This note describes a new algorithm for sentence compression using dependency parsing.

1 Standard First-Order Dependency Parsing

The standard algorithm for first order dependency parsing consists of the following rules.

Premise:

$$(\text{◺}, i, i), (\text{◿}, i, i) \quad \forall i \in \{0 \dots n\}$$

Rules:

$$\frac{(\text{◺}, i, k) \quad (\text{◿}, k+1, j)}{(\sqsubset, i, j)} \; (i \rightarrow j) \qquad \forall i \leq k < j$$

$$\frac{(\text{◺}, i, k) \quad (\text{◿}, k+1, j)}{(\sqsupset, i, j)} \; (j \rightarrow i) \qquad \forall i \leq k < j$$

$$\frac{(\sqsubset, i, k) \quad (\text{◺}, k, j)}{(\text{◺}, i, j)} \qquad \forall i < k \leq j$$

$$\frac{(\text{◿}, i, k) \quad (\sqsupset, k, j)}{(\text{◿}, i, j)} \qquad \forall i \leq k < j$$

Goal:

$$(\text{◺}, 0, n)$$

2 Skip Parsing

Premise:

$$(\text{◺}, i, i), (\text{◿}, i, i) \quad \forall i \in \{0 \dots n\}$$

Rules:

$$\frac{(\text{◺}, i, k) \quad (\text{◿}, k+1, j)}{(\sqsubset, i, j)} \; (i \rightarrow j) \qquad \forall i \leq k < j$$

$$\frac{(\text{◺}, i, k) \quad (\text{◿}, k+1, j)}{(\sqsupset, i, j)} \; (j \rightarrow i) \qquad \forall i \leq k < j$$

$$\frac{(⊏, i, k) \quad (◺, k, j)}{(◺, i, j)} \qquad \forall i < k \leq j$$

$$\frac{(◿, i, k) \quad (⊐, k, j)}{(◿, i, j)} \qquad \forall i \leq k < j$$

$$\frac{(◺, i, k-1) \quad (◿, k, k)}{(\dot{⊏}, i, j)} \ \text{skip}(k) \qquad \forall i < k$$

$$\frac{(\dot{⊏}, i, k) \quad (◺, k, k)}{(◺, i, j)} \qquad \forall i < k$$

Goal:

$$(◺, 0, n)$$

3 Skip Bigram Parsing

In this styles of parsing

Premise:

$$(◺, i, i, i), (◿, i, i, i) \quad \forall i \in \{0 \ldots n\}$$

Rules:

$$\frac{(◺, i, i, i)}{(◺, i, i, p)} \quad \forall i, k, 0 \leq i < p \leq n+1$$

$$\frac{(◺, i, k, k+1) \quad (◿, k+1, j, p)}{(⊏, i, j, p)} \qquad \forall 0 \leq i \leq k < j < p$$

$$\frac{(◺, i, k, k+1) \quad (◿, k+1, j, p)}{(⊐, i, j, p)} \qquad \forall i \leq k < j < p$$

$$\frac{(⊏, i, k, k) \quad (◺, k, j, p)}{(◺, i, j, p)} \qquad \forall i < k \leq j < p$$

$$\frac{(◿, i, k, k) \quad (⊐, k, j, p)}{(◿, i, j, p)} \qquad \forall i \leq k < j$$

$$\frac{(\text{◺}, i, k-1, p) \quad (\text{◿}, k, k)}{(\overset{\cdots}{\text{⏢}}, i, j, p)} \qquad \forall i < k < p$$

$$\frac{(\overset{\cdots}{\text{⏢}}, i, k, p) \quad (\text{◺}, k, k, k)}{(\text{◺}, i, j, p)} \qquad \forall i < k < p$$

Goal:

$$(\text{◺}, 0, n, n+1)$$

4 Second-Order Dependency Parsing

Premise:

$$(\text{◺}, i, i), (\text{◿}, i, i) \quad \forall i \in \{0 \ldots n\}$$

Rules:

$$\frac{(\text{◺}, i, k) \quad (\text{◿}, k+1, j)}{(\square, i, j)} \qquad \forall i \leq k < j$$

$$\frac{(\text{◺}, i, i) \quad (\text{◿}, i+1, j)}{(\text{⏢}_{\searrow}, i, j)} \; (i \to j) \qquad \forall i < j$$

$$\frac{(\text{◺}, i, j-1) \quad (\text{◿}, j, j)}{(\text{⏢}_{\nearrow}, i, j)} \; (j \to i) \qquad \forall i < j$$

$$\frac{(\text{⏢}_{\searrow}, i, k) \quad (\square, k, j)}{(\text{⏢}_{\searrow}, i, j)} \; (i \to k \;\, j) \qquad \forall i \leq k < j$$

$$\frac{(\square, i, k) \quad (\text{⏢}_{\nearrow}, k, j)}{(\text{⏢}_{\nearrow}, i, j)} \; (j \to k \;\, i) \qquad \forall i \leq k < j$$

$$\frac{(\text{⏢}_{\searrow}, i, k) \quad (\text{◺}, k, j)}{(\text{◺}, i, j)} \qquad \forall i < k \leq j$$

$$\frac{(\text{◿}, i, k) \quad (\text{⏢}_{\nearrow}, k, j)}{(\text{◿}, i, j)} \qquad \forall i \leq k < j$$

Goal:

$$(\text{◺}, 0, n)$$