Meanwhile, September housing starts, due Wednesday, are thought to have inched upward.

Goal: Develop a shared task system taking advantage of prior work on surface realization with OpenCCG (Espinosa et al., 2008; White and Rajkumar, 2009; inter alia).

This Year’s Strategy: Try straightforward conversion from deep shared task inputs to “native” OpenCCG inputs, together with techniques for enhanced robustness (see ENLG poster on glue rules for robust chart realization).

Lesson: Input conversion was more difficult than anticipated; these difficulties led us to develop a relation tagger to map shared task inputs to native inputs.

Results: Despite these efforts, frequent head inversions and other more complex structural divergences led to a low percentage of grammatically complete realizations, yielding disappointing results.

Subsequent Analysis: CCG’s principled approach to relativization and extraction was the source of a substantial number of input incompatibilities; such divergences reveal deficiencies in the shared task inputs.

Future Work: We plan to examine whether more comprehensive use of machine learning in input conversion or inducing grammars that are more directly compatible with shared task inputs can achieve high quality results.

Background

OpenCCG is a parsing/realization library for Combinatory Categorial Grammar (Steedman, 2000); a unification-based categorial grammar formalism.

The dataset chosen as the restriction to dependency trees in the surface task is incompatible with CCG’s treatment of coordination, relativization and control (see paper).

OpenCCG implements a chart realization algorithm in the tradition of Kay (1996); such approaches can potentially deliver very high quality, but achieving broad coverage is a challenge, as is robustness to any deviations in the expected input.

OpenCCG’s “native” inputs are derived from gold standard derivations in the CCGbank (Hockenmaier and Steedman, 2007), as is the extracted English grammar; see box at right for details of shared task input conversion.

Realization makes use of an adaptive hypertagging strategy and an averaged perceptron scoring model incorporating n-grams and syntactic features.

For robustness, fragments are assembled if no grammatically complete realization can be found.

Results

Development set scores for all realizations (OSU.1) and grammatically complete realizations only (OSU.2) for the shared task inputs and using native inputs are given in the table below.

As the table shows, the percentage of grammatically complete realizations for the converted shared task inputs is well below the percentage using native inputs, with a corresponding drop in BLEU scores (note that even native scores and completeness are somewhat lower than expected, presumably due to a still mysterious drop in hypertagger performance).

<table>
<thead>
<tr>
<th>System</th>
<th>Shared Task</th>
<th>Native</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU 5-best Coverage</td>
<td>BLEU 5-best Coverage</td>
</tr>
<tr>
<td>OSU.1 (all)</td>
<td>0.1496 / 0.2851 / 96%</td>
<td>0.1508 / 0.3174 / 96%</td>
</tr>
<tr>
<td>OSU.2 (complete)</td>
<td>0.6564 / 0.3871 / 19%</td>
<td>0.8341 / 0.5413 / 76%</td>
</tr>
</tbody>
</table>

Acknowledgements

This work was supported in part by NSF grant HS-0812297 and by an allocation of computing time from the Ohio Supercomputer Center.