Inducing clause-combining operations for natural language generation

David M. Howcroft
Computational Linguistics & Phonetics
Saarland University
66123 Saarbrücken, Germany
howcroft@coli.uni-saarland.de

Michael White
Department of Linguistics
The Ohio State University
Columbus, OH 43210, USA
mwhite@ling.osu.edu

Abstract

Recent work in end-to-end generation has reduced the need for knowledge-engineering, but is insufficiently sensitive to discourse structure. We present a method for inducing clause-combining rules for use in a traditional natural language generation architecture to address this gap. Our algorithm is able to learn all of the clause-combining rules present in the SPaRKy restaurant corpus from exemplary input-output pairs and is currently being extended to include the induction of both lexicalization and referring expression rules. We also describe initial work applying this technique in a new domain.

1 Introduction

A traditional natural language generation (NLG) system uses a sequence of hand-crafted components to generate high-quality text (Reiter and Dale, 2000). This requires considerable amounts of expert attention, leading to recent work to reduce the amount of knowledge-engineering required at the content-planning (Duboue and McKeown, 2001; Barzilay and Lapata, 2005) and surface realization (Reiter, 2010; Rajkumar and White, 2014) levels. So far, however, the intervening sentence planning, or microplanning, has been more difficult to automate.

This has motivated a focus on learning complete end-to-end generation systems. Angeli et al. (2010) describe an end-to-end system which maps database records to surface realizations by selecting a sequence of records, choosing which fields of those records to express, and finally choosing a sequence of words to express the values of those fields. Konstas & Lapata (2013) generalize this approach but point out that handling discourse-level structures remains for future work. As such, their work relies on "pre-aggregating" propositions relating to the same entity into a single database record in order to aggregate propositions. Many aggregation operations, however, are dependent on the linguistic structure of a text and not just the propositional content that can be prespecified as input to the generator.

We present a system which fills this gap by learning clause-combining rules in the context of a traditional NLG architecture. The approach uses lexico-semantic dependency edits to capture these operations and is therefore closely related to Angrosh & Siddhartan's (2014) system for simplification which uses lexico-syntactic rewrite rules. Our system extends their approach, taking into account constraints, based on shared structure between discourse subtrees, that are crucial to the accurate application of aggregation operations.

Here we present an overview of the approach and explain how developers can use rule induction coupled with a broad-coverage parser in the development of generation systems for new domains.

2 Inducing clause-combining rules

2.1 The SPaRKy Restaurant Corpus

Walker et al. (2007) developed SPaRKy as an extension to the MATCH system (Walker et al., 2004) for restaurant recommendations. One byproduct of their study is the SPaRKy Restaurant Corpus (SRC), a collection of content plans, textplans, and surface realizations of those plans accompanied by user ratings. While the propositional content of the text plans is limited to the properties of the various restaurants, such as their food quality or their location, crafting contrastive recommendations requires the careful application of aggregation operations. Therefore this corpus is a good resource for developing novel approaches to clause-combining operations and their induction.

```
rule:
_ infer-rel [ .._.. ]
  Arg1
    G have [ ..H.. ]
      Arg0
        C A [ ..B.. ]
          ...D...
      ...L...
  Arg2
    _ be [ .._.. ]
      Arg0
        E A [ ..B.. ]
          ...F...
      Arg1
        I restaurant [ ..J.. ]
          ...K...

s.t. [equiv(node(C,A,B,D),node(E,A,B,F))]

==>

G have [ ..H.. ]
  Arg0
    C A [ ..B.. ]
      ApposRel
        I restaurant [ ..J.. ]
          ...K...
      ...D...
  ...L...
```

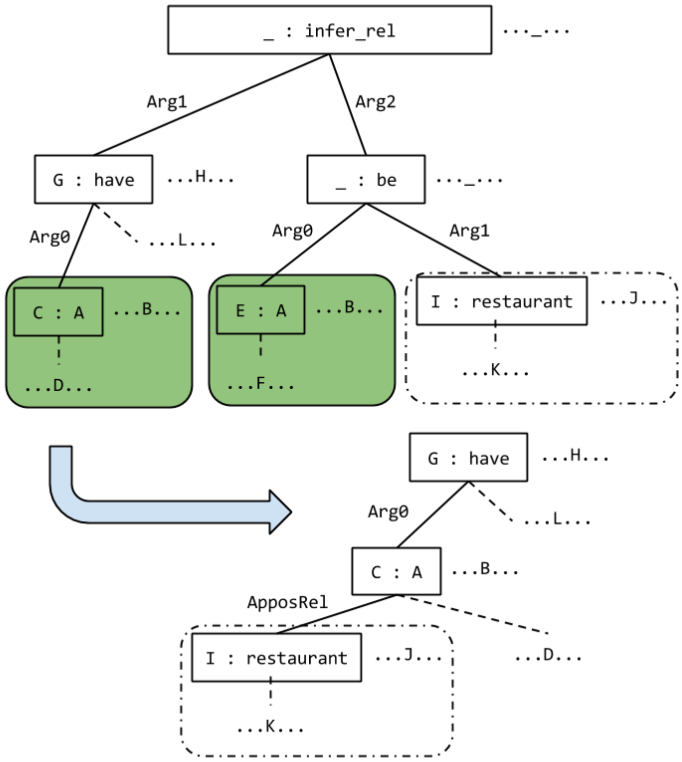

Figure 1: Example of an induced rule for apposition. The rule requires that the subjects of the `have` and `be` predicates be equivalent and, if they are, maps the description of the restaurant to an apposition in the aggregated LF.

2.2 Rule Induction Method

The input-output pairs used by our method are logical forms of the kind used in OpenCCG[1]. During development we used hand-crafted lexicalization rules based on parses of the SRC realizations using a broad-coverage grammar of English, although these rules can be easily acquired using an align-and-factor approach similar to that described below. These lexicalization rules are applied to the SRC textplans to produce input LFs for the learning algorithm. The output LFs are produced by applying a set of hand-crafted clause-combining operations to these input LFs.[2]

Input LFs are aligned to the corresponding output LFs using an alignment routine which aligns unique lexical matches first and then greedily aligns remaining nodes based on the number of parent and child nodes already aligned. Using this alignment the algorithm computes the set of differences between the input and the output in terms of insertions and deletions of nodes, edges, and attributes. The algorithm analyses these edits to determine constraints on the induced rule, specifying when it is licensed and when it is not.

3 Current and Future Work

The approach we describe works well with clean training data, so we are now evaluating its performance on more realistic data. In particular, we are extending the approach to learn mappings from textplans to lexicalized LFs so we can do away with the grammar-derived, hand-crafted rules for lexicalization. Naturally, we are also using parses from the SRC in place of outputs of hand-crafted rules to determine the full extent to which humans can be removed from the pipeline.

Furthermore, we are now beginning work on a new in-car dialogue system whose test-case is booking movie and other reservations in German. This new domain provides an opportunity to apply the approach in the development of a novel NLG system and evaluate the extent to which we have achieved our goal of reducing the need for knowledge-engineering in microplanning.

[1] `http://openccg.sf.net`

[2] These hand-crafted rules result in LFs that are equivalent to those produced by parsing the SRC realizations directly. We generate an initial training set in this fashion to demonstrate the principles of our approach on clean data.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512, Cambridge, MA, October. Association for Computational Linguistics.

Mandya Angrosh and Advaith Siddharthan. 2014. Text simplification using synchronous dependency grammars: Generalising automatically harvested rules. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 16–25, Philadelphia, Pennsylvania, U.S.A., June. Association for Computational Linguistics.

Regina Barzilay and Mirella Lapata. 2005. Collective content selection for concept-to-text generation. In *Proceedings of the HLT/EMNLP*, pages 331–338, Vancouver.

Pablo A. Duboue and Kathleen R. McKeown. 2001. Empirically estimating order constraints for content planning in generation. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 172–179, Toulouse, France, July. Association for Computational Linguistics.

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *Journal of Artificial Intelligence Research*, 48:305–346.

Rajakrishnan Rajkumar and Michael White. 2014. Better surface realization through psycholinguistics. *Language and Linguistics Compass*, 8(10):428–448.

Ehud Reiter and Robert Dale. 2000. *Building natural generation systems*. Studies in Natural Language Processing. Cambridge University Press.

Ehud Reiter. 2010. Natural language generation. In Alexander Clark, Chris Fox, and Shalom Lappin, editors, *The Handbook of Computational Linguistics and Natural Language Processing (Blackwell Handbooks in Linguistics)*, Blackwell Handbooks in Linguistics, chapter 20. Wiley-Blackwell, 1 edition.

Marilyn A. Walker, S. Whittaker, Amanda Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. 2004. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, 28(5):811–840, October.

Marilyn Walker, Amanda Stent, François Mairesse, and Rashmi Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research (JAIR)*, 30:413–456.