Package ‘MoLE’

October 24, 2017

Type Package

Title Modeling Language Evolution

Version 1.0.1

Date 2017-10-23

Author Sander Lestrade

Maintainer Sander Lestrade <samlestrade@protonmail.com>

Description Model for simulating language evolution in terms of cultural evolution (Smith & Kirby (2008) <DOI:10.1098/rstb.2008.0145>; Deacon 1997). The focus is on the emergence of argument-marking systems (Dowty (1991) <DOI:10.1353/lan.1991.0021>, Van Valin 1999, Dryer 2002, Lestrade 2015a), i.e. noun marking (Aristar (1997) <DOI:10.1075/sl.21.2.04ari>, Lestrade (2010) <DOI:10.7282/T3ZG6R4S>), person indexing (Ariel 1999, Dahl (2000) <DOI:10.1075/fol.7.1.03dah>, Bhat 2004), and word order (Dryer 2013), but extensions are foreseen. Agents start out with a protolanguage (a language without grammar; Bickerton (1981) <DOI:10.17169/langsci.b91.109>, Jackendoff 2002, Arbib (2015) <DOI:10.1002/9781118346136.ch27>) and interact through language games (Steels 1997). Over time, grammatical constructions emerge that may or may not become obligatory (for which the tolerance principle is assumed; Yang 2016). Throughout the simulation, uniformitarianism of principles is assumed (Hopper (1987) <DOI:10.3765/bls.v13i0.1834>, Givon (1995) <DOI:10.1075/z.74>, Croft (2000), Saffran (2001) <DOI:10.1111/1467-8721.01243>, Heine & Kuteva 2007), in which maximal psychological validity is aimed at (Grice (1975) <DOI:10.1057/9780230005853_5>, Levelt 1989, Gaerdenfors 2000) and language representation is usage based (Tomasello 2003, Bybee 2010). In Lestrade (2015b) <DOI:10.15496/publikation-8640>, Lestrade (2015c) <DOI:10.1075/avt.32.08les>, and Lestrade (2016) <DOI:10.17617/2.2248195>), which reported on the results of preliminary versions, this package was announced as WDWTW (for who does what to whom), but for reasons of pronunciation and generalization the title was changed.

Depends R (>= 3.0.0)

LazyData TRUE

License GPL-2

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-10-24 07:21:35 UTC

R topics documented:

MoLE-package . . . 3
ACTOR . . . 7
AGENTFIRST . . . 8
ALLNAS . . . 9
ANALYZE . . . 10
CANDIDATESCORE . . . 11
CHECKSUCCESS . . . 12
DECOMPOSE . . . 13
DIE . . . 14
EROSION . . . 15
FIRSTINFIRSTOUT . . . 16
FIRSTSPEAKER . . . 17
FMATCH . . . 18
FORMS . . . 19
FOUND . . . 20
FREQUPDATE . . . 20
FUSE . . . 21
GENERALIZE . . . 23
GROUP . . . 24
INTERPRET . . . 25
INTERPRET.INT . . . 26
MAX . . . 27
NOUNDESEMANTICIZATION . . . 28
NOUNMORPHOLOGY . . . 29
NOUNS . . . 30
PERSONUPDATE . . . 31
PREPARE . . . 32
PROCREATE . . . 33
PRODUCE . . . 34
PROPOSITION . . . 35
PROTOINTERPRETATION . . . 36
REDUCE . . . 37
REFCHECK . . . 38
RESCALE . . . 39
RUN . . . 40
SELECTACTOR . . . 40
SEMUPDATE . . . 41
SITUATION . . . 42
SUCCESS . . . 43
SUMMARY . . . 44
TALK . . . 45
TOPICCOPY . . . 46

TOPICFIRST 47
TURN 48
TYPEMATCH 49
VERBFINAL 50
VERBMORPHOLOGY 51
VERBS 52
VMATCH 52
WORDORDER 53
world 55

Index **60**

`MoLE-package` *Modeling Language Evolution*

Description

Model for simulating language evolution in terms of cultural evolution (Smith & Kirby (2008) <DOI:10.1098/rstb.2008.0145>; Deacon 1997). The focus is on the emergence of argument-marking systems (Dowty (1991) <DOI:10.1353/lan.1991.0021>, Van Valin 1999, Dryer 2002, Lestrade 2015a), i.e. noun marking (Aristar (1997) <DOI:10.1075/sl.21.2.04ari>, Lestrade (2010) <DOI:10.7282/T3ZG6R4S>), person indexing (Ariel 1999, Dahl (2000) <DOI:10.1075/fol.7.1.03dah>, Bhat 2004), and word order (Dryer 2013), but extensions are foreseen. Agents start out with a protolanguage (a language without grammar; Bickerton (1981) <DOI:10.17169/langsci.b91.109>, Jackendoff 2002, Arbib (2015) <DOI:10.1002/9781118346136.ch27>) and interact through language games (Steels 1997). Over time, grammatical constructions emerge that may or may not become obligatory (for which the tolerance principle is assumed; Yang 2016). Throughout the simulation, uniformitarianism of principles is assumed (Hopper (1987) <DOI:10.3765/bls.v13i0.1834>, Givon (1995) <DOI:10.1075/z.74>, Croft (2000), Saffran (2001) <DOI:10.1111/1467-8721.01243>, Heine & Kuteva 2007), in which maximal psychological validity is aimed at (Grice (1975) <DOI:10.1057/9780230005853_5>, Levelt 1989, Gaerdenfors 2000) and language representation is usage based (Tomasello 2003, Bybee 2010). In Lestrade (2015b) <DOI:10.15496/publikation-8640>, Lestrade (2015c) <DOI:10.1075/avt.32.08les>, and Lestrade (2016) <DOI:10.17617/2.2248195>), which reported on the results of preliminary versions, this package was announced as WDWTW (for who does what to whom), but for reasons of pronunciation and generalization the title was changed.

Details

The DESCRIPTION file:

Package:	MoLE
Type:	Package
Title:	Modeling Language Evolution
Version:	1.0.1
Date:	2017-10-23
Author:	Sander Lestrade
Maintainer:	Sander Lestrade <samlestrade@protonmail.com>
Description:	Model for simulating language evolution in terms of cultural evolution (Smith & Kirby (2008) <DOI:10.109

Depends:	R (>= 3.0.0)
LazyData:	TRUE
License:	GPL-2
RoxygenNote:	6.0.1

Index of help topics:

```
ACTOR                   Determine actor role
AGENTFIRST              Actor argument first
ALLNAS                  NA vector identification
ANALYZE                 Determine sentence constituents
CANDIDATESCORE          Score candidate expressions
CHECKSUCCESS            Determine expected communicative success
DECOMPOSE               Decompose words into morphemes
DIE                     Kill agents
EROSION                 Word erosion
FIRSTINFIRSTOUT         Order constituents by activation
FIRSTSPEAKER            Create founding agent
FMATCH                  Compare forms
FORMS                   Generate forms
FOUND                   Found population
FREQUPDATE              Update usage numbers
FUSE                    Fuse words
GENERALIZE              Apply linguistic generalizations
GROUP                   Group words into constituents
INTERPRET               Interpret utterance
INTERPRET.INT           Develop an interpretation
MAX                     Find maximum value
MoLE-package            Modeling Language Evolution
NOUNDESEMANTICIZATION   Bleach word meaning
NOUNMORPHOLOGY          Interpret nominal morphology
NOUNS                   Generate nominal lexicon
PERSONUPDATE            Adjust person value
PREPARE                 Prepare a proposition for production
PROCREATE               Generate new generation of agents
PRODUCE                 Produce utterance
PROPOSITION             Develop initial proposition
PROTOINTERPRETATION     Develop interpretation
REDUCE                  Reduce length of expressions
REFCHECK                Check referential capacity
RESCALE                 Rescale vector values
RUN                     Run simulation
SELECTACTOR             Find actor expression
SEMUPDATE               Update lexicon
SITUATION               Create situational context
SUCCESS                 Determine communicative success
SUMMARY                 Summarize simulation results
```

```
TALK            Let agents talk
TOPICCOPY       Make anaphoric copy of topic
TOPICFIRST      Put topic in first position
TURN            Organize communicative turn
TYPEMATCH       Determine role qualification
VERBFINAL       Put verb final
VERBMORPHOLOGY  Interpret verbal morphology
VERBS           Generate verbal lexicon
VMATCH          Compare vectors
WORDORDER       Use word order for interpretation
world           Model parameters
```

Set the model parameters in `world`. Found a new population (`FOUND`). Run a simulation (`RUN`).

For language to change (and argument-marking grammar to develop), the simulation has to run for several hours.

Author(s)

Sander Lestrade

Maintainer: Sander Lestrade <samlestrade@protonmail.com>

References

Ariel, M. (1999), The development of person agreement markers: from pronouns to higher accessibility markers. In Barlow & Kemmer (Eds.), Usage based models of language (pp. 197-260), Stanford: CSLI.

Aristar, A.R. (1997), Marking and hierarchy. Types and the grammaticalization of case markers. Studies in Language, 21 (2), 313-368.

Arbib, M. A. (2015), Language evolution. An emergentist perspective. In MacWhinney and OGrady (eds), The Handbook of Language Emergence. West Sussex, UK: Wiley/Blackwell, pp. 600-623.

Bhat, D.N.S. (2004), Pronouns. Oxford [etc.]: Oxford University Press.

Bickerton, D. (1981), Roots of language. Ann Arbor, Mich.: Karoma.

Bybee, J. (2010), Language, usage, and cognition. Cambridge, UK: Cambridge University Press.

Croft, W. (2000), Explaining language change: An evolutionary approach. Harlow etc.: Longman.

Dahl, O. (2000), Egophoricity in discourse and syntax. Functions of Language, 7 (1), 37-77.

Deacon, T. (1997), The symbolic species. London: Penquin.

Dowty, D. (1991), Thematic proto-roles and argument selection. Language 67 (3): 547-619.

Dryer, M. S. (2002), Case distinctions, rich verb agreement, and word order type. Theoretical Linguistics 28: 151-157.

Dryer, M. S. (2013), Order of subject, object and verb. In Dryer & Haspelmath (Eds.), The World Atlas of Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology.

Du Bois, J.W. (1987), The discourse basis of ergativity. Language 63 (4): 805-855

Gaerdenfors, P. (2000), Conceptual spaces: The geometry of thought. Cambridge, MA: MIT.

Givon, T. (1995), Functionalism and grammar. Amsterdam/Philadelphia: John Benjamins.

Grice, H. P. (1975), Logic and conversation. In P. Cole & J. L. Morgan (Eds.), Syntax and semantics: Speech acts (Vol. 3, pp. 41-58), New York: Academic Press.

Heine, B. & Kuteva, T. (2007), The genesis of grammar. A reconstruction. Oxford: Oxford University Press.

Hopper, P. J. (1987), Emergent grammar. In Proceedings of BLS (Vol. 13, pp. 139-157),

Jackendoff, R. (2002), Foundations of language: Brain, meaning, grammar, evolution. Oxford: Oxford University Press.

Levelt, W.J.M. (1989), Speaking. From intention to articulation. Cambridge, MA: MIT Press.

Lestrade, S. (2010), The space of case. PhD thesis, Radboud University Nijmegen.

Lestrade, S. (2015a), The interaction of argument-marking strategies. In: S. Lestrade, P. de Swart, and L. Hogeweg (eds), Addenda. Artikelen voor Ad Foolen. Radboud University Nijmegen, 251-256.

Lestrade, S. (2015b), Simulating the development of bound person marking. In: H. Baayen, et al. (eds), Proceedings of the 6th Conference on Quantitative Investigations in Theoretical Linguistics. Tuebingen: University of Tuebingen.

Lestrade, S. (2015c), A case of cultural evolution: The emergence of morphological case. Linguistics in the Netherlands [AVT 32]: 105-115.

Lestrade, S. (2016), The emergence of argument marking. In: S.G. Roberts et al.(eds) The Evolution of Language: Proceedings of the 11th International Conference (EVOLANG11)

Saffran, J.R. (2001), Statistical language learning: Mechanisms and constraints. Current Directions in Psychological Science 12(4): 110-114.

Smith, K. & Kirby, S. (2008), Cultural evolution: Implications for understanding the human language faculty and its evolution. Phil. Trans. R. Soc. B, 363, 3591-3603.

Steels, L. (1997), Constructing and sharing perceptual distinctions. Machine Learning, ECML-97, 4-13.

Tomasello, M. (2003), Constructing a language: A usage-based theory of language acquisition. Cambridge, MA: Harvard University Press.

Van Valin, R. (1999), Generalized semantic roles and the syntax-semantics interface. In F. Corblin, C. Dobrovie-Sorin, & J.-M. Marandin (Eds.), Empirical issues in formal syntax and semantics 2 (pp. 373-389), The Hague: Thesus.

Yang, Ch. (2016), The price of linguistic productivity. Cambridge, MA: MIT Press.

Examples

```
## Not run:
FOUND()
RUN(.00001)
(situation=SITUATION(1))
(proposition=PROPOSITION(1, situation))
(prep=PREPARE(1, proposition, situation))
(utterance=PRODUCE(1, prep))
(interpretation=INTERPRET(2, utterance, situation))
```

```
head(population[[1]]$nouns)

## End(Not run)
```

ACTOR *Determine actor role*

Description

Given two verb roles, which of these is most prominent and hence the actor?

Usage

```
ACTOR(x, y)
```

Arguments

x	First verb role
y	Second verb role

Details

Higher meaning values are more prominent. If tie, first argument is actor

Value

numeric: 1 if first role is actor, 2 if second is.

Author(s)

Sander Lestrade

References

Van Valin, R. (1999). Generalized semantic roles and the syntax-semantics interface. In F. Corblin, C. Dobrovie-Sorin, & J.-M. Marandin (Eds.), Empirical issues in formal syntax and semantics 2 (pp. 373-389). The Hague: Thesus.

See Also

SITUATION SELECTACTOR SELECTUNDERGOER PROPOSITION REFCHECK AGENTFIRST GENERALIZE CHECKSUCCESS WORDORDER VERBMORPHOLOGY INTERPRET.INT FREQUPDATE

Examples

```
a=rep(1, 4)
b=rep(0, 4)
ACTOR(a,b)
```

`AGENTFIRST` *Actor argument first*

Description

Reorganizes constituents of an utterance such that actor is put in sentence-initial position. Only applies if corresponding word-order generalization has been made.

Usage

```
AGENTFIRST(proposition)
```

Arguments

`proposition`	Proposition of which the constituents are reordered.

Details

Applies to intransitives too, which may not be desirable.

Value

a proposition, i.e. a list:

`external`	representation of the external argument
`internal`	representation of the internal argument, if identified
`verb`	representation of the action argument
`target`	target event to be described

Author(s)

Sander Lestrade

References

Matthew S. Dryer. 2013. Order of Subject, Object and Verb. In: Dryer, Matthew S. & Haspelmath, Martin (eds.) The World Atlas of Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology. (Available online at http://wals.info/chapter/81, Accessed on 2017-05-24.)

See Also

`GENERALIZE`

Examples

```
FOUND()
situation=SITUATION(1)
(proposition=PROPOSITION(1, situation))
AGENTFIRST(proposition)
```

ALLNAS *NA vector identification*

Description

Determine whether vectors consist of NA values only.

Usage

```
ALLNAS(x)
```

Arguments

x	x can be simple vector or data frame. Latter is evaluated row-wise.

Value

T/F for single vector, vector with logicals for data frames.

Note

Used as input requirement for `VMATCH`.

Author(s)

Sander Lestrade

See Also

`VMATCH`

Examples

```
x=rep(NA, 8)
ALLNAS(x)
y=data.frame(c(1, rep(NA,2)), rep(NA,3))
ALLNAS(y)
```

ANALYZE *Determine sentence constituents*

Description

Decomposes an utterance into its constituents and their parts (e.g. verb and/or noun markers). All possible analyses are tried, the best is selected.

Usage

```
ANALYZE(hearerID, utterance, situation)
```

Arguments

`hearerID`	Pointer to hearer agent in the population
`utterance`	The utterance to be analyzed.
`situation`	The situation in which the utterance is uttered.

Details

Situation argument is necessary to determine which referential expressions have most likely been used. Suffix could be incorporated noun (l. 51-70), the rest could be a verb, a noun, a verb adposition, or noun adposition (l. 71-102) Nouns can have single suffix only (change once number is implemented; l. 92) Default interpretation is noun (cf. Heine & Kuteva) Analysis starts with identifying verb If verb can not be found by lexeme match, verb suffixes are used, if still unclear plausibility of alternative analysis is checked. Combinations of nouns with local person markers are penalized (to be removed if possessive marking is modelled), just like combinations of nouns with multiple markers (to be removed if case stacking is allowed)

Value

A data frame with the identified constituents and their analyses as entries.

Author(s)

Sander Lestrade

References

Heine, Bernd & Tania Kuteva (2007), The genesis of grammar. A reconstruction. Oxford: Oxford University Press.

See Also

`INTERPRET`

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
(utterance=PRODUCE(1, proposition))
ANALYZE(2, utterance, situation)
```

CANDIDATESCORE *Score candidate expressions*

Description

Provides each candidate expression for some meaning or function with a score in which (depending on the model settings) semantic match, lexeme activation, (relative) frequency of use, recency, collostruction frequency, semantic weight, and/or economy of expression are taken into consideration.

Usage

```
CANDIDATESCORE(lexicon, type = "referringExpression")
```

Arguments

`lexicon`	lexicon with candidate expresions
`type`	Type of function for which an expression has to be found (`referringExpression`, `nounMarker`, `verbMarker`, or `pronoun`).

Details

Collostruction frequencies are determined differently for different type of functions. The lighter, the better; recency starts with 0.

Value

Vector of scores, corresponding to the entries evaluated.

Note

Match and collostruction frequency are calculated separately before `CANDIDATESCORE` can apply. In the example below, the latter is randomly set for illustration purposes.

Author(s)

Sander Lestrade

See Also

`SELECTVERB`, `SELECTACTOR`, `SELECTUNDERGOER`, `REFCHECK`, `TOPICCOPY`, `GENERALIZE`, `CHECKSUCCESS`

Examples

```
FOUND()
lexicon=head(population[[1]]$nouns)
lexicon$match=VMATCH(lexicon[1,1:9], lexicon)
lexicon$collostruction=sample(100, nrow(lexicon))
lexicon$score=CANDIDATESCORE(lexicon)
```

CHECKSUCCESS *Determine expected communicative success*

Description

Check whether the hearer is likely to arrive at the intended role distribution and elaborate if not through explicit role marking.

Usage

```
CHECKSUCCESS(speakerID, proposition, situation)
```

Arguments

`speakerID`	Pointer to the speaker agent
`proposition`	The proposition that is to be conveyed
`situation`	The situation in which the event that the proposition refers to is embedded.

Details

Elaboration is necessary if best typing match leads to wrong distribution of roles, but not if (one of) the roles are marked one way or another First try if indexes are informative, next try appropriate pronominal case form, then check if word order is informative (if generalizations are made) N exceptions should minimally be 4(=8/ln(8)) for Yang's tolerance principle.

Value

A list, i.e. a checked and possibly elaborated proposition.

`external`	representation of the external argument
`internal`	representation of the internal argument, if identified
`verb`	representation of the action argument
`target`	target event to be described

Note

Often, the interpretation of an utterance follows automatically by world knowledge in which case no explicit marking is necessary (e.g. "man book read"). Role marking is only necessary if participants qualify for both roles equally well (e.g. "man woman see") or if a participant qualifies better for another role and outperforms the intended performer in this (e.g. "man pig kill", in which the pig is the intended actor).

Author(s)

Sander Lestrade

References

Levelt, W.J.M. 1983. "Monitoring and self-repair in speech". Cognition 14.41-104

Hurford, J. R. 1989. "Biological evolution of the saussurean sign as a component of the language acquisition device". Lingua 77:2.187-222.

Steels, L. 2003. "Language re-entrance and the inner voice". Journal of Consciousness Studies 10:4-5.173-185.

Blutner, Reinhard, Helen de Hoop \& Petra Hendriks. 2006. Optimal Communication. Stanford: CSLI.

Charles Yang (2016), The price of linguistic productivity. Cambridge, MA: MIT Press.

See Also

PREPARE

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
CHECKSUCCESS(1, proposition, situation)
```

DECOMPOSE *Decompose words into morphemes*

Description

Decomposes words into morphemes on the basis of the lexical entries in the lexicon. If multiple decompositions are possible, all are returned.

Usage

```
DECOMPOSE(hearerID, form)
```

Arguments

hearerID	Pointer to hearer agent
form	Word form that is considered for decomposition

Details

Decomposition is not trivial: Because of sloppy pronunciation (PRODUCE) and differences between speakers, mental representations of morphemes need not match one-to-one the parts of an utterance. Zero morphemes are not allowed. Reduced forms may become suffixes too. Suffixes must be minimally erosionMax long (should be automatically satisfied...). Function applies recursively (max twice)

Value

A vector with morphologically analyzed words, in which morpheme-s are separate-d by hyphen-s ("-")

Author(s)

Sander Lestrade

See Also

ANALYZE

Examples

```
FOUND()
old=world$suffixThreshold
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
(utterance=PRODUCE(1, proposition))
(utterance=gsub(' ', '', utterance))
world$suffixThreshold=20
DECOMPOSE(2, utterance)
world$suffixThreshold=old
```

DIE *Kill agents*

Description

After a prespecified number of utterances (and after having given birth to a new generation of speakers), agents are removed from the (actively speaking) population. Death agents are stored in the graveyard for later inspection.

Usage

```
DIE(agentID)
```

Arguments

agentID	Pointer to agent whose death is considered.

Details

`DIE` is called at the end of each turn, but only applies if the agent is old enough.

Value

New entry in `graveyard`.

Author(s)

Sander Lestrade

Examples

```
FOUND()
population[[1]]$age=world$deathAge+1
DIE(1)
```

EROSION *Word erosion*

Description

If a perceived form differs from the mental representation it is matched with and the form has not been set yet, the hearer agent adjusts its mental representation.

Usage

```
EROSION(hearerID, interpretation)
```

Arguments

`hearerID`	Pointer to the hearer agent whose representations might erode.
`interpretation`	Analysis of the utterance including the actually perceived forms.

Details

Forms will only be adjusted if they have not been frequently used (yet). Pronounced forms may differ from their representations because of reduction in pronunciation (cf. `REDUCE`).

Value

no actual output; the form representations of the hearer agent are updated.

Author(s)

Sander Lestrade

See Also

TURN

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
utterance=PRODUCE(1, proposition)
interpretation=INTERPRET(2, utterance, situation)
EROSION(2, interpretation)
```

FIRSTINFIRSTOUT *Order constituents by activation*

Description

If incremental production is assumed (cf. world), constituents are produced in order of activation.

Usage

```
FIRSTINFIRSTOUT(speakerID, proposition)
```

Arguments

speakerID	Pointer to speaker who's formulating an utterance
proposition	The proposition to be uttered and whose constituents are reordered.

Value

a proposition, i.e. a list:

external	representation of the external argument
internal	representation of the internal argument, if identified
verb	representation of the action argument
target	target event to be described

Author(s)

Sander Lestrade

References

Balota, D. A. & Chumbley, J. I. (1985). The locus of word-frequency in the pronunciation task: Lexical access and/or production? Journal of memory and languages, 24, 89-106.

Bock, K., and Levelt, W.J.M. (1994). Language production. Grammatical encoding. IN M.A. Gernsbacher (Ed.). Handbook of psycholinguistics (pp.741-779). New York: Academic Press

See Also

PREPARE

Examples

```
FOUND()
situation=SITUATION(1)
(proposition=PROPOSITION(1, situation))
FIRSTINFIRSTOUT(speakerID, proposition)
```

FIRSTSPEAKER *Create founding agent*

Description

Creates first agent of a lineage which only consists of a conventional symbolic lexicon (and the infrastructure to count word uses).

Usage

```
FIRSTSPEAKER()
```

Details

Start with 4 for log operations later on. Only with minimally 4 exceptions, Tolerance threshold is minority indeed

Value

age	age of agent at birth=0
generation	generation of agent
fertile	logical for fertility of agent (1 until procreated)
semupdate	logical that says that whether agent has updated its semantics already (0 at birth; cf. SEMUPDATE)
verbs	verbal lexicon
nouns	nominal lexicon
usageHistory	list with actual usages of verbs, nouns, and verb and nominal markers
commonGround	vector with lexemes recently discussed
collostructions	list with collostruction frequencies for subject-verb, object-verb, index-referent, and noun marker-noun combinations
topic	topic
wordOrder	data frame with word order frequencies
topicPosition	data frame with topic position frequencies

Author(s)

Sander Lestrade

References

Charles Yang (2016), The price of linguistic productivity. Cambridge, MA: MIT Press.

See Also

`FOUND`

Examples

```
adam=FIRSTSPEAKER()
str(adam)
```

FMATCH *Compare forms*

Description

Determine match between a given (perceived) form and a list of forms (i.e., the mental representations).

Usage

```
FMATCH(target, lexicon)
```

Arguments

`target`	The form whose matching lexeme is to be identified
`lexicon`	The lexicon in which a match is sought.

Details

Characters are matched one by one from left to right. Mismatches are weighted according to onset priority: mismatches in the beginning of a word are more important than later ones.

Value

vector of matching scores rescaled to 1-0 range.

Author(s)

Sander Lestrade

See Also

ANALYZE

Examples

```
FOUND()
(lexicon=head(population[[1]]$nouns))
target=lexicon$form[1]
FMATCH(target, lexicon)
```

FORMS *Generate forms*

Description

Generates set of unique forms for initial generation of speakers.

Usage

```
FORMS(n, length = world$wordLength, vowels = world$vowels, consonants = world$consonants)
```

Arguments

n	Number of word forms to be generated
length	Length (range) within which word forms have to fall.
vowels	Vowels that are used in the language
consonants	Vowels that are used in the language

Details

Allows for CV and VC

Value

character vector

Author(s)

Sander Lestrade

See Also

VERBS, NOUNS, FOUND, FUSE, PROCREATE

Examples

```
world$vowels; world$consonants; world$wordLength
FORMS(10)
```

FOUND *Found population*

Description

Found a new population of speakers to start a simulation.

Usage

```
FOUND(nAgents = world$nAgents)
```

Arguments

nAgents	number of agents to start with.

Value

starting population

Author(s)

Sander Lestrade

See Also

MULTIRUN

Examples

```
FOUND(4)
names(population)
```

FREQUPDATE *Update usage numbers*

Description

Update frequency numbers in lexicon and usage history.

Usage

```
FREQUPDATE(agentID, meaning, success)
```

Arguments

agentID	pointer to agent whose numbers are to be updated
meaning	Meaning (proposition or interpretation) on the basis of whose constituents the numbers in the usage history and lexicon have to be updated.
success	Logical for success of conversational turn (as number of successful uses are kept track of)

Details

Difference is made between local and third-person pronouns. +1 at the end for log operation and to prevent division by zero

Value

No actual output: updated usage history and lexicon

Author(s)

Sander Lestrade

See Also

TURN

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
population[[1]]$nouns[population[[1]]$nouns$ID%in%c(proposition$external$ID),]
population[[1]]$wordOrder
FREQUPDATE(1, proposition, success=1)
population[[1]]$nouns[population[[1]]$nouns$ID%in%c(proposition$external$ID),]
population[[1]]$wordOrder
```

FUSE — *Fuse words*

Description

Fuses lexical items with frequently co-occuring markers into new lexical items

Usage

```
FUSE(agent)
```

Arguments

`agent` Agent whose lexical items are considered for fusion.

Details

Words are only fused if the combination is used frequently enough (cf. `world`) and if meaning and form result of fusion is not in the lexicon already. Meaning of lexical item is overwritten for those meaning dimensions for which marker is specified only. Semantics of host is mixed with that of marker.

Value

No actual output: agent with updated lexicon

Author(s)

Sander Lestrade

References

Bybee, J. (2010). Language, usage and cognition. New York: Cambridge University Press.

Bybee, J. L. (1985). Morphology. a study of the relation between meaning and form. Amsterdam/Philadelphia: John Benjamins.

See Also

`SEMUPDATE`

Examples

```
FOUND()
agent=population[[1]]
agent$collostructions$flag[1,]$N=agent$nouns$ID[nrow(agent$nouns)-1]
agent$collostructions$flag[1,]$marker=agent$nouns$ID[nrow(agent$nouns)]
agent$collostructions$flag[1,]$frequency=100
agent$nouns[nrow(agent$nouns),]$nounMarker=100
agent$nouns[nrow(agent$nouns)-1,]$person=1
agent$nouns[nrow(agent$nouns),6:9]=NA
agent$collostructions$flag
tail(agent$nouns)

agent=FUSE(agent)
agent$collostructions$flag
tail(agent$nouns)
```

GENERALIZE *Apply linguistic generalizations*

Description

Checks whether the previous use of certain constructions or word orders reaches a generalization threshold. If so, the construction will be used independently from its current communicative value.

Usage

```
GENERALIZE(speakerID, proposition, situation)
```

Arguments

`speakerID`	Pointer to speaker who's considering the use standard use of a construction
`proposition`	The proposition to which the construction applies
`situation`	The communicative situation in which the utterance is made

Details

For the generalization threshold, Yang's Tolerance principle is used, which says that the number of exceptions to a rule for it to be applied/maintained/stipulated has to be below n/log(n), with n being the number of instances the rule (could have) applied. N exceptions should minimally be 4(=8/ln(8)) for Yang to make sense. Generalizations are checked, for word order first (in which grammatical order is overruled by topic generalizations), then for marking (since solutionMethod for marking sometimes dependent on word order). Noun marking first checked at general level, then for more specific dimensions of semantic role. "values=\-values[1:length(speaker$usageHistory$flag\-[[firstArgument$semRole]]$value)]" is necessary for economically stored resurrected agents (if world$saveAll=F and their behavior is checked) Third-person pronoun are only used if single third-person referent in situation.

Value

A list: the proposition, possibly in a generalized form.

`external`	representation of the external argument
`internal`	representation of the internal argument, if identified
`verb`	representation of the action argument
`target`	target event to be described

Author(s)

Sander Lestrade

References

Charles Yang (2016), The price of linguistic productivity. Cambridge, MA: MIT Press.

See Also

PREPARE

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
GENERALIZE(1, proposition, situation)
population[[1]]$wordOrder[3,2:3]=9999
population[[1]]$wordOrder
GENERALIZE(1, proposition, situation)
```

GROUP *Group words into constituents*

Description

Determines each possible constituent ordering (assuming adjacency) of an utterance. Given A B C V, in which V is identified as the verb, B could be a marker of A, or C could be a marker of B.

Usage

```
GROUP(hearerID, analysis)
```

Arguments

hearerID	Pointer to the hearer agent
analysis	Analysis of the utterance in which the individual lexemes have been determined and the verb has been identified.

Details

VerbAdpositions are for topic cross reference only, and may be put on top of verb suffixes. VerbAdpositions are reanalyzed as verbSuffix if index=TRUE and no other verb suffixes (then no proper suffix was available) Only non-local-person noun markers, to be removed if possessive marking is modelled.

Value

list of all possible groupings

[[1]]	First possible grouping analysis
[[2]]	Second possible grouping analysis, if possible, etc.

Author(s)

Sander Lestrade

See Also

INTERPRET

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
utterance=PRODUCE(1, proposition)
utterance=paste(utterance, unlist(strsplit(utterance, ' '))[1])
analysis=ANALYZE(2, utterance, situation)
GROUP(2, analysis)
```

INTERPRET *Interpret utterance*

Description

Determines the best interpretation of an utterance given the situational context. Compares different interpretations if multiple analyses are possible and chooses most likely one given context.

Usage

```
INTERPRET(hearerID, utterance, situation)
```

Arguments

hearerID	Pointer to the hearer agent
utterance	The utterance to be interpreted
situation	Set of events in which utterance was used

Details

#first use explicit role marking #then word order (if still necessary) #then verb morphology (idem)

Value

interpretation, i.e. a list:

external	representation of the external argument
internal	representation of the internal argument, if identified
verb	representation of the action argument
target	target event identified on the basis of interpretation, including matching scores

Author(s)

Sander Lestrade

See Also

`TURN`

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
utterance=PRODUCE(1, proposition)
INTERPRET(2, utterance, situation)
```

`INTERPRET.INT` *Develop an interpretation*

Description

Translates utterance analysis into a proposition (i.e., an interpretation) and determines match with ongoing events.

Usage

```
INTERPRET.INT(hearerID, analysis, situation)
```

Arguments

`hearerID`	Pointer to the hearer agent that is interpreting an analysis
`analysis`	The analysis to be translated
`situation`	Situation in which utterance is interpreted.

Details

`INTERPRET.INT` works internal to `INTERPRET`, which compares the interpretations of the different possible analyses. VerbAdpositions overrule verbSuffixes...

Value

`external`	representation of the external argument
`internal`	representation of the internal argument, if identified
`verb`	representation of the action argument
`target`	target event identified on the basis of interpretation, including matching scores

Author(s)

Sander Lestrade

See Also

`INTERPRET`

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
utterance=PRODUCE(1, proposition)
analysis=ANALYZE(2, utterance, situation)
(analysis=PROTOINTERPRETATION(2, analysis))
INTERPRET.INT(2, analysis, situation)
```

MAX *Find maximum value*

Description

Extension of standard `max` and `min` functions with which rank position(s) can be specified and result can be either rank or value.

Usage

```
MAX(vector, rank = 1, value = FALSE, rank.adjust = TRUE, forceChoice = FALSE)
```

Arguments

`vector`	Vector in which maximum/minimum element needs to be identified
`rank`	value(s) or rank(s) of maximum values.
`value`	Should value or rank be returned?
`rank.adjust`	If maximum value of range of ranks exceeds vector length, should this be adjusted?
`forceChoice`	In case of ties, should all results be returned or only one?

Value

numeric vector (either value or rank)

Warning

If minimum value (of a range of) rank(s) exceeds vector length, results are meaningless.

Author(s)

Sander Lestrade

See Also

MIN, NOUNS, SITUATION, SELECTVERB, SELECTACTOR, SELECTUNDERGOER, REFCHECK, TOPICCOPY, GENERALIZE, CHECKSUCCESS, ANALYZE, TYPEMATCH, NOUNMORPHOLOGY, VERBMORPHOLOGY, INTERPRET.INT, INTERPRET, NOUNDESEMANTICIZATION, VERBDESEMANTICIZATION, SEMUPDATE, DIE

Examples

```
a=rep(1:10, 2)
MAX(a, rank=1:3, value=TRUE, forceChoice=TRUE)
MIN(a, rank=1:3, value=TRUE, forceChoice=TRUE)
```

NOUNDESEMANTICIZATION *Bleach word meaning*

Description

Update meaning representations of nouns/verbs on the basis of usage history.

Usage

```
NOUNDESEMANTICIZATION(agent)
```

Arguments

agent	Agent whose lexicon is to be updated

Details

Cf. Heine and Kuteva p.39: freq is epiphenomenon of extension, not cause; extension by combinatorial flexibility 8 is minimum freq from which Yang applies.

Value

agent (with updated nominal representations)

Author(s)

Sander Lestrade

References

Hopper, P. J. & Traugott, E. C. (2003). Grammaticalization. Cambridge: Cambridge University Press.

Heine, B. & Kuteva, T. (2007). The genesis of grammar. a reconstruction. Oxford: Oxford University Press.

Bybee, J. (2010). Language, usage and cognition. New York: Cambridge University Press.

See Also

`SEMUPDATE`

Examples

```
#only effective if usage history is non-empty
FOUND()
population[[1]]=NOUNDESEMANTICIZATION(population[[1]])
population[[1]]=VERBDESEMANTICIZATION(population[[1]])
```

`NOUNMORPHOLOGY` *Interpret nominal morphology*

Description

Use noun markers to determine event-role distribution (i.e., who is actor and who is undergoer).

Usage

```
NOUNMORPHOLOGY(hearerID, analysis)
```

Arguments

`hearerID`	Pointer to hearer agent who's developing an analysis
`analysis`	Analysis of utterance (result of `ANALYZE`) in which roles have to be determined.

Details

Marker overrules suffix with same host Future work: allow for oblique roles.

Value

Analysis (dataframe) with roles assigned on the basis of nominal markers.

Author(s)

Sander Lestrade

See Also

`INTERPRET`, `VERBMORPHOLOGY`

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
prep=PREPARE(1, proposition, situation)
utterance=PRODUCE(1, prep)
analysis=ANALYZE(2, utterance, situation)
grouping=GROUP(2, analysis)
(analysis=NOUNMORPHOLOGY(2, grouping[[length(grouping)]]))
analysis$role
#repeat if no nounAdposition is identified.
```

NOUNS *Generate nominal lexicon*

Description

Generate nominal lexicon for founding agents

Usage

```
NOUNS(n = world$nNouns, local = world$local)
```

Arguments

n	Number of nominal lexemes
local	Should agents have lexemes to refer to speech-act participants (i.e. 'I/me' and 'you')?

Details

Minimally 2 entries are necessary to create a dataframe.

Value

data frame with randomly generated forms, their meaning representations in terms of numeric vectors, and frequency counters.

Author(s)

Sander Lestrade

See Also

VERBS, FIRSTSPEAKER, SEMUPDATE

Examples

```
NOUNS(10)
```

PERSONUPDATE *Adjust person value*

Description

Adjust person value of noun from third to local (first or second) if it has been recruited frequently to express local reference.

Usage

```
PERSONUPDATE(agent)
```

Arguments

agent	Agent whose nominal lexicon is considered.

Details

Multiple verb markers with same person are taken care of too. Redundant local pronouns and indexes are removed.

Value

Agent (with updated lexicon)

Author(s)

Sander Lestrade

References

Zeevat, Henk. 2007. "Simulating recruitment in evolution". Cognitive Foundations of Interpretation ed. by G. Bouma, I. Kraemer & J. Zwarts, 175-194. Amsterdam: Royal Netherlands Academy of Arts and Sciences.

See Also

SEMUPDATE

Examples

```
FOUND()
population[[1]]=PERSONUPDATE(population[[1]]) #only effective if pronouns have been recruited
```

PREPARE — *Prepare a proposition for production*

Description

Prepare a proposition for production by checking if it will be intelligible and applying generalizations.

Usage

```
PREPARE(speakerID, proposition, situation)
```

Arguments

`speakerID`	Pointer to speaker agent
`proposition`	Proposition that is prepared for production
`situation`	Situation in which proposition is to be uttered.

Details

`PREPARE` involves a number of subroutines: If role distribution is unclear, agents checks if this can be resolved with pronominal case forms and if not with noun markers (`CHECKSUCCESS`). If referential expression is too weak, stronger expressions are used (`REFCHECK`). If generalizations such as PutAgentFirst or IndexFirstPerson are made, they are applied (`GENERALIZE`). If words are frequently used, their forms are reduced (`REDUCE`). Etc. Ingredients of proposition are ordered by activation before other principles apply

Value

a proposition, i.e. a list:

`external`	representation of the external argument, possibly including role marking
`internal`	representation of the internal argument, if identified, , possibly including role marking
`verb`	representation of the action argument, possibly including person indexing
`target`	target event to be described

Author(s)

Sander Lestrade

See Also

`TURN`

Examples

```
FOUND()
situation=SITUATION(1)
(proposition=PROPOSITION(1, situation))
PREPARE(1, proposition, situation)
#result need not be different from simple proposition,
#depends on generalizations and typing scores
```

PROCREATE *Generate new generation of agents*

Description

Generate new generation of agents if (to be) parent generation is old enough.

Usage

```
PROCREATE(speakerID, hearerID)
```

Arguments

speakerID	Pointer to first parent
hearerID	Pointer to second parent

Details

New generation is mix of vocabularies of parents (if `world$crossover` is T), with emptied usage histories. Agents procreate after number of utterances specified by `world$procreationAge`. Meanings of words that have not been used by their parents are modified slightly.

Value

Set of new agents

Author(s)

Sander Lestrade

See Also

TALK

Examples

```
FOUND()
population[[1]]$age=population[[2]]$age=world$procreationAge*world$deathAge+1
PROCREATE(1,2)
```

PRODUCE *Produce utterance*

Description

Turns proposition into actual utterance.

Usage

```
PRODUCE(speakerID, prep)
```

Arguments

speakerID	Pointer to speaker agent
prep	Proposition to be uttered

Details

Internal markers are produced closest to verb (cf. Dryer); not exploited by hearer.

Value

Character string

Author(s)

Sander Lestrade

References

Matthew S. Dryer. 2013. Order of Subject, Object and Verb. In: Dryer, Matthew S. & Haspelmath, Martin (eds.) The World Atlas of Language Structures Online.

See Also

TURN

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
PRODUCE(1, proposition)
```

`PROPOSITION` *Develop initial proposition*

Description

Develop initial proposition that consists of the expressions that refer to the participants of the event to be expressed (both objects and action). Proposition will be elaborated upon in later stages of the production processs, and word may be replaced later if they turn out to be insufficient (cf. `PREPARE`).

Usage

```
PROPOSITION(speakerID, situation)
```

Arguments

`speakerID`	Pointer to speaker agent
`situation`	Situation with target event to be referred to and number of distractor event

Details

Words are ranked on the basis of a combination of semantic match (how well does word refer to its participant), frequency, and recency (cf. `CANDIDATEORDER`). The first word to be sufficiently distinctive is selected for expression.

Value

a proposition, i.e. a list:

`external`	representation of the external argument
`internal`	representation of the internal argument
`verb`	representation of the action argument
`target`	target event to be described

Author(s)

Sander Lestrade

See Also

`TURN`

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
```

PROTOINTERPRETATION *Develop interpretation*

Description

Develop interpretation of an utterance using simple heuristics only (i.e., ignoring grammatical markers and tendencies). Applies if grammar either did not develop yet, or does not suffice for disambiguation.

Usage

```
PROTOINTERPRETATION(hearerID, analysis)
```

Arguments

hearerID	Pointer to hearer agent
analysis	Analysis of the utterance to be interpreted (cf. ANALYZE)

Details

If only one role is unclear, it follows from simple reasoning (V has x and y role, A is x, then B must be y). If both roles are unclear, TYPEMATCH is used.

Value

a dataframe, i.e. the analysis input in which the role column is updated.

Author(s)

Sander Lestrade

See Also

VERBMORPHOLOGY, INTERPRET

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
utterance=PRODUCE(1, proposition)
(analysis=ANALYZE(2, utterance, situation))
PROTOINTERPRETATION(2, analysis)
```

`REDUCE` *Reduce length of expressions*

Description

Reduces length of frequently or recently used expressions by removing final character.

Usage

```
REDUCE(speakerID, proposition)
```

Arguments

`speakerID`	Pointer to speaker agent.
`proposition`	Proposition with words whose forms may be reduced.

Details

Reduction is an online production process only. It does not affect the lexical representation of the speaker (but cf. `EROSION`)

Value

a proposition, i.e. a list:

`external`	representation of the external argument, possibly with shortened form
`internal`	representation of the internal argument, if identified, possibly with shortened form
`verb`	representation of the action argument, possibly with shortened form
`target`	target event to be described

Author(s)

Sander Lestrade

References

Nettle, D. (1999). Linguistic diversity. New York: OUP.

Jurafsky, Daniel, Alan Bell, Michelle Gregory & William D. Raymond. 2001. "Probabilistic relations between words: Evidence from reduction in lexical production". In: J. Bybee and P. Hopper (eds), Frequency and the emergence of linguistic structure, 229-255. Amsterdam/Philadelphia. John Benjamins.

See Also

`PREPARE`

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
REDUCE(1, proposition) #only effective if proposition includes frequently/recently used words
```

REFCHECK *Check referential capacity*

Description

Check if pronoun is sufficiently strong to establish reference to referent. If not, another word is recruited for support, the pronoun being suffixed to the verb.

Usage

```
REFCHECK(speakerID, proposition, situation)
```

Arguments

`speakerID`	Pointer to speaker agent
`proposition`	Proposition in which the referential expressions are checked
`situation`	Situation in which referential relations have to be established

Details

Strength is determined by formal mass, i.e. simple word length. Non-local arguments are matched with real-world argument; local pronoun with role, after which marker is removed. If there's no local pronominal paradigm yet, select prominent noun for local ref

Value

a proposition, i.e. a list:

`external`	representation of the external argument, checked for strength
`internal`	representation of the internal argument, if identified, checked for strength
`verb`	representation of the action argument, possibly including person indexing if original expression for (one of the) event participants fell short
`target`	target event to be described

Author(s)

Sander Lestrade

References

Zeevat, Henk. 2007. "Simulating recruitment in evolution". Cognitive Foundations of Interpretation ed. by G. Bouma, I. Kraemer & J. Zwarts, 175-194. Amsterdam: Royal Netherlands Academy of Arts and Sciences.

Ariel, M. (1999). The development of person agreement markers: From pronouns to higher accessibility markers. In M. Barlow & S. Kemmer (Eds.), Usage based models of language (p. 197-260). Stanford: CSLI.

See Also

`PREPARE`

Examples

```
FOUND()
situation=SITUATION(1)
(proposition=PROPOSITION(1, situation))
REFCHECK(1, proposition, situation)
#only effective if words have grammaticalized already
```

RESCALE *Rescale vector values*

Description

Rescale vector values to -1:1 range (or 0:1 if there are no negative values)

Usage

```
RESCALE(x)
```

Arguments

`x`	Vector to be rescaled

Value

Numeric vector, with abs(max value) of 1

Author(s)

Sander Lestrade

See Also

`CANDIDATESCORE`, `FREQUPDATE`

Examples

```
RESCALE(-10:5)
```

RUN *Run simulation*

Description

Run simulation for specified number of hours. Language change beyond phonological change to happen generally requires multiple hours of simulation.

Usage

```
RUN(nHours = 1)
```

Arguments

`nHours`	Number of hours to run simulation.

Value

No output. Objects in work space (`population`, `graveyard`, `situation`, `proposition`, `utterance`, `interpretation`) are adapted.

Author(s)

Sander Lestrade

See Also

`MULTIRUN`

Examples

```
FOUND()
## Not run: RUN(.000001)
```

SELECTACTOR *Find actor expression*

Description

Select best expression for actor/undergoer/verb participant in the event to be described

Usage

```
SELECTACTOR(speakerID, situation, verb = NULL)
```

Arguments

`speakerID`	Pointer to speaker agent
`situation`	Situation in which event to be described is situated
`verb`	Pointer to verb lexeme used in the utterance to be formulated (if present already)

Details

Verb is relevant because of collostruction frequencies: some agents are more likely to be mentioned given certain verbs (cf. `CANDIDATEORDER`). Works other way around for `SELECTVERB`.

Value

A dataframe with the lexical representation of the agent/undergoer/verb.

Author(s)

Sander Lestrade

See Also

`PROPOSITION`

Examples

```
FOUND()
situation=SITUATION(1)
situation[situation$target==1,]
SELECTACTOR(1, situation)
SELECTVERB(1, situation)
if(!is.na(situation[situation$target==1,]$U1)){
SELECTUNDERGOER(1, situation)
}
```

`SEMUPDATE` *Update lexicon*

Description

Update meaning lexicon on the basis of usage. Involves `NOUNDESEMANTICIZATION`, `VERBDESEMANTICIZATION`, `FUSE`, and `PERSONUPDATE`. Also, words that have become meaningless are replaced.

Usage

```
SEMUPDATE(agentID)
```

Arguments

`agentID`	Pointer to agent whose lexicon is to be updated.

Value

agent with updated lexicon

Author(s)

Sander Lestrade

See Also

`talk`

Examples

```
FOUND()
SEMUPDATE(1)
```

`SITUATION` *Create situational context*

Description

Create situational context that consists of set of events among which the target event to be described.

Usage

```
SITUATION(speakerID)
```

Arguments

`speakerID`	Pointer to speaker agent whose concepts are used to create situation.

Details

Events are generated on the basis of agents' world knowledge. In principle, qualified participants are more likely than unqualified ones (e.g. books are read, not eaten; cf. `world$roleNoise`; `world$referenceNoise`). Local person always known, so if world$local==T, oddsNew for Dahl numbers are adjusted. Situations with multiple events are more likely than situations with single event. Locals are animate. If none of the candidates qualifies argument criterium, only recency is used for topichood (cf. DuBois: preference for actor topic)

Value

dataframe with sets of vectors that specify actions and actors, and if present undergoer participants.

Author(s)

Sander Lestrade

References

John W. DuBois (1987), The discourse basis of ergativity. Language 63 (4)

See Also

TURN

Examples

```
FOUND()
SITUATION(1)
```

SUCCESS — *Determine communicative success*

Description

Determine communicative success by comparing intention of speaker (proposition) and interpretation of hearer.

Usage

```
SUCCESS(proposition, interpretation, situation)
```

Arguments

`proposition`	Intended/speaker meaning
`interpretation`	Interpretation/hearer meaning
`situation`	Contextual situation in which communication took place.

Details

If there are no distractor events ongoing, success is determined by comparing the speaker and hearer meanings; otherwise, communication is successful if the same target event is selected.

Value

Logical: 1 for success; 0 for failure

Author(s)

Sander Lestrade

See Also

TURN

Examples

```
FOUND()
situation=SITUATION(1)
(proposition=PROPOSITION(1, situation))
utterance=PRODUCE(1, proposition)
(interpretation=INTERPRET(2, utterance, situation))
SUCCESS(proposition, interpretation, situation)
```

SUMMARY *Summarize simulation results*

Description

Summarize results of simulation

Usage

```
SUMMARY()
```

Value

List and plots

`generation`	generation of present agent
`order`	word-order generalizations
`topic`	topic-order generalizations
`index`	verb-marker generalizations
`person`	generalizations about role marking per person (e.g. first person undergoers should be marked)
`actor`	actor-marking generalizations on the basis of meaning (e.g. all actors with a zero value on the first dimension should be marked)
`undergoer`	undergoer-marking generalizations on the basis of meaning (e.g. all undergoer with a zero value on the first dimension should be marked)
`markers`	which words were mostly used as markers
`nounMarkerUse1`	proportion of role-marking in total
`nounMarkerUse12`	proportion of role-marking of transitive events
`first`	lexemes with first-person meaning
`second`	lexemes with second-person meaning

Author(s)

Sander Lestrade

See Also

CHECKMARKER, HISTORY

Examples

```
## Not run:
FOUND()
RUN(.0001) #create results to summarize: first generation has to die
world$deathAge=10
DIE(1)
SUMMARY()

## End(Not run)
```

TALK *Let agents talk*

Description

Sample two agents and let them talk with each other.

Usage

```
TALK(nTurns)
```

Arguments

nTurns	Number of communicative turns a conversation last before new agents are selected for communication.

Details

Young agents are less likely to talk with each other.

Value

On screen conversation. (Underlying update of usage history of talking agents.)

Author(s)

Sander Lestrade

See Also

RUN

Examples

```
FOUND()
## Not run: TALK(4)
```

TOPICCOPY *Make anaphoric copy of topic*

Description

Make verb-adjacent anaphoric copy of contrastive topic. Only applies if topics are moved to first position (after this generalization is made).

Usage

```
TOPICCOPY(speakerID, proposition)
```

Arguments

`speakerID`	Pointer to speaker agent
`proposition`	Proposition that agent is formulating

Details

Anaphoric copies only need to distinguish topic from other argument for reestablished/non-continuous topics (cf. Givon)

Value

a proposition, i.e. a list:

`external`	representation of the external argument, possibly including role marking
`internal`	representation of the internal argument, if identified, possibly including role marking
`verb`	representation of the action argument, possibly including person indexing
`target`	target event to be described

Author(s)

Sander Lestrade

References

T. Givon (1976), "Topic, pronoun, and grammatical agreement", In: C. Li (Ed.), Subject and topic, New York, etc.: Academic Press, Inc, 149-188.

See Also

`TOPICFIRST`

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
proposition$verb$topic=0; if('internal'%in%names(proposition)){proposition$internal$topic=0}
proposition$external$topic=1; proposition$external$recency=10
world$topicCopy=FALSE
proposition=TOPICFIRST(1, proposition)
PRODUCE(1, proposition)
proposition=TOPICCOPY(1, proposition)
PRODUCE(1, proposition)
```

TOPICFIRST *Put topic in first position*

Description

Put topic of the utterance in first position if such a tendency was observed in and therefore generalization was made on the basis of previous utterances.

Usage

```
TOPICFIRST(speakerID, proposition)
```

Arguments

`speakerID`	Pointer to speaker agent
`proposition`	Proposition in which topic argument is to be moved

Value

a proposition, i.e. a list:

`external`	representation of the external argument, possibly including role marking
`internal`	representation of the internal argument, if identified, possibly including role marking
`verb`	representation of the action argument, possibly including person indexing
`target`	target event to be described

, in which the topic argument is put first

Author(s)

Sander Lestrade

References

Tomlin, R. S. (1986). Basic word order: Functional principles (Vol. 13). Routledge

Ferrer-i-Cancho, R. (2014). Why might SOV be initially preferred and then lost or recovered? a theoretical framework. In: Proceedings of the 10th international conference (evolang10), pp. 66-73.

Bates, E., & MacWhinney, B. (1987). Competition, variation, and language learning. Mechanisms of language acquisition, 157-193.

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
PRODUCE(1, proposition)
proposition=TOPICFIRST(1, proposition)
PRODUCE(1, proposition)
```

TURN *Organize communicative turn*

Description

Organize communicative turn in conversation. Involves generating a situation (`SITUATION`), developing an utterance (`PROPOSITION`, `PREPARE`, `PRODUCE`), interpreting the utterance (`INTERPRET`), and updating the lexicon (`FREQUPDATE`, `EROSION`).

Usage

```
TURN(speakerID, hearerID)
```

Arguments

`speakerID`	Pointer to speaker agent
`hearerID`	Pointer to hearer agent

Value

Character string on screen, real output: agents with updated usage history.

Author(s)

Sander Lestrade

See Also

`TALK`

Examples

```
FOUND()
TURN(1,2)
```

`TYPEMATCH` *Determine role qualification*

Description

Determine event-role distribution of participants on the basis of role qualification.

Usage

```
TYPEMATCH(hearerID, analysis)
```

Arguments

`hearerID`	Pointer to hearer agent
`analysis`	Analyzed utterance in which verb and participants have been identified

Details

Typematch is only necessary if explicit markers and/or grammar are not sufficiently informative

Value

Analysis with event-role assignment (if possible on the basis of role qualifications)

Author(s)

Sander Lestrade

References

Aristar, A. R. 1997. "Marking and hierarchy. Types and the grammaticalization of case markers". Studies in Language 21:2.313-368.

See Also

`PROTOINTERPRETATION`

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
utterance=PRODUCE(1, proposition)
analysis=ANALYZE(2, utterance, situation)
TYPEMATCH(2, analysis)
```

VERBFINAL	*Put verb final*

Description

Reorganizes constituents of an utterance such that verb is put in sentence-final position. Only applies if corresponding word-order generalization has been made.

Usage

```
VERBFINAL(proposition)
```

Arguments

proposition	Proposition of which the constituents are reordered.

Value

a proposition, i.e. a list:

external	representation of the external argument
internal	representation of the internal argument, if identified
verb	representation of the action argument
target	target event to be described

Author(s)

Sander Lestrade

References

Matthew S. Dryer. 2013. Order of Subject, Object and Verb. In: Dryer, Matthew S. & Haspelmath, Martin (eds.) The World Atlas of Language Structures Online. Leipzig: Max Planck Institute for Evolutionary Anthropology. (Available online at http://wals.info/chapter/81, Accessed on 2017-05-24.)

See Also

GENERALIZE

Examples

```
FOUND()
situation=SITUATION(1)
(proposition=PROPOSITION(1, situation))
VERBFINAL(proposition)
```

VERBMORPHOLOGY *Interpret verbal morphology*

Description

Determine anaphoric reference of verb markers (either verb adpositions or suffixes).

Usage

```
VERBMORPHOLOGY(hearerID, analysis)
```

Arguments

hearerID	Pointer to hearer agent who's developing an analysis
analysis	Analysis of utterance (result of ANALYZE) in which roles have to be determined.

Details

If verb marker cannot be resolved anaphorically, it is reinterpret as a deictic argument.

Value

Analysis (dataframe) with resolved reference of verb markers.

Author(s)

Sander Lestrade

See Also

INTERPRET, NOUNMORPHOLOGY

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
proposition$verb$topic=0; if('internal'%in%names(proposition)){proposition$internal$topic=0}
proposition$external$topic=1; proposition$external$recency=10
proposition=TOPICFIRST(1, proposition)
utterance=PRODUCE(1, proposition)
analysis=ANALYZE(2, utterance, situation)
grouping=GROUP(2, analysis)
for(i in 1:length(grouping)){
if('verbAdposition'%in%grouping[[i]]$role){
print(VERBMORPHOLOGY(2, grouping[[i]]))
} }
```

VERBS *Generate verbal lexicon*

Description

Generate verbal lexicon for founding agents

Usage

```
VERBS(n = world$nVerbs)
```

Arguments

n	Number of verbal lexemes

Value

data frame with randomly generated forms, their meaning representations in terms of sets of numeric vectors (for action, actor role, and undergoer role), and frequency counters.

Author(s)

Sander Lestrade

See Also

NOUNS, FIRSTSPEAKER, SEMUPDATE

Examples

```
VERBS(10)
```

VMATCH *Compare vectors*

Description

Compare vectors properly taking into account uniform vectors (with same values on all dimensions) and non-specified dimensions.

Usage

```
VMATCH(x, y, incomparable = 0, noise=TRUE)
```

Arguments

`x`	target vector
`y`	vector (numeric or data frame/list) or set of vectors whose similarity to the target is to be determined
`incomparable`	Value to be returned for incomparable vectors, in which all dimension pairs contain underspecified values
`noise`	logical for addition of noise to outcome (default is TRUE)

Details

Differences between vectors are determined per dimension, weighted (cf. `world$weigh`), and then averaged. If vectors are not specified for certain target dimensions, this does not count as a mismatch.

Value

numeric

Author(s)

Sander Lestrade

See Also

`ACTOR`, `NOUNS`, `SITUATION`, `SELECTVERB`, `SELECTACTOR`, `SELECTUNDERGOER`, `PROPOSITION`, `REFCHECK`, `TOPICCOPY`, `GENERALIZE`, `CHECKSUCCESS`, `ANALYZE`, `TYPEMATCH`, `NOUNMORPHOLOGY`, `VERBMORPHOLOGY`, `INTERPRET.INT`, `SUCCESS`, `PERSONUPDATE`, `FUSE`, `SEMUPDATE`

Examples

```
FOUND()
vectors=head(population[[1]]$nouns[,1:9])
target=vectors[1,]
vectors[2,]=NA
VMATCH(target, vectors)
```

`WORDORDER` *Use word order for interpretation*

Description

Use observed word-order tendencies for interpretation of role distribution. E.g., if agents were observed to come first mostly, assign first constituent agent role.

Usage

```
WORDORDER(hearerID, analysis)
```

Arguments

`hearerID`	Pointer to hearer agent
`analysis`	Analysis of utterance in which roles have to be determined.

Details

To check if word order can be used, Yang's Tolerance principle is used.

Value

a proposition, i.e. a list:

`external`	representation of the external argument, possibly including role marking
`internal`	representation of the internal argument, if identified, possibly including role marking
`verb`	representation of the action argument, possibly including person indexing
`target`	target event to be described

Author(s)

Sander Lestrade

References

Charles Yang (2016), The price of linguistic productivity. Cambridge, MA: MIT Press.

See Also

`INTERPRET`

Examples

```
FOUND()
situation=SITUATION(1)
proposition=PROPOSITION(1, situation)
utterance=PRODUCE(1, proposition)
analysis=ANALYZE(2, utterance, situation)
WORDORDER(2, analysis)
population[[2]]$wordOrder[2,]$success=999 #makes AUV standard
population[[2]]$wordOrder
WORDORDER(2, analysis)
```

world *Model parameters*

Description

Model parameters that hold during the simulation for the entire lineage

Usage

```
data("world")
```

Format

The format is: List of 64 $ nAgents : num 2 $ deathAge : num 2000 $ procreationAge : num 0.55 $ crossover : logi TRUE $ replace : logi TRUE $ weigh : logi TRUE $ distinctions : num [1:9] 2 2 2 2 2 9 9 9 9 $ wordLength : int [1:3] 8 9 10 $ vowels : chr [1:6] "a" "e" "i" "o" ... $ consonants : chr [1:15] "b" "d" "f" "g" ... $ nNouns : num 499 $ nVerbs : num 199 $ proportionIntrans : num 0.2 $ linkingPreference : num 5 $ local : logi TRUE $ useCommonGround : logi TRUE $ commonGroundStart : num 3 $ dahlS : num [1:4] 21 10 21 44 $ dahlA : num [1:4] 38 22 33 7 $ dahlO : num [1:4] 3 3 10 84 $ oddsNewA : num 0.0333 $ oddsNewOther : num 0.25 $ referenceNoise : num 0.2 $ roleNoise : num 0.3 $ nEvents : int [1:11] 10 11 12 13 14 15 16 17 18 19 ... $ nTurns : int [1:16] 5 6 7 8 9 10 11 12 13 14 ... $ talkAge : num 0.05 $ turnChange : num [1:2] 2 1 $ personTopicality : num [1:4] 2 1 2 2 $ topicContinuity : num [1:2] 3 1 $ checkSuccess : logi TRUE $ solutionMethod : chr "bestMarker" $ reductionFrequencyThreshold : num 0.05 $ reductionCollostructionThreshold: num 3 $ reductionRecencyThreshold : num 2 $ formSetFrequency : num 3 $ suffixThreshold : num 6 $ refCheck : logi TRUE $ referenceThreshold : num 4 $ generalization : logi TRUE $ firstInFirstOut : logi TRUE $ distinctiveness : num 0.05 $ candidateScoring : chr "all" $ frequency : chr "relative" $ activationImpact : num 0.2 $ collostructionImpact : num 0.2 $ semanticWeightImpact : num 0.1 $ economyImpact : num 0.1 $ recencyDamper : num 5 $ activationNoise : num 2 $ functionBlocking : logi TRUE $ wordOrder : logi TRUE $ topicCopy : logi TRUE $ semUpdateAge : num 0.5 $ erosion : logi TRUE $ erosionMax : num 2 $ formBlocking : logi TRUE $ desemanticization : logi TRUE $ desemanticizationCeiling : num 0.4 $ desemanticizationPower : num 2 $ minimalSpecification : num 1 $ verbalRoleMarker : logi FALSE $ semUpdateThreshold : num 0.02 $ saveAll : logi FALSE

Details

`nAgents`: number of founding agents

`deathAge`: age, in number of utterances, at which agents die

`procreationAge`: point at which agents procreate (relative to their death age). If NA, no offspring. Best to procreate after semUpdate;)

`crossover`: If true, lexicon of off spring is combination of those of parents. If false, each parent will get a child with identical lexicon

`replace`: Should minor modifications be made to non-used words?

`weigh`: In comparing meanings and determining whose the actor, should meaning dimensions be equally important (F) or should first dimension be more important than second, but less important than second plus third, etc. (T). Slows down simulation in combination with high number of events per situation (>10)

The following set of parameters applies to the lexicon specifically:

`distinctions`: dimensionality and distinctionality of meaning representations (distinctions are normalized to 0–1 range).

`wordLength`: initial length of words, can be single valued or range.

`vowels`: vowels of alphabet constituting the words

`consonants`: consonants of alphabet constituting the words

`nNouns`: number of nouns in the lexicon

`nVerbs`: number of verbs in the lexicon

`proportionIntrans`: proportion of intransitive verbs in both lexicon and events. Probably .5 in real life, but smaller in the interest of argument marking

`linkingPreference`: preference of external (internal) predicate role for higher (lower) values ("prominent performers"). linkingPreference is odds of highest against lowest role/value. 1 is no preference.

`local`: Do agents have the words/the possibility to refer to themselves?

The following set of parameters applies to the generation of the situational context, i.e., the set of target and distractor events (cf. Steels).

`useCommonGround`: Do speech participants share a common ground or are all words/concepts equally likely and accessible.

`commonGroundStart`: number of elements (excluding speech participants) that are present in common ground when conversation starts. Elements are randomly selected from lexicon.

`dahlS`: odds for intransitive subject to be 1, 2, 3Animate, and 3Inanimate person respectively (based on Dahl 2000, 45-51)

`dahlA`: odds for external role to be 1, 2, 3Animate, and 3Inanimate person. First three numbers are summed if local==F.

`dahlO`: odds for internal role to be 1, 2, 3Animate, and 3Inanimate person.

`oddsNewA`: odds for a non common-ground element to enter as A argument of one of the events in the situation (element will be added to the common ground if discussed; cf. DuBois 1987: 828, Table 7)

`oddsNewOther`: odds for a non common-ground element to enter as S or O argument of one of the events in the situation (element will be added to the common ground if discussed)

`referenceNoise`: how much "referential" noise is there in the world (0–1)? The less noise, the closer the world matches the concepts and relations in the language.

`roleNoise`: How much noise is there in the world with respect to the event roles that nouns are expected and found to perform.

`nEvents`: Number of events that are ongoing in speech situation, one of which is selected to talk about. If set to 1, no distractor events occur.

The following set of parameters applies to the conversations two agents have:

`nTurns`: What is the range of communicative turns conversations consist of (before common ground is reset) `talkAge`: At which point (relative to their death age) do agents start to talk? (Until then, they only listen) If zero, less learning from parents

`turnChange`: odds for speech-act participants to change speech-act roles

`personTopicality`: Preference for speaker, addressee, animate third person, and inanimate third person respectively to be the topic of the utterance and participant in a situation (based on Dahl's S and A numbers)

`topicContinuity`: odds for continuing with the same topic vs starting a new one

The following set of parameters applies to the production process:

`checkSuccess`: Should expected recovery of meaning be checked? (cf. Aristar for "typing" scores)

`solutionMethod`: If check success shows utterance should be elaborated, how is this done? Options: firstFail, bestMarker, worstPerformer, random, secondArgument, internal, external, both

`reductionFrequencyThreshold`: Relative frequency threshold at which forms get reduced.

`reductionCollostructionThreshold`: Absolute collostruction-frequency threshold at which forms get reduced.

`reductionRecencyThreshold`: idem for recency

`formSetFrequency`: number of times an item has to be used before its form is set, after which its representation will no longer change

`suffixThreshold`: productionEffort threshold (in number of characters) at which words markers suffixed to their host

`refCheck`: Should referential threshold be reached for words to refer?

`referenceThreshold`: production effort (in number of characters) necessary for an utterance to be sufficiently referential (a la Ariel). If lower, a more expressive expression is added sentence first.

`generalization`: Should agents try to derive generalizations from the tendencies they observe? Applies from second generation onwards only (cf. Yang)

`firstInFirstOut`: Is utterance production incremental? (cf. Bock and Levelt)

The following set of parameters applies both to the production and interpretation process:

`distinctiveness`: If two forms are similar in meaning (or in role typing in case of global marking), how big should the difference be for the speaker to think the distinction is sufficiently clear?

`candidateScoring`: In what order should candidates be considered (first one to suffice is selected): by activation, frequency, match, economy, collostruction, all.

`frequency`: If frequency plays a role, should it do so absolutely or relatively (i.e. frequency as argument, or role or index marker)

`activationImpact`: if candidateOrdering=='all', how should (rescaled) activation be weighed with respect to match? Activation is function of frequency and recency. Impact==1: equally, impact below 1: impact times less important, impact above 1: impact times more important.

`collostructionImpact`: If candidateOrdering=='all', how should (rescaled) collostruction frequency be weighed with respect to match? Also used by VERBMORPHOLOGY

`semanticWeightImpact`: If candidateOrdering=='all', how should semantic weight be weighed with respect to match (given Grice: do not say more than necessary)

`economyImpact`: If candidateOrdering=='all', how should economy be weighed with respect to match (given Grice: do not say more than necessary)

`recencyDamper`: decreases activation of most recent items [RESCALE(jitter(log((frequency+1)/(recency+1+recencyImpact)
factor=activationNoise))]

`activationNoise`: noise factor that is added to activation values of items [RESCALE(jitter(log((frequency+1)/(recency+1+r
factor=activationNoise))]

`functionBlocking`: Should frequent usage for some function (argument, role marker, index marker) inhibit other functions? (only applies if frequency==relative). And: should reference to certain person values block others?

`wordOrder`: Should agents try to use word-order generalizations to mark/determine roles?

`topicCopy`: Should a (pronominal) copy of a reestablished topic be put adjacent to the verb (a la Givon; only applies if topicFirst has been derived)?

The following set of parameters applies both to the process of language change

`semUpdateAge`: At which point (relative to their death age) do agents update their lexical representations? Should be lower than procreationAge for cultural evolution to apply

`erosion`: Should forms erode?

`erosionMax`: How short may form representations become in number of characters?

`formBlocking`: Should agent refrain from reducing forms if this leads to ambiguity?

`desemanticization`: Should forms desemanticize?

`desemanticizationCeiling`: proportion of utterances in which an item occurs at which it desemanticizes maximally (.3?)

`desemanticizationPower`: Development of thresholds for subsequent dimensions to be removed. 1 for linear development. Best between 1 and 2? The lower, the more difficult to desemanticize, as the threshold develops linearly to the same target (desemanticizationCeiling)

`minimalSpecification`: minimum number of dimensions along which referential items have to be specified (in the presence of other candidate expressions for same person). If null, words will be replaced once meaningless

`verbalRoleMarker`: Can verb markers be distinctive for role (within person)? Cf. Bhat...

`semUpdateThreshold`: proportion of number of utterances in which a construction has to occur before it is fused/lexicalized

The following parameter is for data management:

`saveAll`: Should usageHistory be stored in graveyard?

References

Ariel, M. (1999). The development of person agreement markers: from pronouns to higher accessibility markers. In M. Barlow & S. Kemmer (Eds.), Usage based models of language (pp. 197-260). Stanford: CSLI.

Aristar, A.R. (1997). Marking and hierarchy. Types and the grammaticalization of case markers. Studies in Language, 21 (2), 313-368.

Bhat, D.N.S. (2004), Pronouns. Oxford [etc.]: Oxford University Press.

Croft, W. (2000). Explaining language change: An evolutionary approach. Harlow etc.: Longman.

Dahl, O. (2000). Egophoricity in discourse and syntax. Functions of Language, 7 (1), 37-77.

Deacon, T. (1997). The symbolic species. London: Penquin.

Dowty, D. (1991). Thematic proto-roles and argument selection. Language 67 (3), 547-619.

Du Bois, J.W. (1987), "The discourse basis of ergativity". Language 63 (4): 805-855

Gaerdenfors, P. (2000). Conceptual spaces: The geometry of thought. Cambridge, MA: MIT.

Givon, T. (1995). Functionalism and grammar. Amsterdam/Philadelphia: John Benjamins.

Grice, H. P. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), Syntax and semantics: Speech acts (Vol. 3, pp. 41-58). New York: Academic Press.

Heine, B. & Kuteva, T. (2007). The genesis of grammar. a reconstruction. Oxford: Oxford University Press.

Hopper, P. J. (1987). Emergent grammar. In Proceedings of BLS (Vol. 13, pp. 139-157).

Jackendoff, R. (2002). Foundations of language: Brain, meaning, grammar, evolution. Oxford: Oxford University Press.

Levelt, W.J.M. (1989). Speaking. From intention to articulation. Cambridge, MA: MIT Press.

Smith, K. & Kirby, S. (2008). Cultural evolution: Implications for understanding the human language faculty and its evolution. Phil. Trans. R. Soc. B, 363, 3591-3603.

Steels, L. (1997). Constructing and sharing perceptual distinctions. Machine Learning, ECML-97, 4-13.

Tomasello, M. (2003). Constructing a language: A usage-based theory of language acquisition. Cambridge, MA: Harvard University Press.

Van Valin, R. (1999). Generalized semantic roles and the syntax-semantics interface. In F. Corblin, C. Dobrovie-Sorin, & J.-M. Marandin (Eds.), Empirical issues in formal syntax and semantics 2 (pp. 373-389). The Hague: Thesus.

Yang, Ch. (2016), The price of linguistic productivity. Cambridge, MA: MIT Press.

Examples

```
length(world)
head(world, 10)
```

Index

∗Topic **argument marking**
  MoLE-package, 3
∗Topic **datasets**
  world, 55
∗Topic **language change**
  MoLE-package, 3
∗Topic **language evolution**
  MoLE-package, 3
∗Topic **misc**
  ACTOR, 7
  AGENTFIRST, 8
  ALLNAS, 9
  ANALYZE, 10
  CANDIDATESCORE, 11
  CHECKSUCCESS, 12
  DECOMPOSE, 13
  DIE, 14
  EROSION, 15
  FIRSTINFIRSTOUT, 16
  FIRSTSPEAKER, 17
  FMATCH, 18
  FORMS, 19
  FOUND, 20
  FREQUPDATE, 20
  FUSE, 21
  GENERALIZE, 23
  GROUP, 24
  INTERPRET, 25
  INTERPRET.INT, 26
  MAX, 27
  NOUNDESEMANTICIZATION, 28
  NOUNMORPHOLOGY, 29
  NOUNS, 30
  PERSONUPDATE, 31
  PREPARE, 32
  PROCREATE, 33
  PRODUCE, 34
  PROPOSITION, 35
  PROTOINTERPRETATION, 36
  REDUCE, 37
  REFCHECK, 38
  RESCALE, 39
  RUN, 40
  SEMUPDATE, 41
  SUCCESS, 43
  SUMMARY, 44
  TALK, 45
  TOPICCOPY, 46
  TOPICFIRST, 47
  TURN, 48
  TYPEMATCH, 49
  VERBFINAL, 50
  VERBMORPHOLOGY, 51
  VERBS, 52
  VMATCH, 52
  WORDORDER, 53
∗Topic **simulation**
  MoLE-package, 3

ACTOR, 7
AGENTFIRST, 8
ALLNAS, 9
ANALYZE, 10

CANDIDATESCORE, 11
CHECKSUCCESS, 12

DECOMPOSE, 13
DIE, 14

EROSION, 15

FIRSTINFIRSTOUT, 16
FIRSTSPEAKER, 17
FMATCH, 18
FORMS, 19
FOUND, 20
FREQUPDATE, 20
FUSE, 21

GENERALIZE, 23
GROUP, 24

INTERPRET, 25
INTERPRET.INT, 26

MAX, 27
MIN *(*MAX*)*, 27
MoLE *(*MoLE-package*)*, 3
MoLE-package, 3

NOUNDESEMANTICIZATION, 28
NOUNMORPHOLOGY, 29
NOUNS, 30

PERSONUPDATE, 31
PREPARE, 32
PROCREATE, 33
PRODUCE, 34
PROPOSITION, 35
PROTOINTERPRETATION, 36

REDUCE, 37
REFCHECK, 38
RESCALE, 39
RUN, 40

SELECTACTOR, 40
SELECTUNDERGOER *(*SELECTACTOR*)*, 40
SELECTVERB *(*SELECTACTOR*)*, 40
SEMUPDATE, 41
SITUATION, 42
SUCCESS, 43
SUMMARY, 44

TALK, 45
TOPICCOPY, 46
TOPICFIRST, 47
TURN, 48
TYPEMATCH, 49

VERBDESEMANTICIZATION *(*NOUNDESEMANTICIZATION*)*, 28
VERBFINAL, 50
VERBMORPHOLOGY, 51
VERBS, 52
VMATCH, 52

WORDORDER, 53
world, 55