JBNU at MRP 2020: AMR Parsing using a Joint State Model for Graph-Sequence Iterative Inference

Jinwoo Min, Seung-Hoon Na
Dept. Computer Science, Jeonbuk National University

Jong-Hun Shin, Young-Kil Kim
Electronics and Telecommunication Research Institute (ETRI)
Introduction

• Concentrate on Abstract Meaning Representation (AMR) frameworks

• Models
  – **Joint State Vector**: we propose a joint state model for the graph-sequence iterative inference of (Cai and Lam, 2020)
• **AMR Parsing as Sequence-to-Graph Transduction (Sheng Zhang, ACL ‘19)**

![Diagram](image)

- **Two Stage** (Node Prediction, Edge Prediction)
  - Propose the extended Pointer generator (add target attention from existing nodes)
  - Apply bi-attention attention for edge Prediction
• AMR Parsing via Graph ⇔ Sequence Iterative Inference (Cai and Lam, 2020)

• Iterative Inference (dual decision)
  – To gradually expand the graph, iterative inference is applied at each expansion step.
  – concept solver and the relation solver are conceptually two attention mechanisms over the sequence and graph respectively.
AMR Parsing using a Joint State Model for Graph-Sequence Iterative Inference

- **Simplified Iterative Inference Model**
  - Using joint state vector $z_t$, we generate new concepts and determine relations new concept and partially Constructed graph
Encoder

- **Multi Layer Transformer** (Vaswani et al., 2017)
  - Sentence Encoder: multi Layer transformer Encoder
    - Input features
      - Input words, CNN based char-level word features, POS tags, NER tags
      - BERT-based Features
    
    $$ h_0, h_1, \cdots, h_n = \text{SequenceEncoder}((BOS, w_1, \cdots, w_N)) $$

  - Graph Encoder: multi Layer transformer Decoder
    - Masked Self Attention & Source Attention
    - Input features
      - Input nodes, CNN based char-level node features

    $$ s_0, s_1, \cdots, s_i = \text{GraphEncoder}(G = \{c_1, \cdots, c_i\}) $$
Concept Solver

- Generates a new concept using pointer generator
  - Pointer generator [See et al., 2017]
  - Using joint State Vector $z_t$

$$
q_t = W^Q z_t \\
k_{1:n} = W^K h_{1:n} \\
v_{1:n} = W^V h_{1:n} \\
[\alpha_t, r_t] = Attention(q_t, k_{1:n}, v_{1:n}) \\
z_t' = z_t + r_t
$$

$$
P^{(vocabs)} = \text{softmax} \left( W^{(vocabs)} z_t' + b^{(vocabs)} \right) \\
[p_1, p_2, p_3] = \text{softmax} \left( W^{(switch)} z_t \right) \\
P(c) = p_0 \cdot P^{(vocabs)}(c) + \\
p_1 \cdot \left( \sum_{i \in L(c)} \alpha_t[i] \right) + p_2 \cdot \left( \sum_{i \in T(c)} \alpha_t[i] \right)
$$
Relation Solver

- Determine relations new node and existing nodes
  - Using joint State Vector $z_t$
  - To solve reentrancies problem, we use multi-head attention. For each head $h$, we calculate an attention distribution over all existing node.
  - If no relationship exists, it will point to the dummy node

\[
q_t^h = W^Q_h z_t \\
k_{0:i}^h = W^K_h s_{0:i} \\
v_{0:i}^h = W^V_h s_{0:i} \\
[\beta_t^h, r_t^h] = \text{Attention}(q_t^h, k_{0:i}^h, v_{0:i}^h) \\
\beta_t[i] = \max_{h=1}^H \beta_t^h[i]
\]
Iterative inference : Joint State Model

• **Simplified Iterative Inference**
  
  – define joint State Vector $z_t$
  
  – After selectively sum the information of sequence and partially constructed graph using a gate, we iteratively updates the state vector $z_t$.

\[
\begin{align*}
  z_0 &= \text{fusion}(h_0, s_i) \\
  g_t &= \sigma(z_t) \\
  [-, z_t^{\text{seq}}] &= \text{Attention}(z_t, h_{1:n}, h_{1:n}) \\
  [-, z_t^{\text{graph}}] &= \text{Attention}(z_t, s_{0:i}, s_{0:i}) \\
  z_{t+1} &= z_t + (1 - g_t) z_t^{\text{seq}} + g_t z_t^{\text{graph}}
\end{align*}
\]
Official Results

- Official Results
  - compares the results of the top-2 systems
  - Overall, our system ranks between 3rd and 4th place among all participants which submitted to the AMR framework.

- Future Works
  - joint state models by reformulating an iterative inference based on attention results from the concept and relation solvers.