An Architecture for Aggregation in Text Generation

James Shaw
Dept. of Computer Science
Columbia University
New York, NY 10027, USA
shaw@cs.columbia.edu

Kathleen McKeown
Dept. of Computer Science
Columbia University
New York, NY 10027, USA
kathy@cs.columbia.edu

Published in IJCAI, Poster Session, 1997.

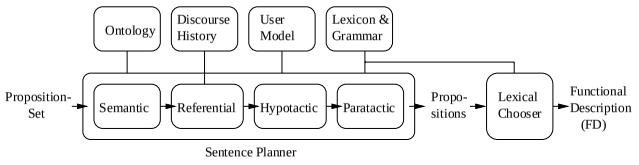

Figure 1: Architecture of Sentence Planner

1 Introduction

A traditional natural language generation system architecture consists of a content planner and a surface realizer. The content planner packages information as verb-based, clause-sized propositions, each of which is realized as a single sentence. The surface realizer maps the semantic propositions into actual sentences. In practical applications, many of these propositions share common features, such as the entity being described or discussed. If a generation system simply generates each proposition as a sentence, the output will contain many repetitive and redundant references to common features. A better approach is to detect shared entities among the adjacent propositions and combine them to remove redundancies.

To achieve such capability, we added a *sentence planner* between the content planner and surface realizer. Its main task is aggregation – the combining of semantically related propositions in order to produce concise and fluent expressions. During the aggregation process, before an operator is applied to propositions, the lexicon is consulted to make sure that the operator is applicable. This guarantees that the combined proposition can be realized as a surface string. The system also uses an ontology to generate referring expressions and generalization, both of which result in concise expressions.

2 The Architecture

The goal of the sentence planner is to transform a set of input propositions into a minimum amount of words under lexical, grammatical, and pragmatic constraints. This transformation process occurs in multiple stages as shown in Figure 1. In each stage, a set of combining operators is applied to the propositions.

3 Aggregation Operators

The following aggregation operators are applied to the propositions in the order described:

- **Semantic**: Ontological subsumption substitutes a more general concept from the domain ontology in place of the set of the children underneath the general concept (e.g., "drugs" vs. "nitroglycerin and levophed").
- **Referential**: The system selects an adequate description to refer to entities (e.g., "both arms" vs. "left arm and right arm").
- **Hypotactic**: Various modifying syntactic constructions, such as adjectives, PPs, present/past participle clauses and relative clauses, are used in place of a full sentence (e.g., "diabetic patient" vs. "the patient has diabetes").
- **Paratactic**: It takes advantage of parallel structures at the syntactic level to delete repetitive constituents and shorten expressions by using conjunction.

The sentence planner works at a semantic level and uses lexical information to guarantee that there is at least one way to express the combined propositions. The task of paraphrasing and exact word choice are carried out at the lexical chooser stage. This division allows the sentence planner to concentrate on operations which involve multiple propositions – detecting similarities and removing them. In contrast, the lexical chooser concentrates on paraphrasing and word choice, which occur after the sentence boundaries have been determined.

4 Conclusion

The aggregation operators use common grammatical constructions to produce *concise* sentences. We have built a prototype in our MAGIC system which generates coordinated multimedia presentation for care-givers in the Intensive Care Unit. Currently we are adding additional operators and scaling up the system.