A Graph-to-Sequence Model for AMR-to-Text Generation
Lin Feng Song¹, Yue Zhang², Zhiguo Wang³ and Daniel Gildea¹
¹University of Rochester, ²Singapore University of Technology and Design, ³IBM Research, ⁴Westlake Institute for Advanced Study

Contribution
- Introduce graph recurrent network (GRN) for modeling AMR graph.
- It shows better performance than a sequential LSTM encoder on linearized AMRs.
- We release our code at https://github.com/freesunshine0316/neural-graph-to-seq-mp.

Baseline: sequence-to-sequence

We model the graph state \( g = \{ h_j \}_{j \in V} \) via state transition, \( h_j \) incorporates larger context through the gated (LSTM-based) state transition.

\[
c_j^t, h_j^t = \text{LSTM}(\{x_{i,j}^t\}; \{h_{i,j}^t\}; \{c_{j-1}\})
\]

The representation vector \( x_{i,j}^t \) for edge \( (i, j, l) \) is calculated by the edge label embedding \( e_{ij} \) and the concept of the other node \( v_i \).

\[
x_{i,j}^t = W_e(e_{ij}; e_i) + b_i \quad x_{i,j}^t = W_e(e_{ij}; h_{i,j}^t) + b_i
\]

Copy mechanism

Graph recurrent network (GRN)

Main results
- LDC2015E86, Train/Dev/Test: 16833/1368/1371
- Up to 2M raw data (Gigaword) parsed by JAMR

Dev BLEU scores against transition steps for the graph encoder.

Dev BLEU scores against transition steps for the graph encoder.

Development experiments

Example Outputs

Percentage of Dev AMRs with different diameters.