Constraining MGbank: Agreement, L-Selection and Supertagging in Minimalist Grammars

John Torr
School of Informatics, University of Edinburgh
s1344326@sms.ed.ac.uk

Minimalist Grammars (MGs)

- MGs (Stabler 1997) are a computationally-oriented (MCFG equivalent) formalisation of Chomsky's (1995) Minimalist Program.
- Strongly lexicalised formalism with syntactic movement operations, in which lexical categories consist of an ordered list of features which must be checked and deleted as the derivation is constructed. E.g:

him :: d -case
helps :: d= +CASE v

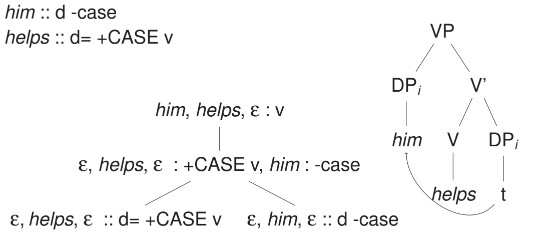

Figure: An MG Derivation tree for the VP *him, helps* (left); and its corresponding Xbar phrase structure tree (right). The object moves to check its case feature. The verb-object ordering will subsequently be restored by V-to-v head movement.

- To derive the sentence *he helps him* we expand the lexicon:

he :: d -case
[trans] :: >v= =d lv
[pres] :: lv= +CASE t
[decl] :: t= c

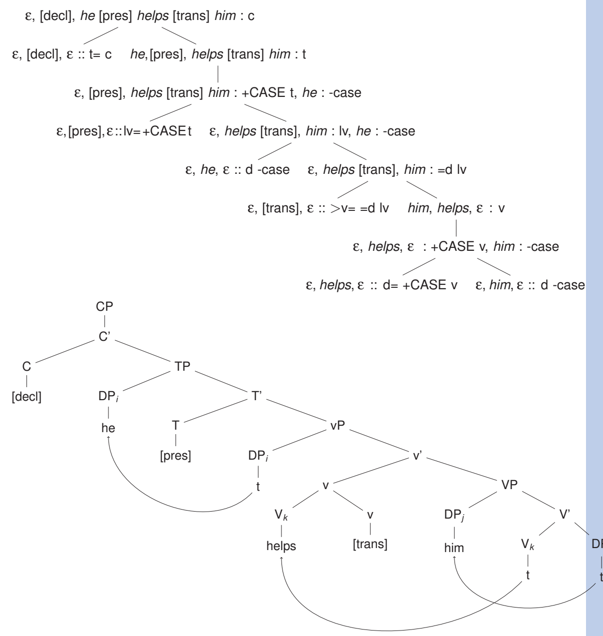

Figure: A derivation tree (top) and Xbar tree (bottom) for the sentence *he helps him*.

- MGs include the movement operations and phonetically null heads of mainstream linguistic Minimalism, both of which can be computationally costly unless carefully constrained (the worst case complexity of MG chart parsing is n^{2k+3} (Fowlie and Koller 2017)).
- A non-statistical parser was used to generate MGbank, the first wide-coverage corpus of MG trees, and we present two ways in which its search space was constrained.

Selectional Restrictions 1: Case Agreement

- In the above grammar, *he* and and *him* have exactly the same categories, so the grammar overgenerates *him helps he*.
- We could split -case into -nom/-acc, and +CASE into +NOM/+ACC.
- But this is not very elegant, as it increases *k* and in cases of syncretism (*it, this, that, you*, [det] etc) blows up the lexicon.

...Case Agreement Continued

- Instead, we subcategorise the structure building features with selectional properties (NOM, ACC, FIN, DECL, etc) and requirements (+NOM, +ACC, -FIN, +DECL etc).
- +X indicates that the feature selected or licensed must bear property X, while -X indicates that it must not bear this property.
- The following updated lexical entries will successfully block *him helps he*.

he :: d -case{NOM}
him :: d -case{ACC}
helps :: d= +CASE{+ACC} v
[pres] :: lv= +CASE{+NOM} t

Selectional Restrictions 2: Subcategorization and Subject-Verb Agreement

- Selectional requirement features can also be used to constrain local subcategorization, for example to block *Jack wants that she help(s)*.
- However, cases of long-distance subcategorization also exist: *Jack* **demanded** *that she* **be** *there on time*.
- Subject-Verb agreement is also non-local in Minimalist analyses, being mediated by the null T head.
- To capture such phenomena, we introduce selectional variables, *x, y, z* etc, which allow selectional properties and requirements to be percolated up the tree. E.g.*helps* will bear a +3SG feature on its v selectee which will percolate (via vP) up to the +CASE licensor of T where it will enforce the non-local agreement.

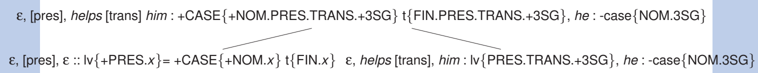

Figure: Merge of T with vP with percolation of selectional properties and requirements.

- Syncretism easily handled by allowing features to bear multiple properties from the same paradigm, e.g. *it* :: d -case{NOM.ACC}

Supertagging with MGs

- Supertagging first introduced by Bangalore and Joshi (1999) for LTAG, but has since proven most effective at making CCG parsing highly efficient.
- Applies Markovian part-of-speech tagging techniques to strongly lexicalised formalisms, thereby reducing the parser's search space.
- Although MGs are strongly lexicalised, they contain null heads, which is problematic because existing supertaggers can only tag what they can see.
- We present an algorithm for extracting a set of complex LTAG-like categories from a corpus of derivation trees, which anchors null heads to overt ones.

for each derivation tree τ:
  for each null head η *in* τ:
    if η *is a proform:*
      linkWithGovernor(η);
    else:
      linkWithHeadOfComplement(η);
  groupLinksIntoSupertags()

- The resulting supertags are composed of one overt category and zero or more null categories. The following is the supertag for *helps* from the derivation tree oppositie (subscripts indicate obligatory checking relations):

[decl] :: t=^{1} c
[pres] :: lv=^{2} +CASE t^{1}
[trans] :: v=^{3} =d lv^{2}
helps :: d= +CASE v^{3}

	rei	ab	rei-A'	ab-A'	ov	ccg
|tags|	3087	2087	1883	1181	717	342
1-best	79.1	81.1	83.0	84.2	88.0	92.4
2-best	88.4	90.2	91.1	91.9	95.3	97.1
3-best	91.6	93.5	94.1	94.8	97.1	98.3
10-best	96.4	97.4	97.9	98.2	99.2	99.5
25-best	97.6	98.5	98.9	99.1	99.7	99.7
40-best	98.0	98.7	99.0	99.4	99.8	99.8

Table: Preliminary results of an LSTM MG supertagger, trained on 13,000 trees.