

## HALogen Statistical Sentence Generator

**Irene Langkilde-Geary**

Information Sciences Institute  
University of Southern California  
ilangkil@isi.edu

**Kevin Knight**

Information Sciences Institute  
University of Southern California  
knight@isi.edu

### 1 Introduction

HALogen is a broad-coverage, general-purpose, natural language sentence generation system that combines symbolic rules with statistical models of language. It can be used to do generation in the context of larger tasks such as translation, summarization, and human-computer dialogue. Its coverage of English syntax and correctness of output has been empirically verified using a test section of the Penn Treebank corpus (Langkilde-Geary, 2002). On a set of 2400 automatically-derived inputs, 80% produced output that was about 94% correct when the input was almost fully specified. Accuracy was measured using IBM Bleu scores and NIST simple string accuracy scores which compared outputs to the original sentences. About 57% of the outputs were exact matches with the original. Using minimally specified inputs, accuracy was still more than 51%, 5% of which were exact matches.

HALogen is a successor to Nitrogen (Langkilde, 2000), (Langkilde and Knight, 1998a), and (Langkilde and Knight, 1998b) with a two-stage architecture. In the first stage, symbolic mapping rules transform an input into a forest of possible expressions. In the second stage a statistical ngram model computes the mostly likely N outputs.

### 2 Highlights

Highlights of features and capabilities include:

- Input:
  - broad coverage of syntactic phenomenon

- deep and shallow syntactic and semantic relations available for the input, giving user more control over output, and ability to override system-supplied defaults
- meta \*OR\* nodes possible in input at every level of nesting, not just top level
- repeatable modifier and adverbial relations
- ability to permute or not the ordering of repeated features
- compound values permitted for instance/head relation, to represent scope or constrain constituent ordering
- template-like capability using :template and :filler roles with labels
- Mapping Rules are more versatile and modular:
  - Only 255 hand-written rules used to achieve broad coverage and flexible input abstraction levels
  - feature/value matching capability includes “not”, “all”, “optional”, “exhaustive”
  - rule execution modifiers: repeatable, continue, and cut
  - arbitrary function calls possible
- Ranker: CMU toolkit-built bigram and trigram models, with ‘with-length’ option
- Efficiency: improved cache and rule matching procedure
- Robustness: In case of problematic input, offers choice between constituent dropping versus generation failure

- Weights possible in grammar rules, input, and for concept-to-word mappings
- Polished output

## References

- I. Langkilde and K. Knight. 1998a. Generation that exploits corpus-based statistical knowledge. In *Proc. COLING-ACL*.
- I. Langkilde and K. Knight. 1998b. The practical value of n-grams in generation. In *Proc. International Natural Language Generation Workshop*.
- I. Langkilde-Geary. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proc. INLG*.
- I. Langkilde. 2000. Forest-based statistical sentence generation. In *Proc. NAACL*.