Reinforcement Learning for NLP

Advanced Machine Learning for NLP
Jordan Boyd-Graber

DEPENDENCY PARSING

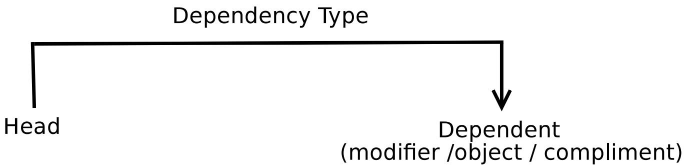
Adapted from slides by Neelamadhav Gantayat and Ryan MacDonald

- Turns sentence into syntactic structure
- Essential for information extraction and other NLP tasks

Lucien Tesnière, 1959

The sentence is an organized whole, the constituent elements of which are words. Every word that belongs to a sentence ceases by itself to be isolated as in the dictionary. Between the word and its neighbors, the mind percieves connections, the totality of which forms the structure of the sentence. The structural connections establish dependency relations between the words.

- **Basic Assumption:** Syntactic structure essentially consists of lexical items linked by binary asymmetrical relations called dependencies.



Example of dependency parser output

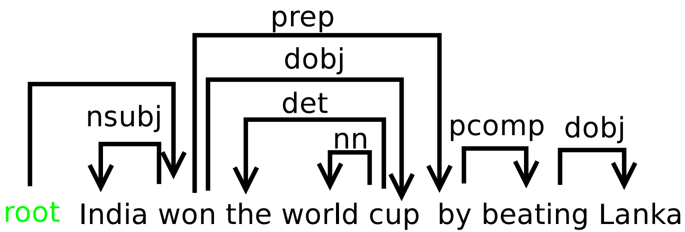


Figure: Output of Stanford dependency parser

Example of dependency parser output

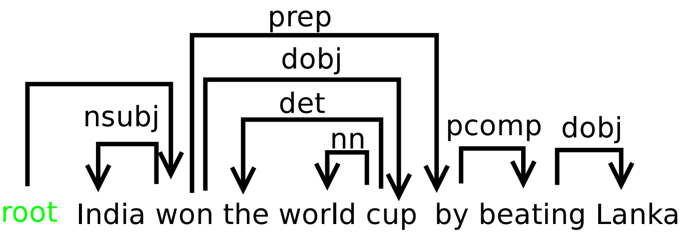


Figure: Output of Stanford dependency parser

- Verb has an artificial root
- Notion of phrases: “by” and its children
- So how do we choose these edges?

Criteria for dependency

D is likely a dependent of head H in construction C:

- H determines syntactic category of C and can often replace C
- H gives semantic specification of C; D specifies H
- H is obligatory; D may be optional
- H selectes D and determines whether D is obligatory
- The form of D depends on H (agreement or government)
- The linear position of D is specified with reference to H

Which direction?

Some tricky cases . . .

- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation

`I can see that they rely on this and that.`

Which direction?

Some tricky cases . . .

- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation

```
I can see that they rely on this and that.
```

Which direction?

Some tricky cases . . .

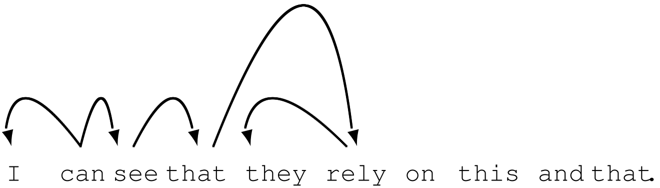
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation

I can see that they rely on this and that.

Which direction?

Some tricky cases . . .

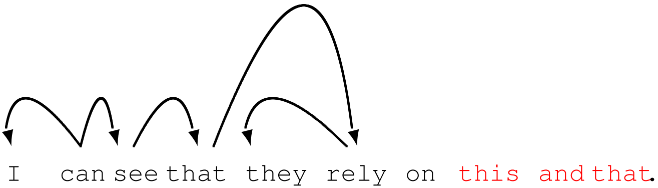
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

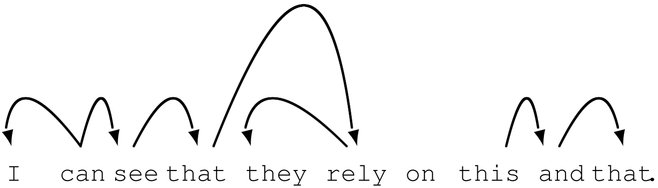
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

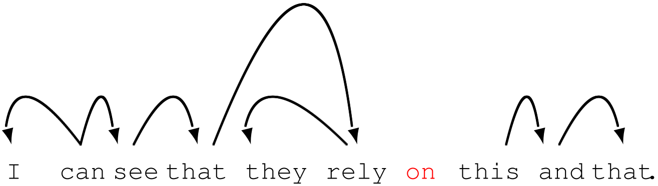
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

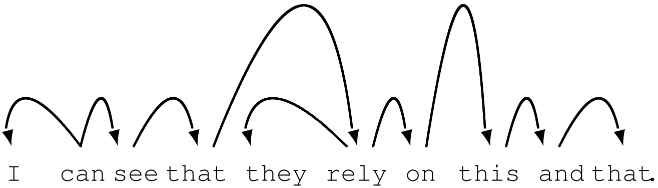
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

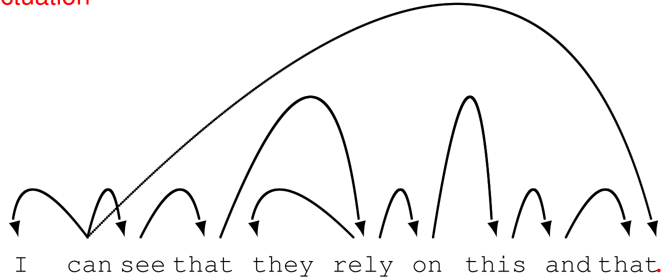
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

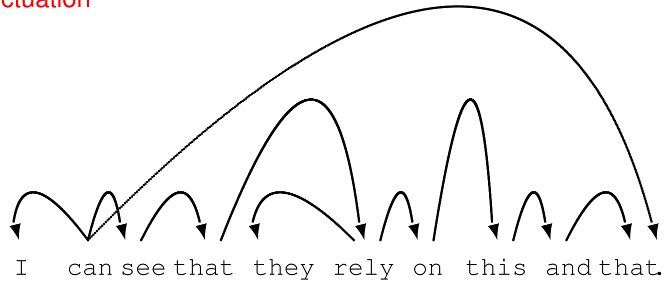
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



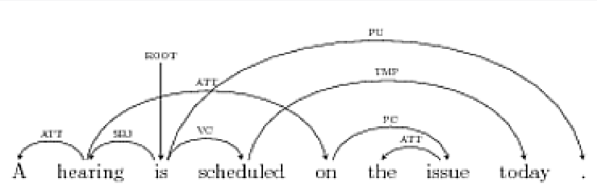
Dependency Parsing

- Input: Sentence $x = w_0, w_1, ..., w_n$
- Output: Dependency graph $G = (V, A)$ for x where:
  - $V = 0, 1, ..., n$ is the vertex set,
  - A is the arc set, i.e., $(i, j, k) \in A$ represents a dependency from w_i to w_j with label $l_k \in L$

Projectivity

- Equivalent to planar embedding
- Most theoretical frameworks do not assume projectivity
- Non-projective structures needed for free word order and long-distance dependencies

Non-projective example

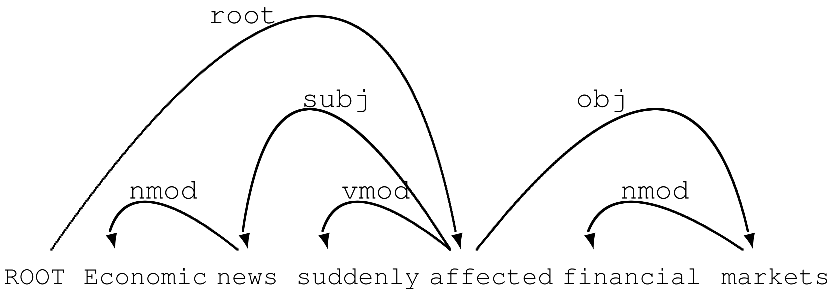


- The algorithm later we'll discuss is projective

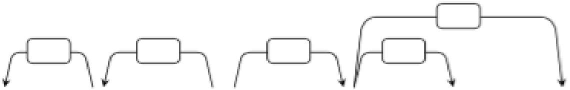
Which direction?

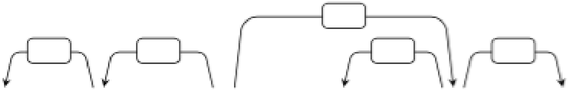
Some clear cases . . .

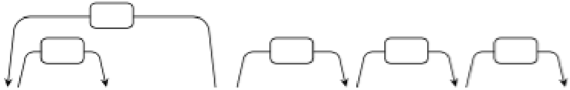
- Modifiers: “nmod” and “vmod”
- Verb slots: “subject” and “object”



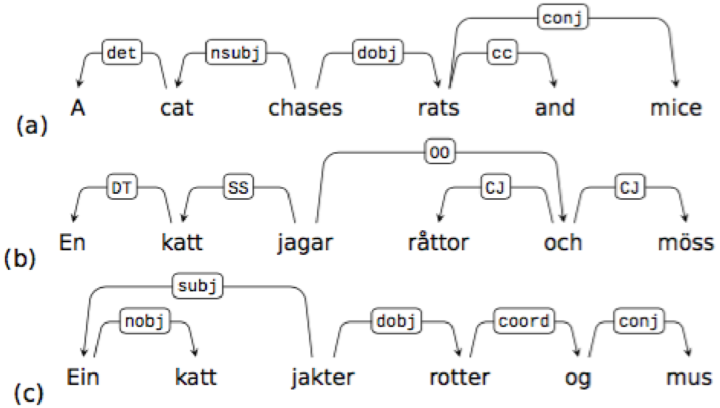
Not all choices are consistent


(a)


(b)


(c)

Not all choices are consistent



Universal Dependencies Project

http://universaldependencies.org/

Mapping between languages that:

1. satisfactory on linguistic grounds for the analysis of individual languages.
2. good for linguistic typology, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.
3. suitable for rapid, consistent annotation by a human annotator.
4. suitable for training highly accurate parsers.
5. easily comprehensible and used by a non-linguist, whether a language learner or an engineer with prosaic needs for language processing.
6. useful for downstream language understanding tasks (relation extraction, reading comprehension, machine translation, . . .).