Whipping the Linking Algorithm into the feature structure shape

Valeria Generalova
generalo@hhu.de

Heinrich Heine University of Dusseldorf
TreeGraSP

August 5, 2021

Overview

1 Intro

2 Translation of the Linking Algorithm
- Main features responsible for (argument) linking
- Procedural rules → static constraints

3 Linking scenarios

4 Linking in Head-marking languages

5 Conclusion

Introduction

Main aim

Formulate the Linking Algorithm in the form of features and constraints implementable with XMG language.

Background:

- Van Valin 2005, Ch. 5 – original (procedural) LA
- Osswald and Kallmeyer 2018 – formalization of RRG, “a clear distinction between declarative and procedural elements”
- Crabbé et al. 2013; Petitjean, Duchier, and Parmentier 2016 – description of XMG (eXtensible MetaGrammar)
- Kallmeyer et al. 2016 – formalization of the Actor-Undergoer Hierarchy implemented with XMG
- Generalova and Petitjean 2020 – prototype of a small RRG-based XMG project

Method

- General approach: encode the claims of the Linking Algorithm and not the logic behind it.
- Main contribution: determine what features are responsible for each step of the classical linking and where to specify them in the metagrammatical description.
- Process:
  - extract from the original guidelines what can be represented as features and discuss where in the metagrammar they must be introduced.
  - extract imperative guidelines and the context of their realization (e. g. “assign macrorole depending upon the language”) and discuss how to realize them as constraints.
- Disclaimer: main focus on Syntax⟶Semantics Linking

General architecture

Lexicon

all morphemes together with their semantic structure (frames); features percolate to higher levels of syntactic descriptions

Construction Classes

complex classes with several (`syn`, `sem`, `iface`) dimensions; describe generalizations and list varieties of constructions

Language Plugins

one variable with a lot of features describing the grammar, including the list of available constructions; intersects with CC

Most features are defined in the Lexicon and Language Plugins and then used by Construction Classes. Construction Classes introduce constraints on feature unification and disjunctions.

Main features responsible for (argument) linking

- Morphological **cases** are defined in Language Plugins
- The default **word order** is encoded in the LP; a special `class TreeShapeByWordOrder` disjoints all possible varieties and becomes imported to other Construction Classes
- Non-default word order is part of the constructions; the feature value is specified separately for this construction, another disjoint variety is imported
- **Transitivity** (valency) of the verb is encoded in the Lexicon; the value percolates to select syntactic templates
- Syntactic accusativity / ergativity is specified in the LP so that only appropriate templates are chosen; the **alignment** pattern itself is asserted in Construction Classes

Procedural rules → static constraints

Our solution repeats the Linking Algorithm itself, not the underlying reasoning!

Classic LA	→	**Static LA**
“if”-statements	→	disjunction of conjunctions
determine the voice	→	values come from Lexicon
replace with ∅	→	the label for the argument in the semantic structure does not unify with any label of a syntactic constituent
assign to other	→	invalid; all the features are assigned at once
negative constraints (if X is not Y)	→	can be handled with boolean features; usually appear as part of larger disjunctions

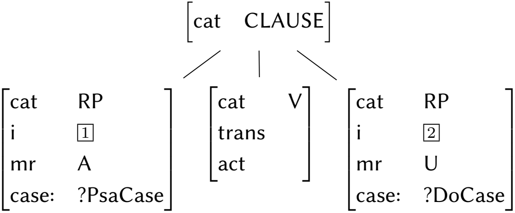
Macrorole and direct core argument status assignment

- The macrorole status of the sole argument is specified in the lexical entry for the verb (cf. co-existence of unaccusative and unergative verbs in a language)
- In Construction Classes, there are several classes for 1-argument cores that link MRs to PSAs bearing different overt cases
- The direct core argument status can be deduced from case marking; for that, cases in Language Plugins are formulated in functional terms (e.g., `psaCase`, `recipCase`, `demAgCase`, etc.)
- Correspondence between MR and case is specified in a CC of type Tree Shape, e.g. in `class TreeShapeTwoArgActive` includes `node ?RP1 [case=?PsaCase, mr=actor]`

The class TreeShape

- Roughly corresponds to the concept of diathesis (Khrakovsky 1979)
- Specifies the number of arguments, the voice, the verbal derivation
- Imports syntactic templates with the specified number of arguments and word order
- The semantic representation is built from the frame of the lexical root and additional frames of verbal morphemes (all stored in the Lexicon)

2-Argument Transitive Active



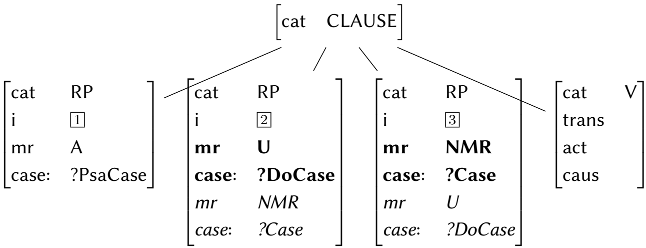
$$\begin{bmatrix}\text{cat} & \text{CLAUSE}\end{bmatrix}$$

$$\begin{bmatrix}\text{cat} & \text{RP}\\ \text{i} & \boxed{1}\\ \text{mr} & \text{A}\\ \text{case:} & \text{?PsaCase}\end{bmatrix} \quad \begin{bmatrix}\text{cat} & \text{V}\\ \text{trans} & \\ \text{act} & \end{bmatrix} \quad \begin{bmatrix}\text{cat} & \text{RP}\\ \text{i} & \boxed{2}\\ \text{mr} & \text{U}\\ \text{case:} & \text{?DoCase}\end{bmatrix}$$

$$\begin{bmatrix}\textit{event} & \\ \text{AGENT} & \boxed{1}\\ \text{PATIENT} & \boxed{2}\\ \text{Actor} & \boxed{1}\\ \text{Undergoer} & \boxed{2}\end{bmatrix}$$

- Feature `wordorder: SVO` specified for convenience
- Everything else is part of this class’ specifications
- Once an individual sentence in a language has to be parsed, the syntactic template with the correct word order is selected and the morphology is specified. All the linking is pre-defined!

3-Argument Causative Active I



$$\begin{bmatrix}\text{cat} & \text{CLAUSE}\end{bmatrix}$$

$$\begin{bmatrix}\text{cat} & \text{RP}\\ \text{i} & \boxed{1}\\ \text{mr} & \text{A}\\ \text{case:} & \text{?PsaCase}\end{bmatrix} \quad \begin{bmatrix}\text{cat} & \text{RP}\\ \text{i} & \boxed{2}\\ \textbf{mr} & \textbf{U}\\ \textbf{case:} & \textbf{?DoCase}\\ \textit{mr} & \textit{NMR}\\ \textit{case:} & \textit{?Case}\end{bmatrix} \quad \begin{bmatrix}\text{cat} & \text{RP}\\ \text{i} & \boxed{3}\\ \textbf{mr} & \textbf{NMR}\\ \textbf{case:} & \textbf{?Case}\\ \textit{mr} & \textit{U}\\ \textit{case:} & \textit{?DoCase}\end{bmatrix} \quad \begin{bmatrix}\text{cat} & \text{V}\\ \text{trans} & \\ \text{act} & \\ \text{caus} & \end{bmatrix}$$

$$\begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \begin{bmatrix}\textit{activity} & \\ \textsc{effector} & \boxed{1}\end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix}\textit{predicate} & \\ \textsc{Actor-like} & \boxed{2}\\ \textsc{Undergoer-like} & \boxed{3}\end{bmatrix}\\ \text{Actor} & \boxed{1}\\ \text{Undergoer} & \boxed{2}\text{ (bf) or }\boxed{3}\text{ (it)}\end{bmatrix}$$

3-Argument Causative Active II

- Regular parts are shared, boldface parts are disjoint with italic parts
- The difference is linking concerns only the Undergoer assignment
- No special function to account for selecting one of the disjoint options introduced: the sentence automatically matches the right one, since the word order and the cases are determined
- Values of `?Case` are specified in a further class
- Situation of complete doubling with impossible macrorole assignment (like in Yaqui) would be the third option in this disjunction?

Expanding the MG and refining the linking

- New languages:
  - new language plugins
  - intersection of existing constructions with new plugins is done automatically
  - new disjunctions in existing constructions might be needed
- New constructions:
  - can import and refine existing linking scenarios or build from scratch
  - adding new features to language plugins might be needed

Head-marking languages

- Affixes are true arguments, nouns appear in extra-core slots (Van Valin 2013)
- Usually, there are several sets of affixes, so, the identification of arguments is not complicated
- Linking of arguments is similar in dependent-marking and in head-marking languages
- Features concerning the order of the constituents have to be refined
- The open question is how to associate the noun in the ECS with the correct affix on the verb

Conclusion

Conclusions

- The existing procedural Linking Algorithm can be repeated in the shape of static feature structure
- Constructional Schemas are no different from general rules in the architecture
- Language-specific features also control the choice of one option from the whole set of possibilities
- Linking in new constructions still needs to be studied; new rules can be added easily, reusing much information from existing classes

Further studies

- Features for discourse-pragmatics
- Linking nouns to affixes in head-marking languages

Thank you!

Your feedback is very welcome:
generalo@hhu.de

These slides will be available at
valeria-generalova.com

References I

Crabbé, Benoit et al. (2013). “XMG: extensible metagrammar.” In: *Computational Linguistics* 39.3, pp. 591–629.

Generalova, Valeria and Simon Petitjean (2020). “A prototype of a metagrammar describing three-argument constructions with a morphological causative.” In: *Typology of Morphosyntactic Parameters* 3.2, pp. 29–51.

Kallmeyer, Laura et al. (2016). “Argument linking in LTAG: A constraint-based implementation with XMG.” In: *Proceedings of the 12th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 12)*, pp. 48–57.

Khrakovsky, V. S. (1979). “Diathesis.” In: *Acta linguistica academiae scientiarum Hungaricae* 29.3/4, pp. 289–308.

Osswald, Rainer and Laura Kallmeyer (2018). “Towards a formalization of Role and Reference Grammar.” In: *Applying and expanding Role and Reference Grammar (NIHIN Studies)*. Ed. by Rolf Kailuweit, Eva Staudinger, and Lisann Künkel. Freiburg: Albert-Ludwigs-Universität, Universitätsbibliothek, pp. 355–378.

Petitjean, Simon, Denys Duchier, and Yannick Parmentier (2016). “XMG 2: describing description languages.” In: *International Conference on Logical Aspects of Computational Linguistics*. Springer, pp. 255–272.

Van Valin Jr., Robert D. (2005). *Exploring the syntax-semantics interface*. Cambridge University Press.

Van Valin Jr., Robert D. (2013). “Head-marking languages and linguistic theory.” In: *Language typology and historical contingency: In honor of Johanna Nichols*. Ed. by Balthasar Bickel et al. Vol. 104. Typological Studies in Language, pp. 91–124.