**Glue Rules for Robust Chart Realization**

Michael White
Department of Linguistics, The Ohio State University

**SUMMARY**

- **Goal**: Show how glue rules can be used to increase the robustness of statistical chart realization, in a manner inspired by dependency realization.
- **Comparison**: In contrast to the use of glue rules in MT—like previous work with XLE on improving robustness with hand-crafted grammars—they are invoked here as a fallback option when no complete realization can be found.
- **Benefits**: Unlike an earlier technique of greedily assembling fragments, glue rules are well-integrated into the chart-based search, enabling n-best outputs and compatibility with disjunctive inputs.
- **Results**: Experiments with OpenCCG indicate that glue rules yield substantially improved realizations in comparison to greedy fragment assembly.

**Motivation**

- Robustness continues to be a problem for chart realizers: since Ker’s (1996) pioneering work, broad coverage realizers have been developed for BIPS6, LFG and CCG, but none come close to 100% coverage. (see paper).
- With realization shared tasks, “non-native” inputs make robustness even more of an issue.
- By contrast, recent statistical dependency approaches (Guo et al., 2018; 2010, Gal and Venkatesharya, 2009; Boloni et al., 2010) which show explicit grammatical constraints—easily achieve 100% coverage.
- Here: use MT-inspired glue rules (Chiang, 2007) as a fall-back option, emulating dependency realization in cases where no grammatically complete realization can be found.
- Implementation: Approach formalized in Combinatory Categorial Grammar (Steedman, 2000) and implemented in OpenCCG, improving upon earlier greedy fragment concatenation method; should be applicable to other grammatical frameworks as well.

**Glue Rules**

- **Basic Idea**: Concatenate grammatical constituents, avoiding spurious ambiguity.
  \[ X \rightarrow \text{frag} + \text{frag} \]
- **A Twist**: Glue rules invoked only after chart completed with no complete realization, and limited to empty cells, agenda sorted by edge coverage then model score (i.e. preference given to larger, grammatical edges).
- **Another**: Relaxes the LF chunking constraints (see White, 2006), missing elementary predication (EPs) are made optional, as are ones associated with instantiated unary rules (see paper for related work on grammar-based error detection).
- **A Third**: To allow glue rules to be applied recursively: fragments that complete an LF chunk or disjunction are marked as completed fragments (\text{frag}), so that they may be used as the glue rule as the right category (where fragments are normally disabled).
- **Emulating Dependent Realization**: Since LF chunking constraints are applied as usual, the fragment gluing phase becomes tantamount to exploring different permutations of heads and phrases headed by their dependents, much as in dependency realization, that is, since fragment edges are constructed by assembling existing edges in other order, all permutations of edges whose EPs fall within an LF chunk will eventually be tried (subject to search constraints), with preference given to the orderings with the best model scores.

**Evaluation**

- A lexico-grammar extracted from CCGBank Sections 62-21 was used to realize the LFs.
- With fragmentary realizations, glue rules were compared to the earlier greedy assembly method.
- Averaged perception models were used for scoring, with and without the dependency features (note that named entity and agreement features were not used in this work).
- An oracle model using n-gram precision score (approximating BLEU) provided a topline.

**Conclusions and Future Work**

- Glue rules enhance robustness by providing a fall-back option when no grammatically complete realization can be found.
- Unlike an earlier technique of greedily assembling fragments, glue rules enable n-best outputs, are compatible with deputitive inputs, and explore a larger search space.
- They differ from the fragment concatenation rules used in hand-crafted grammars for the XLE realizer in applying recursively, enabling the glue rules to emulate dependency realization.
- The experimental results indicate that glue rules can yield improved realizations, though a sizable gap in quality remains between grammatically complete and fragmentary realizations.
- In future work, we plan to experiment with realization ranking models incorporating richer dependency-based features, and to examine the impact of such models on shared task results.

**Acknowledgements**

This work was supported in part by NSF grant BCS-0812257. Thanks to Chris Brew for helpful discussion.