BME-TUW at SR’20
Lexical grammar induction for surface realization

Gábor Recski¹, Ádám Kovács¹,², Kinga Gémes¹,², Judit Ács², András Kornai³

¹TU Wien
firstname.lastname@tuwien.ac.at

²Dept. of Automation and Applied Informatics, Budapest U of Technology
lastname.firstname@aut.bme.hu

³SZTAKI Institute of Computer Science
andras@kornai.com

Third Workshop on Multilingual Surface Realisation, 12/12/2020
Rule-based system for word order restoration + DL reinsertion improves the grammar-based approach in Kovács et al. (2019) still inferior to DL systems, but opens up new possibilities.
Summary

- Rule-based system for word order restoration + DL reinflection

Improves the grammar-based approach in Kovács et al. (2019) but opens up new possibilities.
Summary

- Rule-based system for word order restoration + DL reinflection
- Improves the grammar-based approach in Kovács et al. (2019)
Summary

- Rule-based system for word order restoration + DL reinforcement
- Improves the grammar-based approach in Kovács et al. (2019)
- Still inferior to DL systems, but opens up new possibilities
Interpreted Regular Tree Grammars

Interpreted Regular Tree Grammars (IRTGs, Köller and Kuhlmann, 2011) encode the correspondence between operations over a string algebra and an s-graph algebra (Courcelle and Engelfriet, 2012; Köller, 2015).

VERB → _nsubj (VERB, NOUN)

\[
\text{Read: constructing the subgraph } \text{VERB} \rightarrow _nsubj \text{NOUN} \text{ corresponds to concatenation in the order NOUN VERB.}
\]
Interpreted Regular Tree Grammars (IRTGs, Koller and Kuhlmann, 2011) encode the correspondence between operations over a string algebra and an s-graph algebra (Courcelle and Engelfriet, 2012; Koller, 2015).
Interpreted Regular Tree Grammars (IRTGs, Koller and Kuhlmann, 2011) encode the correspondence between operations over a string algebra and an s-graph algebra (Courcelle and Engelfriet, 2012; Koller, 2015).

VERB -> _nsubj(VERB, NOUN)
[string] *(?2, ?1)
[ud] f_depl(merge(merge(?1,"(r<root> :nsubj d1<dep1>)"),r_depl(?2)))

Read: constructing the subgraph VERB \xrightarrow{nsubj} NOUN corresponds to concatenation in the order NOUN VERB.
Generating subgraphs

For a head word with N dependents, we enumerate \( \sim 3^N \) subgraphs.
Generating subgraphs

\[ \text{He/PRON} \xrightleftharpoons{\text{nsubj}} \text{enjoy/VERB} \xrightarrow{\text{obj}} \text{it/PRON}. \]
Generating subgraphs

\[ He/PRON \xleftarrow{nsubj} \text{enjoy/VERB} \xrightarrow{obj} it/PRON. \]

For a head word with \( N \) dependents, we enumerate \( \sim 3^N \) subgraphs.
## Model statistics

| Lang | $N_{patt}$ | $D_{max}$ | $|V|$ | $D_{words}$ | $N_{tok}$ |
|------|------------|-----------|-------|-------------|----------|
| ar   | 8.6M       | 4.8       | 14K   | 36.9        | 224K     |
| en   | 29.8M      | 5.0       | 25K   | 17.6        | 352K     |
| es   | 50.2M      | 5.5       | 48K   | 29.0        | 827K     |
| fr   | 37.1M      | 5.7       | 34K   | 24.6        | 429K     |
| hi   | 17.2M      | 5.5       | 15K   | 21.1        | 281K     |
| id   | 7.0M       | 5.2       | 19K   | 21.8        | 98K      |
| ja   | 14.5M      | 5.6       | 24K   | 22.5        | 160K     |
| ko   | 8.6M       | 3.9       | 119K  | 12.9        | 353K     |
| pt   | 27.2M      | 5.2       | 32K   | 25.7        | 462K     |
| ru   | 41.6M      | 4.7       | 51K   | 18.0        | 946K     |
| zh   | 14.8M      | 6.8       | 20K   | 24.7        | 99K      |
Generating subgraphs

For each UD graph, we generate a separate IRTG.

For each subgraph, we add the most frequent rule.

Identical rule weights → grammars favor shorter derivations with more specific rules.
Generating subgraphs

- For each UD graph, we generate a separate IRTG
Generating subgraphs

- For each UD graph, we generate a separate IRTG
- For each subgraph, we add the most frequent rule
Generating subgraphs

- For each UD graph, we generate a separate IRTG
- For each subgraph, we add the most frequent rule
- Identical rule weights $\rightarrow$ grammars favor shorter derivations with more specific rules
Generating subgraphs

really/ADV enjoyed/VERB I/PRON reading/VERB it/PRON
Generating subgraphs

I really enjoyed reading it.
Hierarchical SR

Cut UD graphs along edges between clauses: acl, advcl, ccomp, xcomp, conj.
Hierarchical SR

Cut UD graphs along edges between clauses: acl, advcl, ccomp, xcomp, conj.
Hierarchical SR

Cut UD graphs along edges between clauses: acl, advcl, ccomp, xcomp, conj.

I really enjoyed reading it.

Diagram:

- **ROOT**
  - **ADVMOD**
    - I
  - **NSUBJ**
    - really
  - **VERB**
    - enjoyed
  - **OBJ**
    - reading
  - **PRON**
    - it
Hierarchical SR

Cut UD graphs along edges between clauses: acl, advcl, ccomp, xcomp, conj.

Really/ADV enjoyed/VERB I/PRON reading/VERB it/PRON

Recski, Kovács, Gémes, Ács, Kornai
BME-TUW at SR'20
SR'20 Workshop, 12/12/2020
Hierarchical SR

Cut UD graphs along edges between clauses: acl, advcl, ccomp, xcomp, conj.

I really enjoyed reading it.

Recski, Kovács, Gémes, Ács, Kornai
Hierarchical SR

In a sample of 500 English sentences, we run 1794 iterations of the core method, and observe recursion depths up to 6.
In a sample of 500 English sentences, we run 1794 iterations of the core method, and observe recursion depths up to 6.

*Perhaps had we not gone into this restaurant believing Zahav was going to be golden as its name suggests (and as the many golden reviews seem to attest), we would have enjoyed a decent little expensive experience.*
In a sample of 500 English sentences, we run 1794 iterations of the core method, and observe recursion depths up to 6.

Perhaps had we not gone into this restaurant believing Zahav was going to be golden as its name suggests (and as the many golden reviews seem to attest), we would have enjoyed a decent little expensive experience.

gone $\xrightarrow{advcl}$ believing $\xrightarrow{ccomp}$ going $\xrightarrow{xcomp}$ golden $\xrightarrow{advcl}$ suggests $\xrightarrow{conj}$ seem $\xrightarrow{advcl}$ attest
## Evaluation

<table>
<thead>
<tr>
<th>Team</th>
<th>Meaning</th>
<th></th>
<th></th>
<th>Readability</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ewt Ave.</td>
<td>Ave. z</td>
<td>wiki Ave.</td>
<td>Ave. z</td>
<td>ewt Ave.</td>
</tr>
<tr>
<td>HUMAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>75.7</td>
</tr>
<tr>
<td>IMS</td>
<td>92.7</td>
<td>0.534</td>
<td>92.3</td>
<td>0.475</td>
<td>73.9</td>
</tr>
<tr>
<td>ADAPT</td>
<td>90.7</td>
<td>0.476</td>
<td>91.6</td>
<td>0.441</td>
<td>72.5</td>
</tr>
<tr>
<td>Concordia</td>
<td>87.0</td>
<td>0.332</td>
<td>88.7</td>
<td>0.275</td>
<td>70.2</td>
</tr>
<tr>
<td>BME 2020</td>
<td>79.3</td>
<td>0.086</td>
<td>81.8</td>
<td>-0.050</td>
<td>58.2</td>
</tr>
<tr>
<td>BME 2019</td>
<td>77.4</td>
<td>0.024</td>
<td>82.4</td>
<td>-0.074</td>
<td>56.7</td>
</tr>
</tbody>
</table>
## Evaluation

<table>
<thead>
<tr>
<th>Data</th>
<th>Meaning</th>
<th>Readability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BME 2020</td>
<td>BME 2019</td>
</tr>
<tr>
<td>Ave.</td>
<td>Ave. z</td>
<td>Ave.</td>
</tr>
<tr>
<td>en_ewt</td>
<td>79.3</td>
<td>0.086</td>
</tr>
<tr>
<td>en_wiki</td>
<td>81.8</td>
<td>-0.050</td>
</tr>
<tr>
<td>ru_syn</td>
<td>81.2</td>
<td>-0.166</td>
</tr>
<tr>
<td>ru_wiki</td>
<td>78.2</td>
<td>-0.079</td>
</tr>
<tr>
<td>es_ancora</td>
<td>70.2</td>
<td>-0.276</td>
</tr>
<tr>
<td>es_wiki</td>
<td>69.8</td>
<td>-0.170</td>
</tr>
</tbody>
</table>
Plans

Use 'unlimited' silver standard UD data

Learn rule weights

Qualitative analysis of performance gap

Recski, Kovács, Gémes, Ács, Kornai

BME-TUW at SR’20

SR’20 Workshop, 12/12/2020
Plans

- Use ‘unlimited’ silver standard UD data
Plans

- Use ‘unlimited’ silver standard UD data
- Learn rule weights
Plans

- Use ‘unlimited’ silver standard UD data
- Learn rule weights
- Qualitative analysis of performance gap
All components of our system are free and open source:

<table>
<thead>
<tr>
<th>Component</th>
<th>URL</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word order restoration</td>
<td>github.com/adaamko/surface_realization</td>
<td>MIT</td>
</tr>
<tr>
<td>Reinflection</td>
<td>github.com/juditacs/deep-morphology</td>
<td>MIT</td>
</tr>
<tr>
<td>IRTG generation</td>
<td>github.com/recski/tuw-nlp</td>
<td>MIT</td>
</tr>
<tr>
<td>IRTG parsing</td>
<td>github.com/coli-saar/alto</td>
<td>Apache 2.0</td>
</tr>
</tbody>
</table>
Courcelle, Bruno and Joost Engelfriet (2012). *Graph structure and monadic second-order logic.* Cambridge University Press.

