Dependency Parsing

Jordan Boyd-Graber

University of Maryland

Intro

Adapted from slides by Neelamadhav Gantayat and Ryan MacDonald

Dependency Syntax

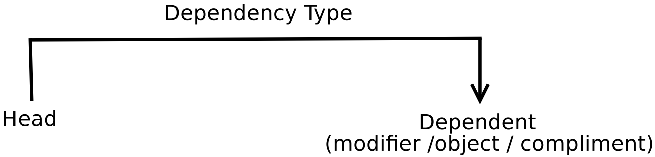
- Turns sentence into syntactic structure
- Essential for information extraction and other NLP tasks

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.

Dependency Grammar

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



Example of dependency parser output

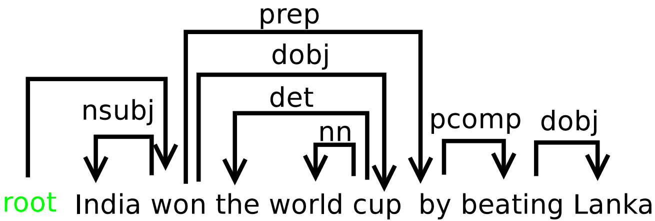


Figure: Output of Stanford dependency parser

Example of dependency parser output

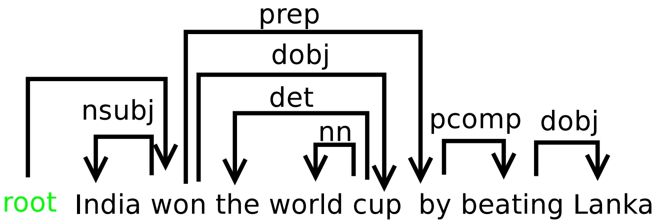


Figure: Output of Stanford dependency parser

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

Criteria for dependency

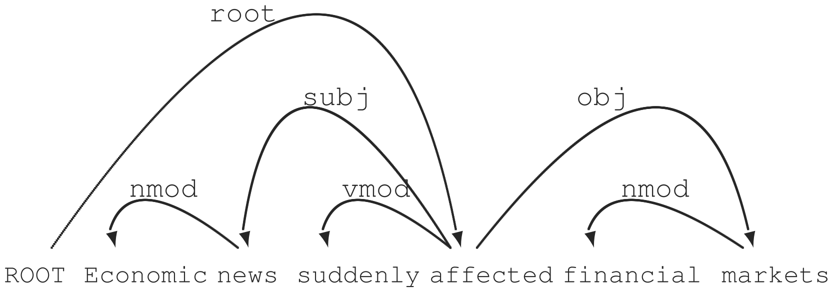
D is likely a dependent of head *H* in construction *C*:

- *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 clear cases ...

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



Which direction?

Some tricky cases . . .

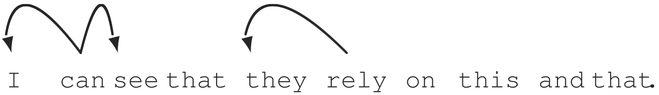
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation

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

Which direction?

Some tricky cases . . .

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

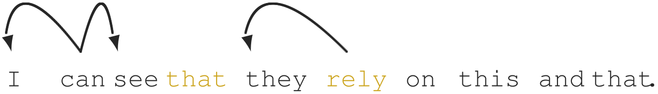


I can see that they rely on this and that.

Which direction?

Some tricky cases . . .

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

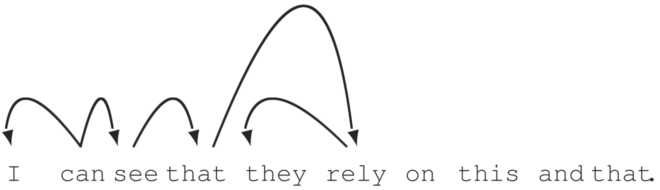


I can see that they rely on this and that.

Which direction?

Some tricky cases . . .

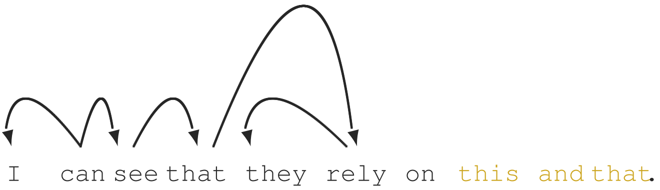
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

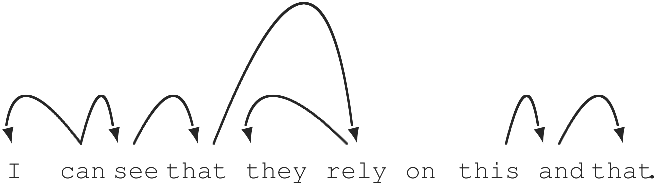
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

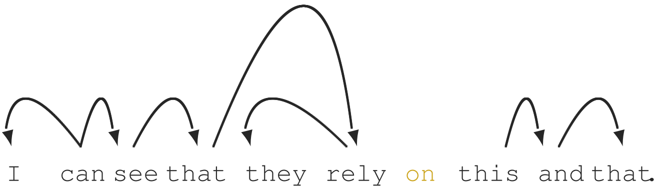
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

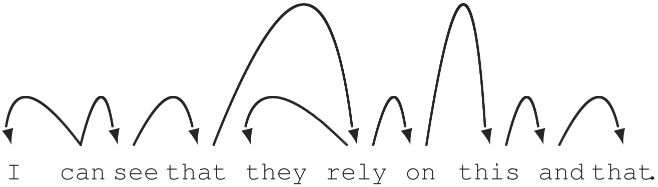
- Complex verb groups
- Subordinate clauses
- Coordination
- Prepositions
- Punctuation



Which direction?

Some tricky cases . . .

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

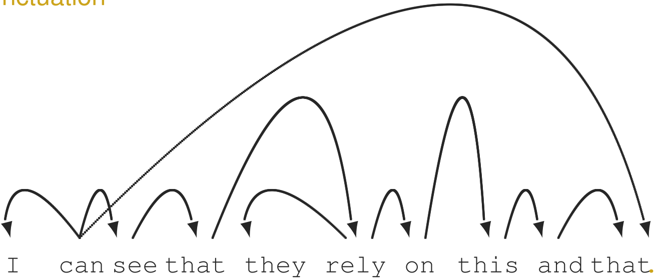


I can see that they rely on this and that.

Which direction?

Some tricky cases . . .

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

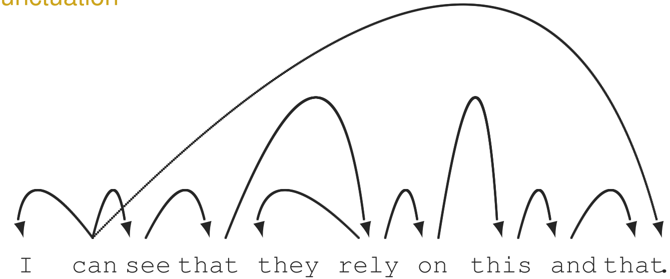


I can see that they rely on this and that.

Which direction?

Some tricky cases . . .

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



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 $l_k \in L$
- Notational Conventions
    - $i \longrightarrow j \equiv \exists k : (i, j, k) \in A$ (unlabeled dependency)
    - $i \longleftrightarrow j \equiv i \longrightarrow \vee j \longrightarrow i$ (undirected dependency)
    - $i \longrightarrow *j \equiv i = j \vee \exists i' : i \longrightarrow i', i' \longrightarrow *j$ (unlabeled closure)
    - $i \longleftrightarrow *j \equiv i \vee \exists i' : i \longleftrightarrow i','i \longleftrightarrow *j$ (undirected closure)

Conditions

- Intuitions
  - Syntactic structure is complete (Connectedness)
  - Syntactic structure is hierarchical (Acyclic)
  - Every word has at most one syntactic head (Single-Head)
- Connectedness is enforced by adding special root node

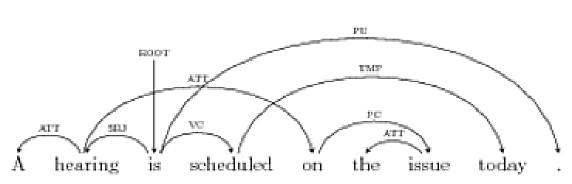
Conditions

- Connected: $\forall i, j \in V, i \longleftrightarrow * j$
- Acyclic: If $i \longrightarrow j$, then not $j \longrightarrow * i$
- Single-head: If $i \longrightarrow j$, then not $i' \longrightarrow j \forall i' \neq i$
- Projective: If $i \longrightarrow j$, then $i \longrightarrow * i'$ for any i' such that $i < i' < j$ or $j < i' < i$.

Projectivity

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

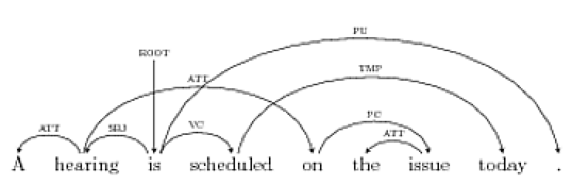
Non-projective example



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

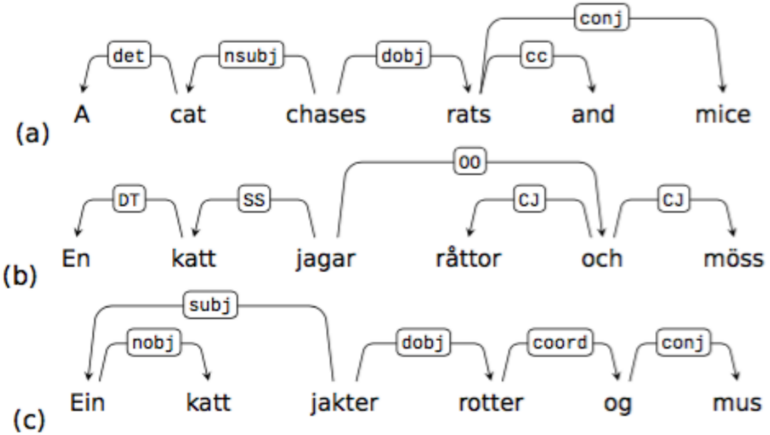
Evaluation Methodology

- How many sentences are exactly correct

Evaluation Methodology

- How many sentences are exactly correct
- Edge accuracy
  1. Labeled attachment score (LAS):
     i.e. Tokens with correct head and label
  2. Unlabeled attachment score (UAS):
     i.e. Tokens with correct head
  3. Label accuracy (LA):
     i.e. Tokens with correct label
- Performance on downstream task (e.g., information extraction)

Not all Choices are Consistent



Universal Dependencies Project

`http://universaldependencies.org/`

Mapping between languages that:

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