

Lecture 05. Dependency Parsing

I. Syntactic Structure : Consistency and Dependency

1. Sentence Structure

-문장의 구조를 파악하는 방법

1) Parse-structure grammar(context-free grammar, CFG)

2) dependency structure grammar

2. Phase-structure grammar

-문장을 unit으로 구성, 각 unit들을 progressively nest

-starting unit : words -> phrases -> bigger phrases

3. Dependency structure grammar

-한 단어가 어떤 다른 단어에 의존적인지 나타내는 방식

II. Dependency Grammar and Treebanks

1. Treebank

-사이 직접 문항들의 dependency를 파악해서 dependency structure를 구성한 데이터셋

-영어 뿐만 아니라 다양한 언어들에 대해서 생성

2. Dependency parser

-다양한 method로 dynamic programming, graph algorithm, constraint satisfaction, transition-based parsing(deterministic dependency parsing)을 쓴다.

3. Greedy transition-based parsing

-초기 상태 : [root]만 요소로 보유하고 있는 stack, 문장 전체를 구성하는 단어들로 구성된 buffer, 결과를 저장하는 A

-종료 조건 : stack에 [root]만 원소로 남는 경우, buffer가 비어있는 경우

-문제점 : 종류가 정말 많지만 dependency를 찾아내는데, 종류가 정말 많지만 실제로 사

용되는 것이 매우 적고(sparse) 관계를 개선하는데 비용이 많이 든다.

-> Neural Dependency Parser사용

4. Neural Dependency Parser

-neural net을 이용해서 문장의 dependency를 파악하는 방법

-Maltparser : 다음 action(shift, left, right arc)을 어떻게 취해야 하는지 softmax classifier에 의해서 결정한다.

-C&M parser : Maltparser와 유사한 방식으로 작동한다. 단어와 함께 POS(Part-of-Speech), dependency-label을 input으로 제공한다. 좋은 성능과 빠른 처리 속도를 제공한다.

-evaluation

1) unlabeled attachment score(UAS) : arc(dependency)만 일치하는지 확인한다. Label은 별도로 확인하지 않는다.

2) labeled attachment score(LAS) : arc와 label 모두 일치하는지 확인한다.

III. Transition-based dependency parsing