Syntax

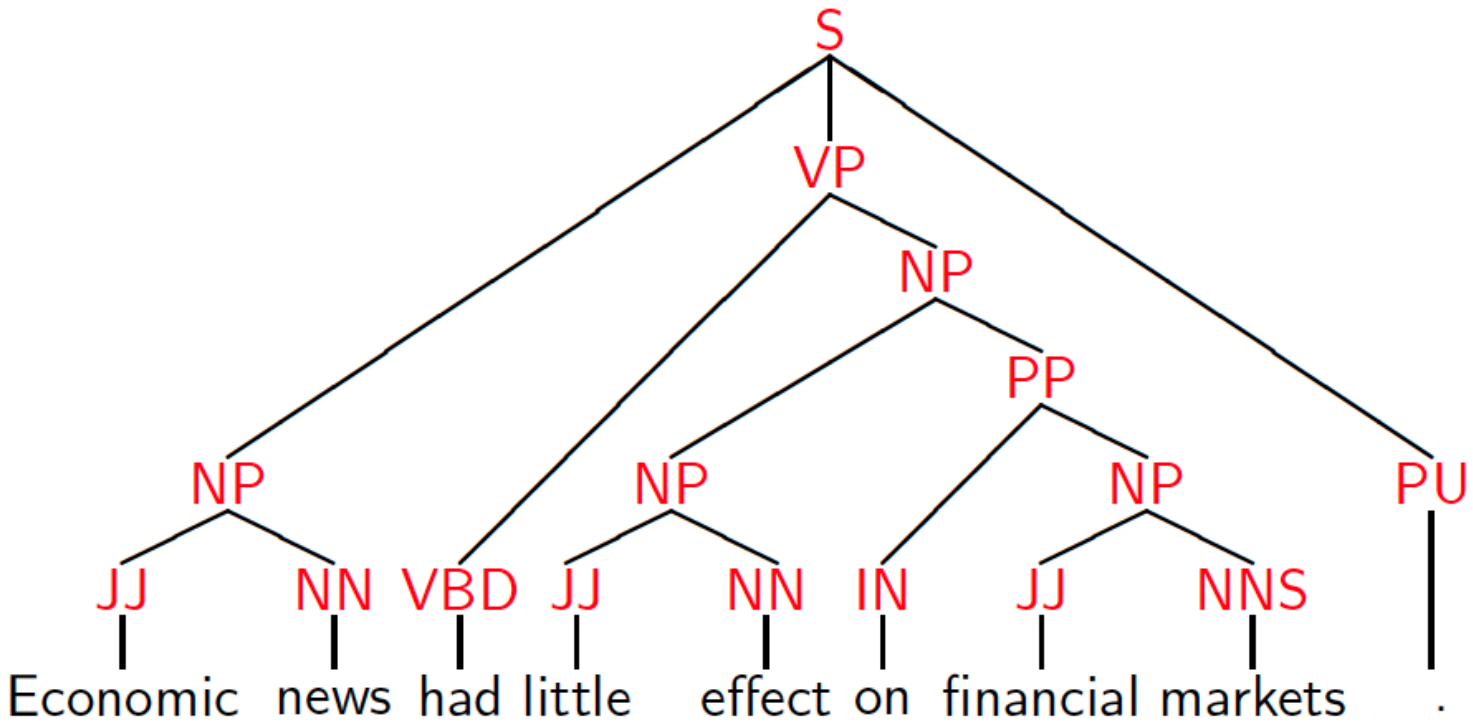
Dependency Graphs

Till now, we studied Constituency Parsing that used Context Free Grammars to capture word groups

Now we will study another form of parsing - Dependency Parsing - that tries to infer the relationship between words in a given text

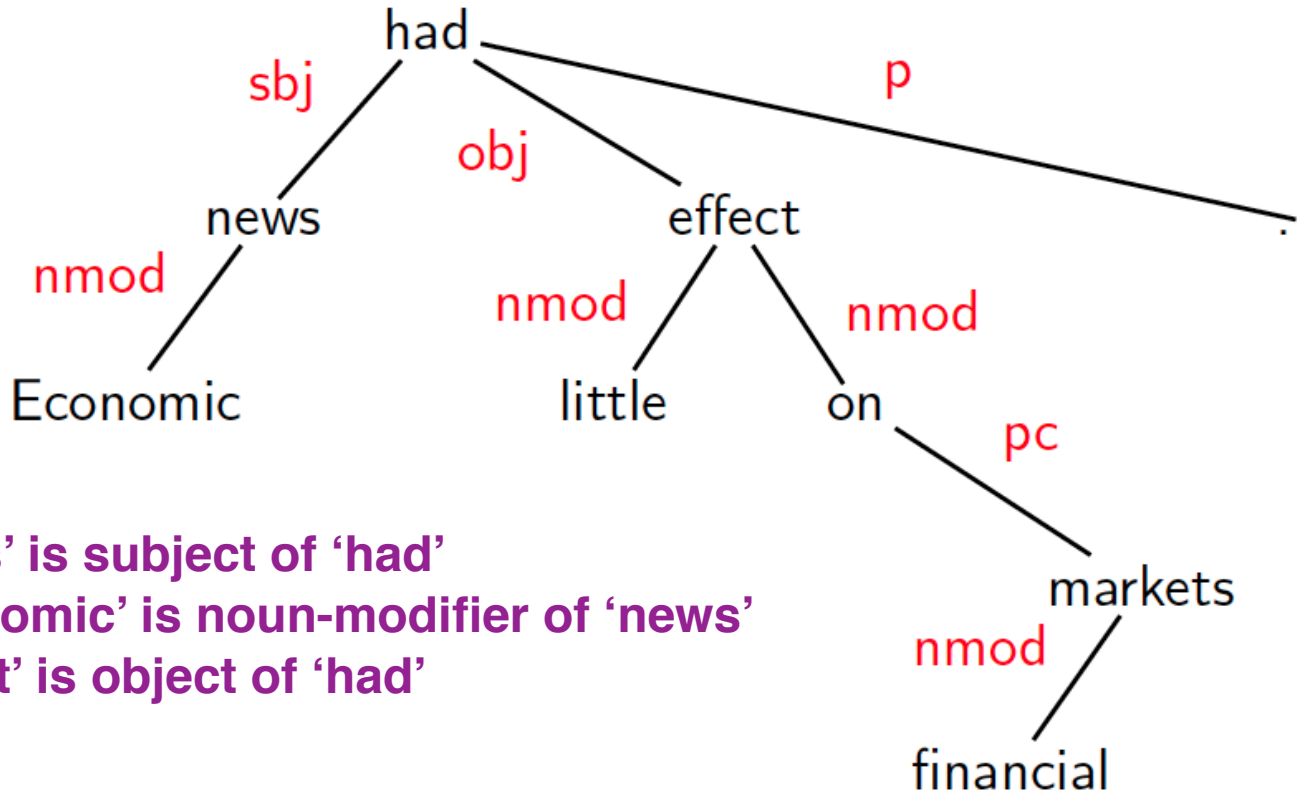
Phrase Structure

A tree structure where the nodes are different Parts-of-speech (JJ, NN, VBD, …) and phrase-types (NP, VP, PP, …)



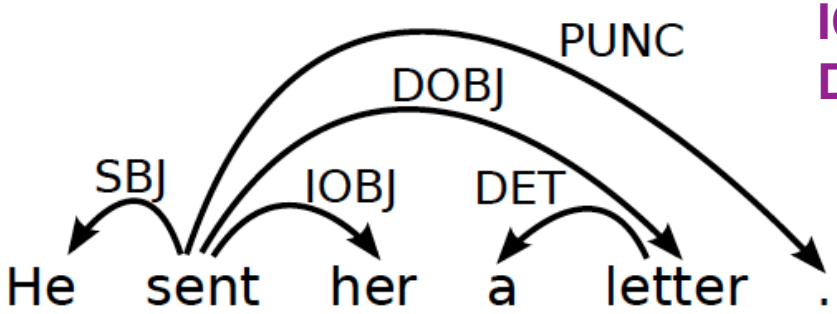
Dependency Structure Representation

A different tree structure where the nodes are the words themselves. The edges (to be considered as directed) denote the relations between the words, e.g., subject, object, …



'news' is subject of 'had'
'Economic' is noun-modifier of 'news'
'effect' is object of 'had'

Dependency Structure

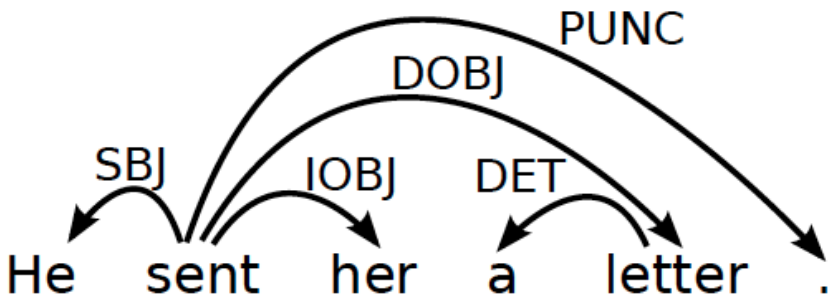


SBJ: subject
IOBJ: indirect object
DOBJ: direct object

- Connects the words in a sentence by putting arrows between the words.
- Arrows show relations between the words and are typed by some grammatical relations.
- Arrows connect a head (governor, superior, regent) with a dependent (modifier, inferior, subordinate).
- Usually dependencies form a tree.

Criteria for Heads and Dependents

H—>D is a construction C
letter —> a



Linear positions: English mostly follows subject-verb-object

Verb —> subject
Verb —> object

Criteria for a syntactic relation between a head H and a dependent D in a construction C

- H determines the syntactic category of C; H can replace C.
- D specifies H.
- H is obligatory; D may be optional.
- H selects 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.

Comparison

Comparison between phrase structure and dependency structure

Phrase structures explicitly represent

- Phrases (nonterminal nodes)
- Structural categories (nonterminal labels)

E.g., noun phrase, verb phrase

Dependency structures explicitly represent

- Head-dependent relations (directed arcs)
- Functional categories (arc labels)

E.g., verb —> subject
verb —> object

Dependency Graphs

- A dependency structure can be defined as a directed graph G, consisting of
  - a set V of nodes,
  - a set A of arcs (edges),
- Labeled graphs:
  - Nodes in V are labeled with word forms (and annotation).
  - Arcs in A are labeled with dependency types.

Annotations like POS tags can also be made nodes

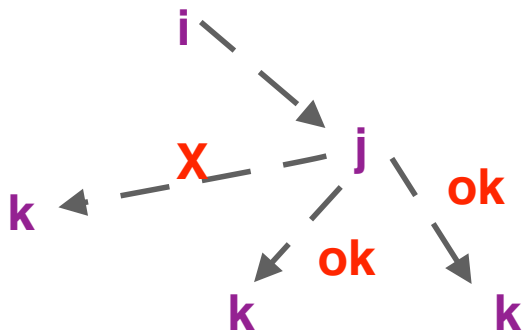
Dependency types: SBJ, OBJ, ...

Dependency Graphs

- A dependency structure can be defined as a directed graph G, consisting of
  - a set V of nodes,
  - a set A of arcs (edges),
- Labeled graphs:
  - Nodes in V are labeled with word forms (and annotation).
  - Arcs in A are labeled with dependency types.
- Notational convention:
  - Arc (w_i, d, w_j) links head w_i to dependent w_j with label d
  - $w_i \xrightarrow{d} w_j \Leftrightarrow (w_i, d, w_j) \in A$
  - $i \rightarrow j \equiv (i, j) \in A$
  - $i \rightarrow^* j \equiv i = j \vee \exists k : i \rightarrow k, k \rightarrow^* j$ **There is a path from i to j**

Formal conditions on Dependency Graphs

- G is connected:
  - For every node i there is a node j such that $i \rightarrow j$ or $j \rightarrow i$.
- G is acyclic:
  - if $i \rightarrow j$ then not $j \rightarrow^* i$.
- G obeys the single head constraint:
  - if $i \rightarrow j$ then not $k \rightarrow j$, for any $k \neq i$.
- G is projective:
  - if $i \rightarrow j$ then $j \rightarrow^* k$, for any k such that both j and k lie on the same side of i.

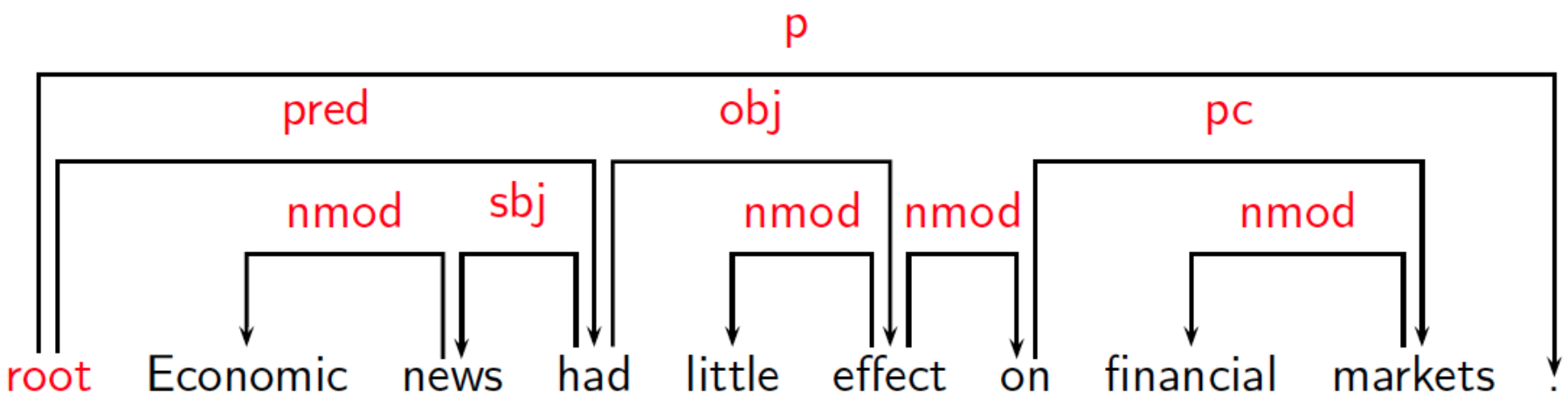


Projectivity is language-dependent

Formal Conditions: Basic Intuitions

Connectedness, Acyclicity and Single-Head

- **Connectedness:** Syntactic structure is complete.
- **Acyclicity:** Syntactic structure is hierarchical.
- **Single-Head:** Every word has at most one syntactic head.
- **Projectivity:** No crossing of dependencies.



Dependency Parsing

Dependency Parsing

- **Input:** Sentence $x = w_1, \ldots, w_n$
- **Output:** Dependency graph G

Parsing Methods

- Deterministic Parsing
- Maximum Spanning Tree Based
- Constraint Propagation Based

The first two methods rely on labeled training data (data-driven parsing). The third method does not need any labeled data, but needs to know some constraints that the grammar follows.