR: h6
    BODY: h4 ]
  [ "_man_n_rel"
    LBL: h7
    ARG0: x5 ]
  [ "exist_q_rel"
    LBL: h8
    ARG0: x10 [ x SPECI: BOOL COG-ST: COG-ST PNG.PERNUM: PERNUM ]
    RSTR: h11
    BODY: h9 ]
  [ "_nose_n_rel"
    LBL: h12
    ARG0: x10 ]
  [ "exist_q_rel"
    LBL: h13
    ARG0: x15 [ x SPECI: BOOL COG-ST: COG-ST PNG.PERNUM: PERNUM ]
    RSTR: h16
    BODY: h14 ]
  [ "_heart_n_rel"
    LBL: h17
    ARG0: x15 ]
  [ "_catch_v_lv_rel"
    LBL: h18
    ARG0: e2
    ARG1: x5
    ARG2: x15
    ARG3: x10 ] >
HCONS: < h6 qeq h7 h11 qeq h12 h16 qeq h17 > ]
```

← coverb

← light verb

# Add Light Verbs to the Lexicon

```
joo_2 := lv-tr-verby-lex &
[ STEM < "joo" >,
  SYNSEM [ LKEYS.KEYREL.PRED "_do_v_lv_rel",
            LOCAL.CAT.VAL.COMPS.FIRST.LOCAL [ CAT.HEAD.LVC joo ] ] ].  
  
joo_3 := lv-it-verby-lex &
[ STEM < "joo" >,
  SYNSEM [ LKEYS.KEYREL.PRED "_do_v_lv_rel",
            LOCAL.CAT.VAL.COMPS.FIRST.LOCAL [ CAT.HEAD.LVC joo ] ] ].  
  
nya_2 := lv-tr-verby-lex &
[ STEM < "nya" >,
  SYNSEM [ LKEYS.KEYREL.PRED "_catch_v_lv_rel",
            LOCAL.CAT.VAL.COMPS.FIRST.LOCAL [ CAT.HEAD.LVC nya ] ] ].  
  
nya_3 := lv-tr-nouny-lex &
[ STEM < "nya" >,
  SYNSEM [ LKEYS.KEYREL.PRED "_catch_v_lv_rel",
            LOCAL.CAT.VAL.COMPS.FIRST.LOCAL [ CAT.HEAD.LVC nya ] ] ].
```

# Grammar Rules

# Change Grammar Rules

```
decl-head-subj-phrase :+  
[ HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.LVC lv-none ].
```

```
head-comp-phrase := basic-head-1st-comp-phrase & head-initial-head-nexus &  
[ NON-HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.LVC lv-none ].
```

```
comp-head-phrase := basic-head-1st-comp-phrase & head-final-head-nexus &  
[ NON-HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.LVC lv-none ].
```

```
head-comp-phrase-2 := basic-head-2nd-comp-phrase & head-initial-head-nexus &  
[ NON-HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.LVC lv-none ].
```

```
comp-head-phrase-2 := basic-head-2nd-comp-phrase & head-final-head-nexus &  
[ NON-HEAD-DTR.SYNSEM.LOCAL.CAT.HEAD.LVC lv-none ].
```

- **decl-head-subj-phrase**
  - prevents object from combining with coverb in "nom acc coverb lv" before coverb + lv have combined
- **others**
  - LVC lv-none to prevent these rules from combining light verbs with coverbs

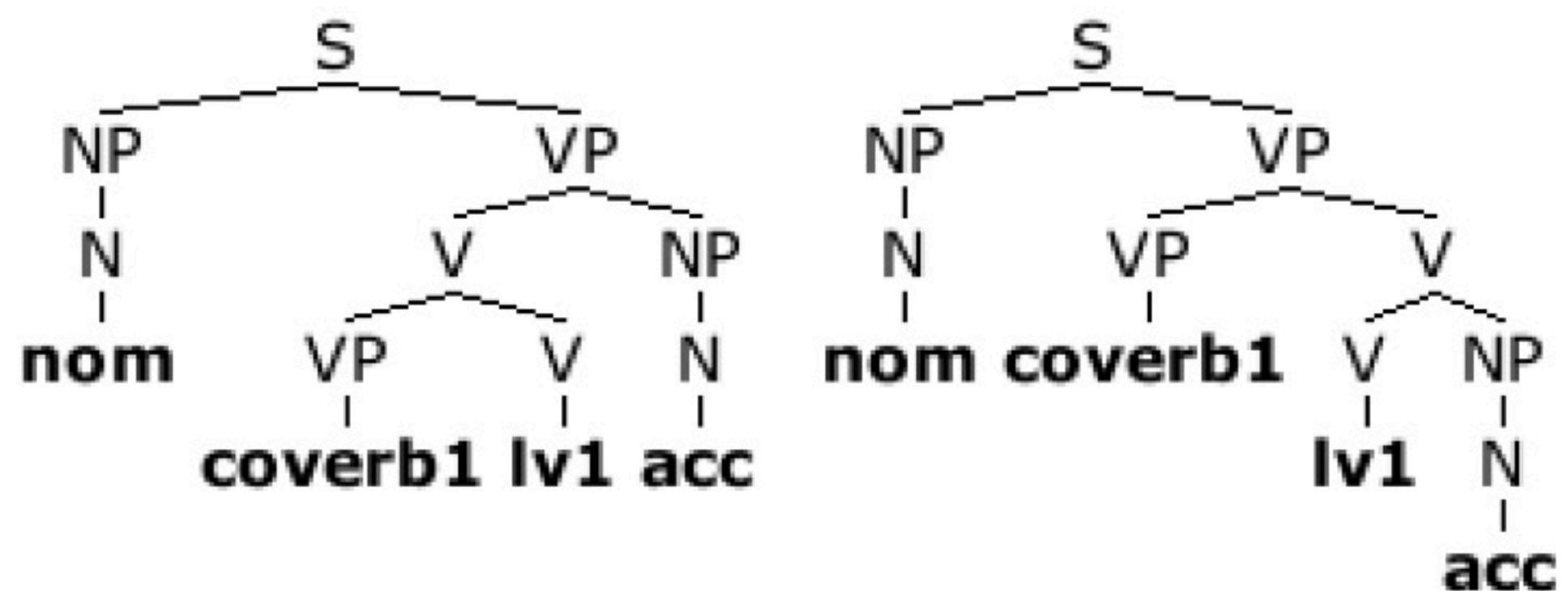
# Add Grammar Rules

```
head-final-head-nexus-lv := head-final &  
[ SYNSEM.ATTACH lmod ].
```

```
head-comp-phrase-lvc := basic-head-1st-comp-phrase & head-initial-head-nexus &  
[ HEAD-DTR.SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.LOCAL.CAT.HEAD.LVC all-lv ].
```

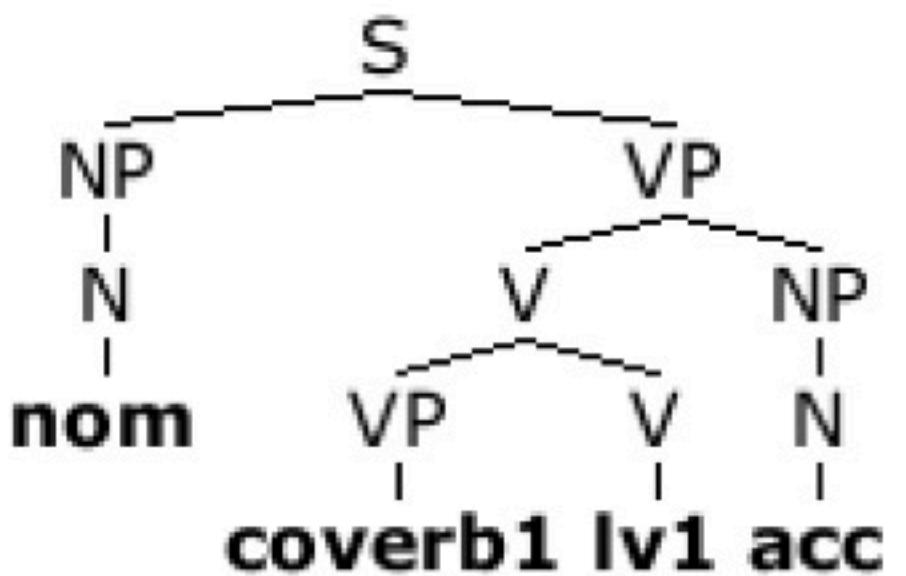
```
comp-head-phrase-lvc := basic-head-1st-comp-phrase & head-final-head-nexus-lv &  
[ HEAD-DTR.SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.LOCAL.CAT.HEAD.LVC all-lv ].
```

- allows coverb to combine with light verb before object in “nom coverb lv acc” constructions
- LVC all-lv to combine light verb and coverb
  - light verb seeks coverb with LVC all-lv

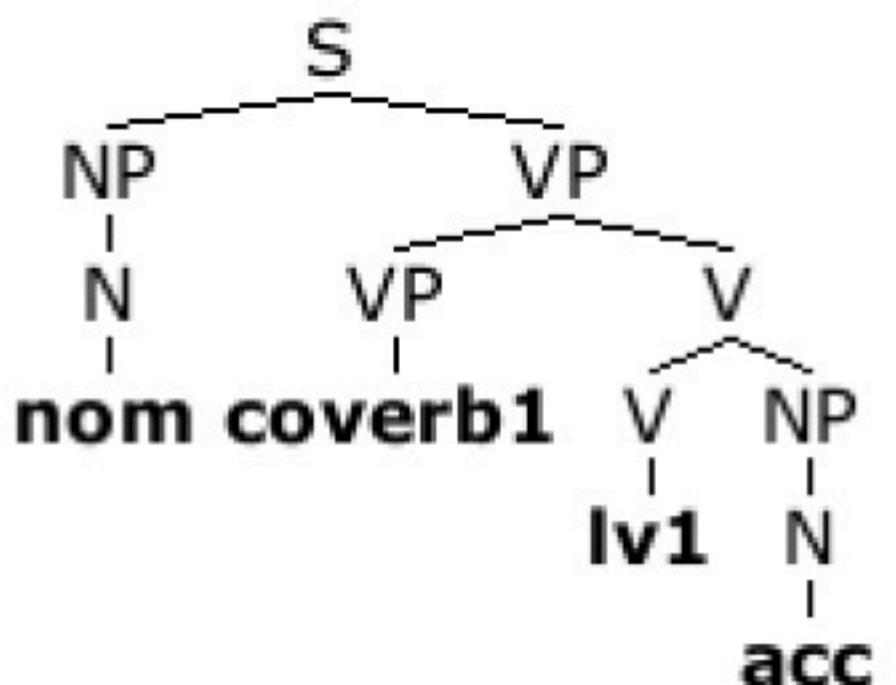


# Add Grammar Rules

```
comp-head-phrase-lvc := basic-head-1st-comp-phrase & head-final-head-nexus-lv &  
[ HEAD-DTR.SYNSEM [ LOCAL.CAT.VAL.COMPS.FIRST.LOCAL.CAT.HEAD.LVC all-lv,  
LIGHT + ] ].
```



```
comp-head-phrase-lvc := basic-head-1st-comp-phrase & head-final-head-nexus-lv &  
[ HEAD-DTR.SYNSEM.LOCAL.CAT.VAL.COMPS < [ LOCAL.CAT.HEAD.LVC all-lv ] > ].
```



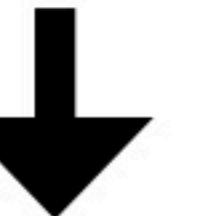
1. Can the coverb pick up dependents on its own (e.g. det, modifier, complement)?
  - if no, LIGHT + e.g. Bardi
2. Does the light verb have to combine with the coverb-headed constituent first?
  - if yes, length of COMPS list = 1 e.g. Persian

# Additional Phenomena

# Change Adverb Negation

```
neg-adv-lex := basic-scopal-adverb-lex &
[ SYNSEM.LOCAL.CAT [ VAL [ SPR < >,
                           COMPS < >,
                           SUBJ < >,
                           SPEC < > ],
                     POSTHEAD -,
                     HEAD.MOD < [ LOCAL.CAT.HEAD verb,
                                   LIGHT + ] > ] ].
```

From negation library



```
neg-adv-lex := basic-scopal-adverb-lex &
[ SYNSEM.LOCAL.CAT [ VAL [ SPR < >,
                           COMPS < >,
                           SUBJ < >,
                           SPEC < > ],
                     POSTHEAD -,
                     HEAD.MOD < [ LOCAL [ CAT.HEAD verb & [ LVC lv-none ],
                                   CONT.HOOK.INDEX.E.MOOD irr ] ] > ] ].
```

- remove **LIGHT +**
- **MOD is LVC lv-**  
**none to prevent**  
**neg adverb from**  
**attaching to**  
**coverb (Bardi-**  
**specific)**
- **verb must be**  
**have irrealis**  
**mood (Bardi-**  
**specific)**

# Add LVC Coordination

```
vp-coord-phrase :+ [ SYNSEM.LOCAL.CAT.HEAD.LVC #lvc,  
LCOORD-DTR.SYNSEM.LOCAL.CAT.HEAD.LVC #lvc,  
RCOORD-DTR.SYNSEM.LOCAL.CAT.HEAD.LVC #lvc ].
```

- prevents coverb + LVC parses
- forces LVC feature to match on left and right daughters

```
vp-bottom-coord-phrase :+ [ SYNSEM.LOCAL.CAT.HEAD.LVC #lvc,  
NONCONJ-DTR.SYNSEM.LOCAL.CAT.HEAD.LVC #lvc ].
```

- \*\*Not used in Bardi**
- prevents LVC + coverb parses

# Matrix Questionnaire

# Matrix Questionnaire - LVCs

Light Verb Constructions — Mozilla Firefox

Light Verb Constructions

127.0.0.1:9000/matrix.cgi?subpage=lvc

## Light Verb Constructions [documentation]

If your language uses light verb constructions (LVCs), mark the appropriate options below.

What possible coverbs are allowed in your language?  
(selecting one or more of these options will allow you to add coverbs of that type to the lexicon)

Noun Coverbs  
 Verb Coverbs

The word order within an LVC is:

The coverb is before the light verb  
 The coverb is after the light verb  
 The coverb can be before or after the light verb

Can words go in between the coverb and the light verb?

yes  
 no

Main page

- \*Gen Info
- \*Word Order
- Number
- \*Person
- Gender
- \*Case
- Poss
- Dir-inv
- TAM
- Evidentials
- Features
- Neg
- Coord
- Y/N Os
- Wh-Qs
- Info Str
- Arg Opt
- Nmz
- Light Verb
- Constructions
- Embed Claus
- Clausal Mod
- ?Lexicon
- Morph
- Toolbox Import
- Test S
- TbG Options

Choices file  
(right-click to download)  
Save & stay  
Clear current  
subpage  
Create grammar: #  
tgz, zip

Light Verb Constructions — Mozilla Firefox

Light Verb Constructions

127.0.0.1:9000/matrix.cgi?subpage=lvc

## Light Verb Constructions [documentation]

If your language uses light verb constructions (LVCs), mark the appropriate options below.

What possible coverbs are allowed in your language?  
(selecting one or more of these options will allow you to add coverbs of that type to the lexicon)

Noun Coverbs  
 Verb Coverbs

Do noun coverbs take noun-normal dependents?

yes  
 no

Do verb coverbs take noun-normal dependents?

yes  
 no

The word order within an LVC is:

The coverb is before the light verb  
 The coverb is after the light verb  
 The coverb can be before or after the light verb

Can words go in between the coverb and the light verb?

yes  
 no

Main page

- \*Gen Info
- \*Word Order
- Number
- \*Person
- Gender
- \*Case
- Poss
- Dir-inv
- TAM
- Evidentials
- Features
- Neg
- Coord
- Y/N Os
- Wh-Qs
- Info Str
- Arg Opt
- Nmz
- Light Verb
- Constructions
- Embed Claus
- Clausal Mod
- ?Lexicon
- Morph
- Toolbox Import
- Test S
- TbG Options

Choices file  
(right-click to download)  
Save & stay  
Clear current  
subpage  
Create grammar: #  
tgz, zip

# Matrix Questionnaire - Lexicon

Lexicon — Mozilla Firefox

Lexicon 127.0.0.1:9000/matrix.cgi

Define a copula to introduce complements.  
[visualize copula hierarchy \(experimental\)](#)

Add a Copula Type

**Determiners**

Add a Determiner

**Adverbs**

Add an Adverb

Note that the adverbs are not yet fully supported and will be underspecified as to where they attach.

**Adpositions**

Add an Adposition

Add a Case-marking and/or Information Structural Adposition

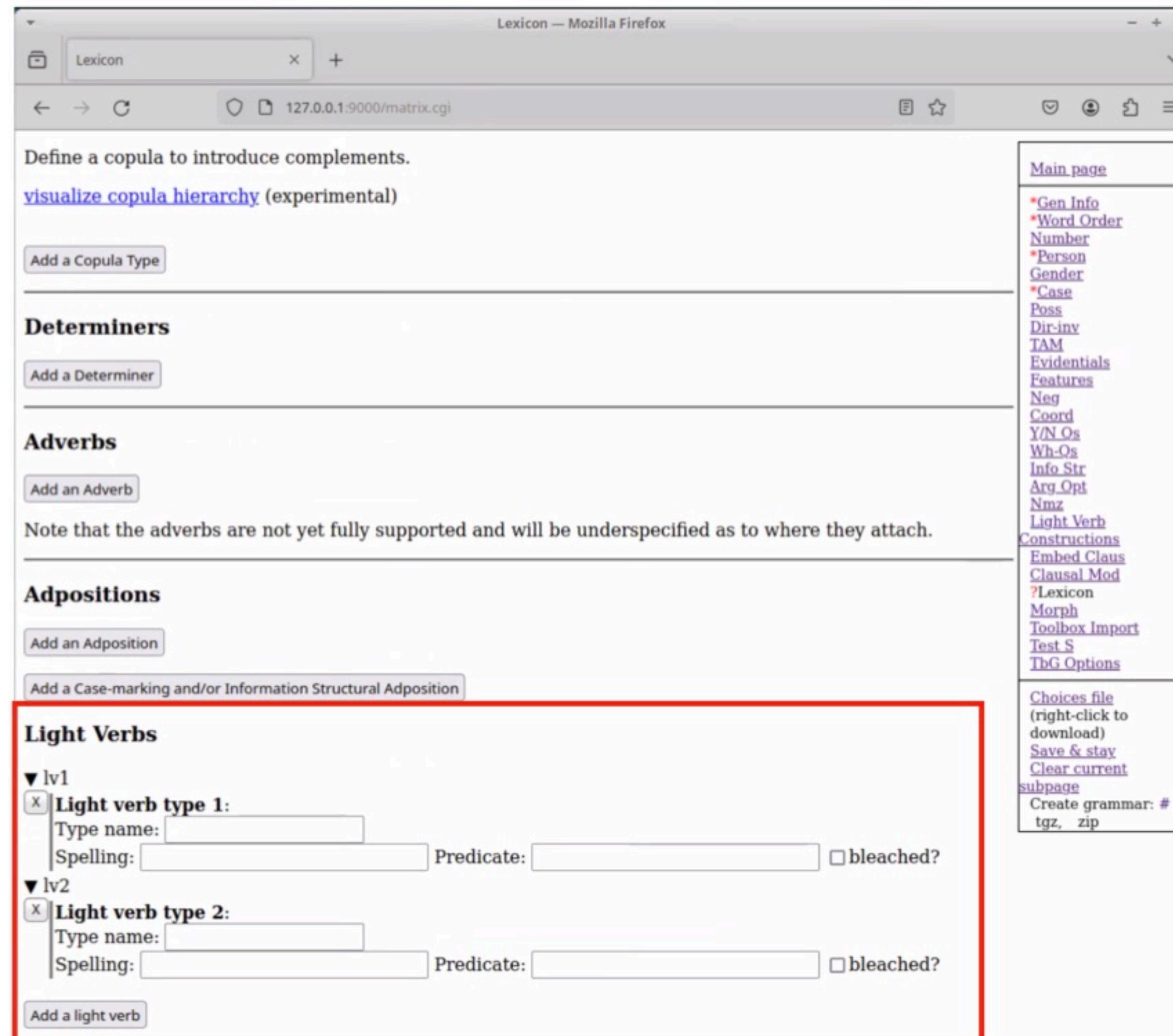
**Light Verbs**

▼ lv1  
X Light verb type 1:  
Type name:   
Spelling:  Predicate:   bleached?

▼ lv2  
X Light verb type 2:  
Type name:   
Spelling:  Predicate:   bleached?

Add a light verb

Main page  
\*Gen Info  
\*Word Order  
Number  
\*Person  
Gender  
\*Case  
Poss  
Dir-inv  
TAM  
Evidentials  
Features  
Neg  
Coord  
Y/N Os  
Wh-Qs  
Info Str  
Arg Opt  
Nmz  
Light Verb  
Constructions  
Embed Claus  
Clausal Mod  
?Lexicon  
Morph  
Toolbox Import  
Test S  
TbG Options  
Choices file  
(right-click to download)  
Save & stay  
Clear current  
subpage  
Create grammar: #  
tgz, zip



# Matrix Questionnaire - Lexicon

Lexicon — Mozilla Firefox

127.0.0.1:9000/matrix.cgi

**noun1**

**Noun type 1:**

Type name:

Supertypes:

This is a personal pronoun type

This is a question pronoun (like who/what)

Features:

Add a Feature

For nouns of this type, a determiner is:  obligatory  optional  impossible

Stems:

Spelling:  Predicate:

Add a Stem

Morphotactic Constraints:

Add a Require constraint

Add a Forbid constraint

Light Verb Constructors:

This can be a coverb  
 This is only a coverb

Light verbs that take this coverb class as an argument:

**noun1\_lv1**

Light verb(s):  LVC valence:

Choose a light verb  lv1  lv2

Add a Noun Type

Main page

- \*Gen Info
- \*Word Order
- Number
- \*Person
- Gender
- \*Case
- Poss
- Dir-inv
- TAM
- Evidentials
- Features
- Neg
- Coord
- Y/N Qs
- Wh-Qs
- Info Str
- Arg Opt
- Nmz
- Light Verb
- Constructions
- Embed Claus
- Clausal Mod
- ?Lexicon
- Morph
- Toolbox Import
- Test S
- TbG Options

Choices file (right-click to download)  
[Save & stay](#) [Clear current subpage](#) [Create grammar: # tgz, zip](#)

Lexicon — Mozilla Firefox

127.0.0.1:9000/matrix.cgi

Type name:

Supertypes:

Features:

Add a Feature

Argument structure: ?

If this verb class includes bipartite stems, select the position class for the affix portion of the stems:

Stems:

Spelling:  Predicate:

Add a simple Stem

Add a bipartite Stem

Morphotactic Constraints:

Add a Require constraint

Add a Forbid constraint

Light Verb Constructors:

This can be a coverb  
 This is only a coverb

Light verbs that take this coverb class as an argument:

**verb1\_lv1**

Light verb(s):  LVC valence:

**verb1\_lv2**

Light verb(s):  LVC valence:

**verb1\_lv3**

Light verb(s):  LVC valence:

Choose a light verb

Main page

- \*Gen Info
- \*Word Order
- Number
- \*Person
- Gender
- \*Case
- Poss
- Dir-inv
- TAM
- Evidentials
- Features
- Neg
- Coord
- Y/N Qs
- Wh-Qs
- Info Str
- Arg Opt
- Nmz
- Light Verb
- Constructions
- Embed Claus
- Clausal Mod
- ?Lexicon
- Morph
- Toolbox Import
- Test S
- TbG Options

Choices file (right-click to download)  
[Save & stay](#) [Clear current subpage](#) [Create grammar: # tgz, zip](#)

# Current / Next Steps

- Finish implementing grammars for Persian, Urdu, and Japanese
- Finish Matrix questionnaire/customization backend
- Evaluation

# Discussion

# References

- Bender, E. M., Flickinger, D., & Oepen, S. (2002). The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In COLING-02: Grammar Engineering and Evaluation.
- Bender, E. M., Drellishak, S., Fokkens, A., Poulson, L., & Saleem, S. (2010). Grammar customization. *Research on Language and Computation*, 8, 23-72.
- Bowern, C. (2012). A grammar of Bardi. de Gruyter.
- Butt, M. (2010). The light verb jungle: Still hacking away. Complex predicates in cross-linguistic perspective, 48-78.