The General Linearisation Model
– GLM –

The Use of Dependency Structures in Word Order Determination

Ciprian-Virgil Gerstenberger

Computational Linguistics, University of Saarland, Saarbrücken
Meaning Text Theory (1)

<table>
<thead>
<tr>
<th>representations</th>
<th>rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>{SemR}</td>
<td>semantic component</td>
</tr>
<tr>
<td>{DSyntR}</td>
<td>deep-syntactic component</td>
</tr>
<tr>
<td>{SSyntR}</td>
<td>surface-syntactic component</td>
</tr>
<tr>
<td>{DMorphR}</td>
<td>deep-morphological component</td>
</tr>
<tr>
<td>{SMorphR}</td>
<td>surface-morphological component</td>
</tr>
<tr>
<td>{DPhonR}</td>
<td>deep-phonetic component (phonology)</td>
</tr>
<tr>
<td>{SPhonR}</td>
<td></td>
</tr>
</tbody>
</table>
Meaning Text Theory (2)

- multistratal theory
- representation at all levels are multi-faceted: syntactic structures, anaphoric structures, prosodic structures, communicative structures
- the correspondence between two adjacent levels are made by rules
- rules are not transformational but interpretive
- rules are intended to apply in either direction

Example:

\[ R_n > R_{n+1} | C \]

where \( R_n \) is some structural feature of level \( n \), \( R_{n+1} \) is the corresponding feature at next level, and \( C \) is a set of conditions which control the application of the rule.
Meaning Text Theory (3)

Natural Language Generation: from meaning to text

1. $SemR \rightarrow SyntR$: mapping a net into a tree

2. $SyntR \rightarrow MorphR$: mapping a tree into a unary tree, i.e., a sequence of words

⇒ Step 2 = linearisation

Note: no matter how the level of the sequence of words is called – Deep Morphological (Mel’čuk) or Topological Level (Kahane/Gerdes) –, the same mapping occurs in all frameworks!
Linearisation (1)

**Input structure:** a dependency tree (or SSyntR in sense of MTT)

```
<table>
<thead>
<tr>
<th></th>
<th>sprt</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>sub</td>
<td>obj</td>
</tr>
<tr>
<td>hel</td>
<td>paris</td>
<td>apf</td>
</tr>
<tr>
<td>der</td>
<td>seh</td>
<td>een</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rot</td>
</tr>
</tbody>
</table>
```

**Output:** grammatically correct strings of words

*Paris gibt der sehr schönen Helena einen roten Apfel.*
*Einen roten Apfel gibt Paris der sehr schönen Helena.*
Linearisation (2)

(MTT) Assumptions:

- the dependency structure does not store word order information

Really?
- Maria, Peter und Paul gingen ins Kino.

- an SSyntR stores all linguistic material that has to be linearised

Really?
- Maria – eine Kinogängerin aus Leidenschaft – ging gestern ins Theater.
- Maria, eine Kinogängerin aus Leidenschaft, ging gestern ins Theater.

What about punctuation at this level of representation?
Linearisation (3)

Goals:

- to get all grammatically correct sequences

- to get only grammatically correct sequences

- to use knowledge about the information structure of a sentence to be able to choose the best candidates for a given context

- to choose the sequence which fits best the communication situation in a given context (incorporating the linearisation task into a greater NLG system, using discourse history, etc.)
Mapping dependency trees to word sequences (1)

- relating the nodes of a dependency tree by a **linear order relation** (LO-relating the nodes)
- two-dimensionality of a tree → three kinds of linearisation rules:

1. **vertical rules**: LO-relating a node to its children

   ![Diagram of vertical rules]

   

2. **horizontal rules**: LO-relating a node to its siblings

   ![Diagram of horizontal rules]

Examples of LO-rules in MATE format:

<table>
<thead>
<tr>
<th>det-V rule</th>
<th>det-H rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL: ( ?X { c:?r-&gt;?Y } )</td>
<td>GL: ( ?X { c:?p-&gt;?Y \ c:?q-&gt;?Z } )</td>
</tr>
<tr>
<td>GR: ( ?Ydm { b-&gt; ?Xdm } )</td>
<td>GR: ( ?Ydm { b-&gt; ?Zdm } )</td>
</tr>
<tr>
<td>CD: ( ?r = det; )</td>
<td>CD: ( ?p = det; )</td>
</tr>
<tr>
<td>CR: ( ?X &lt;=&gt; ?Xdm; )</td>
<td>CR: ( ?Y &lt;=&gt; ?Ydm; )</td>
</tr>
<tr>
<td>( ?Y &lt;=&gt; ?Ydm; )</td>
<td>( ?Z &lt;=&gt; ?Zdm; )</td>
</tr>
</tbody>
</table>

*Generalized Dependency Theory, 2004, Ciprian Gerstenberger*
3. diagonal rules: LO-relating two nodes which are not immediately related (child/parent or siblings)

– Should all nodes be LO-related pairwise?
– In many cases, this is unnecessary!

Idea: grouping nodes and LO-relating these groups!

⇒ grouping rules

– How to group nodes?
– Use linguistic knowledge!
Analysis versus Generation

- analysis is description, generation is prescription

- analysis assumes sophisticated structures for language description, the result of generation is just a chain of words

- language models postulate elements (traces, empty topological fields, etc.) which are factually not existent in the string of words generation is supposed to output

Claims:

- such models as Kahane/Gerdes complicate the task of linearisation unnecessarily

- linearisation as a step of NLG needs a model without phantoms
The General Linearisation Model – GLM (1)

Features

GML is a simple linearisation model comprising

1. a sole type of entities:
   (a) Linear Order Part (LOP)

2. two different relations holding between entities:
   (a) Part-Of Relation
   (b) Linear Order Relation

3. four types of rules to map a dependency tree into a Linear Order Part:
   (a) LOP forming rules
   (b) vertical rules
   (c) horizontal rules
   (d) diagonal rules
The General Linearisation Model – GLM (2)

Definitions

- **Linear Order Part**: A Linear Order Part (LOP) is a language item which is supposed to be linearised as a continuous part of a sentence.

- **Part-Of Relation**: A Part-Of-Relation (POR) holding between two LOPs $\lambda_1$ and $\lambda_2$ ($\lambda_1 \sqsubset \lambda_2$) states that $\lambda_1$ is part of $\lambda_2$. The POR is reflexive, anti-symmetric, and transitive.

- **Linear Order Relation**: A Linear Order Relation (LOR) holding between two different LOPs $\lambda_1$ and $\lambda_2$ ($\lambda_1 \prec \lambda_2$) states that $\lambda_1$ precedes $\lambda_2$. The LPR is irreflexive, asymmetric, and transitive.
The General Linearisation Model – GLM (3)

Two different LOPs can either PO-relate or LO-relate but not both; they can not overlap.

Let $\lambda_1$, $\lambda_2$, and $\lambda_3$ be LOPs:

1. if $\lambda_1 \sqsubset \lambda_2$, then $\lambda_1 \not\subset \lambda_2$
2. if $\lambda_1 \sqsubset \lambda_2$, then $\lambda_2 \not\subset \lambda_1$
3. if $\lambda_1 \prec \lambda_2$, then $\lambda_1 \not\subset \lambda_2$
4. if $\lambda_1 \prec \lambda_2$, then $\lambda_2 \not\subset \lambda_1$
5. if $\lambda_2 \sqsubset \lambda_1$ and $\lambda_2 \sqsubset \lambda_3$, then either $\lambda_1 \sqsubset \lambda_3$ or $\lambda_3 \sqsubset \lambda_1$

\[ \begin{array}{c}
\lambda_1 \prec \lambda_2 \\
{\text{[das Buch]}_{\lambda_1}} \quad {\text{[auf dem Tisch]}_{\lambda_2}}
\end{array} \]

\[ \begin{array}{c}
\lambda_1 \sqsubset \lambda_2 \\
{\text{[[das Buch]}_{\lambda_3} \quad {\text{[auf dem Tisch]}_{\lambda_1}]}_{\lambda_2}}
\end{array} \]
Corrolaries:

- the smallest LOP is a single word
- there is no empty LOP
- an LOP can consist of different (smaller) LOPs
- given a dependency tree as input structure, the biggest LOP is the one consisting of all nodes of the tree

- let $\lambda_1$, $\lambda_2$, $\lambda_3$, and $\lambda_4$ be LOPs, and $\lambda_1 \prec \lambda_2$, then
  1. if $\lambda_1 \sqsubset \lambda_3$ and $\lambda_2 \not\sqsubset \lambda_3$ then $\lambda_3 \prec \lambda_2$
  2. if $\lambda_2 \sqsubset \lambda_4$ and $\lambda_1 \not\sqsubset \lambda_4$ then $\lambda_1 \prec \lambda_4$
  3. if $\lambda_1 \sqsubset \lambda_3$, $\lambda_2 \sqsubset \lambda_4$, $\lambda_3 \not\sqsubset \lambda_4$, and $\lambda_4 \not\sqsubset \lambda_3$, then $\lambda_3 \prec \lambda_4$

- let $\lambda_1$, $\lambda_2$, $\lambda_3$, and $\lambda_4$ be LOPs, and $\lambda_3 \prec \lambda_4$, then
  1. if $\lambda_1 \sqsubset \lambda_3$ then $\lambda_1 \prec \lambda_4$
  2. if $\lambda_2 \sqsubset \lambda_4$ then $\lambda_3 \prec \lambda_2$
  3. if $\lambda_1 \sqsubset \lambda_3$ and $\lambda_2 \sqsubset \lambda_4$ then $\lambda_1 \prec \lambda_2$
The General Linearisation Model – GLM (5)

Input structure – Dependency Tree

Intermediate structures – LOPs

Output structures – Sentences

Paris gibt der sehr schönen Helena einen roten Apfel.
Paris gibt einen roten Apfel der sehr schönen Helena.
Difficulty of writing linearisation grammars

⇒ getting only grammatically correct sequences

• not a technical problem

• a problem of interpersonal agreement of what is grammatically correct and what not (even among native speakers)

Example: focus sensitive particles

1. Nur **Maria** hat das Buch gelesen.
2. **Maria** nur hat das Buch gelesen.
3. **Maria** hat nur das Buch gelesen.
4. **Maria** hat das Buch nur gelesen.
5. **Maria** hat das Buch gelesen nur.
Grammaticality Test: 11 native speakers of German

1. Nur Maria$_{Focus}$ hat das Buch gelesen.
   
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
</tr>
</tbody>
</table>

2. Maria$_{Focus}$ nur hat das Buch gelesen.
   
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>6</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

3. Maria$_{Focus}$ hat nur das Buch gelesen.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

4. Maria$_{Focus}$ hat das Buch nur gelesen.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

5. Maria$_{Focus}$ hat das Buch gelesen nur.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bewertungsschlüssel:

- A ⇒ grammatisch falsch
- B ⇒ grammatisch richtig, aber an der Grenze
- C ⇒ grammatisch richtig, besser als B
- D ⇒ grammatisch richtig, besser als B und als C
Conclusions

- GLM is conceived especially for generation (for analysis would not be too useful)
- the conception of the model is based on the surface of a sentence
- no use of empty structures
- abstract enough to be used as a tool for coping with linearisation in every language
- abstract enough to cope with different syntactic structures in a uniform way (an LOP can be a word, a phrase, a mittelfeld, a subclause, etc.)
- simple model: a sole type of entities (LOP), two different relations holding between them (POR and LOR), and four different mapping rules (LOP forming, vertical, horizontal, and diagonal rules)