Advanced Natural Language Processing
Lecture 15
Dependency Grammar

(slides by Mark Steedman, Bonnie Webber and Johanna Moore)

25 October 2013

Based on: Joakim Nivre, Dependency Grammar and Dependency Parsing
Lecture 13 invoked **Dependency Structures** for headed PCFG rules, whose attachment probabilities could reflect relations between specific lexical items:
Here we’ll consider these structures, their properties and other things they are useful for.

- In CFGs, syntactic parse trees include phrasal nodes representing *constituents*.

- It turns out you can get a lot done with just binary relations among the words in an utterance.

- In a dependency grammar framework, a parse is a tree where:
  - the nodes stand for the words in an utterance
  - the arcs between the words represent dependency relations between pairs of words
  - relations may or may not be typed (labeled)
Constituency vs. Dependency

S

VP

NP

NP

NP

NP

NP

PP

JJ

NN

VBD

JJ

NN

IN

JJ

NNS

Economic

news

had

little

effect

on

financial

markets

.  

P

NMOD

SBJ

OBJ

NMOD

NMOD

NMOD

JJ

NN

VBD

JJ

NN

IN

JJ

NNS

Economic

news

had

little

effect

on

financial

markets

.
Dependency vs. Constituency

In CFG-style phrase-structure grammars, the main focus is on constituents.

E.g., VP seems to be a constituent of English. It can be (a) deleted as a unit, (b) conjoined as a unit, and/or (c) fronted as a unit.

(1) a. I like ice cream. Do you 0? (VP ellipsis)
b. I like ice cream and hate bananas. (VP conjunction)
c. I said I would hit Fred, and hit Fred I did. (VP fronting)

Not all languages have a VP constituent — e.g., German doesn’t.

Essentially, the patterns of English lead one to think in terms of structure and constituency.
Dependency vs. Constituency

The patterns of languages that aren’t as *configurational* as English may be better described in terms of dependencies between their words.

(2)  
   a. A verb may have an agent, with *nominative* inflection, that may be found before or after it.  
   b. It may have an object, with *accusative* inflection, somewhere else.  
   c. It may have other arguments elsewhere that play other roles.  

Arguments depend on their heads, and this is the essence of a Dependency Grammar.

Even though English is *configurational*, it too can be described in terms of dependencies.
Dependency as Extreme Lexicalization

- Dependencies express how words in a sentence depend on other words, motivating their presence in the sentence and what they are contributing.

- Linguistic analysis based on dependency is clearly going to be even more lexicalized, than in Tree Adjoining Grammar.

- A dependency grammar attempts to directly account for patterns of dependencies between words, but (depending on the language) will also have to provide some account of word order and configurational patterning.
Dependency Relations

Key to a Dependency Grammar are its dependency relations, all of which are binary and asymmetric. All relations hold between

- a head (H)
- a dependent (D)

Economic news had little effect on financial markets.

[From Joakim Nivre, Dependency Grammar and Dependency Parsing]

How do we identify each of the heads and dependents in a sentence?
Dependency Relations: Endocentric vs. Exocentric

A dependency relation is either endocentric or exocentric, depending on how its head relates to the rest of the construction:

- In an endocentric construction, the dependent of a head can be dropped without damaging the grammaticality of the sentence.

  (3) Economic news had little effect on financial markets. (endocentric: financial ← market)

- In an exocentric construction, dropping the dependent would make the sentence ungrammatical.

  (4) Economic news had little effect on financial markets. (exocentric: on → markets)
Dependency Relations: Complements vs. Modifiers

The dependent in a dependency relation may serve its head as

- a complement – e.g., both news and market serve as complements (i.e., arguments) to affected in

(5) The news affected the market.

- a modifier (i.e., adjunct) – e.g., within a week modifies affected in

(6) Within a week, the news affected the market.

- a specifier – e.g., the specifies market in

(7) The news affected the market.
Dependency Relations: Valency

The number of complements (i.e., arguments) that a head takes is its valency, i.e., its predicate-argument structure, which is the basis for its meaning.

- **sigh** has valency \(= 1\)
- **affect** has valency \(= 2\)
- **give** and **exchange** both have valency \(= 3\)

Dependents that correspond to arguments of the predicate can be obligatory or optional, but can occur only once with each predicate instance.

Dependents that are not arguments tend to be optional and can occur 0 or more times.
Problems

While some constructions fit well into dependency relations, others do not, e.g.,

- coordination
- auxiliary verb constructions
- complementizers (*that*)

Nevertheless, *some* stand has to be taken, since the only thing a dependency parser can produce is a dependency structure!
Two Analyses of Coordination

[Diagram showing two analyses of coordination with the words They operate ships and and banks.]
Dependency Relations: One or more inventories

Dependency Grammars differ in their inventory of dependency relations, and in whether they produce a single analysis based on one inventory (a *monostratal theory*) or several analyses based on multiple inventories (a *multi-stratal theory*).

[from M. Buch-Kromann & Iorn Korzen, The unified annotation of syntax and discourse in the Copenhagen Dependency Treebanks, *Proceedings of the Fourth Linguistic Annotation Workshop*, Uppsala, July 2010.]
Dependency Relations: Syntactic or Semantic

<table>
<thead>
<tr>
<th>Surface-oriented grammatical functions</th>
<th>Semantic-oriented role types</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject, nsubj, csubj</td>
<td>agent</td>
</tr>
<tr>
<td>object, dobj</td>
<td>patient</td>
</tr>
<tr>
<td>nmodifier, nmod</td>
<td>goal</td>
</tr>
<tr>
<td>etc</td>
<td>etc</td>
</tr>
</tbody>
</table>

These may yield dependency structures that differ only in their labels, or the structures themselves may differ:

(8) I believe in the system

- Surface-oriented: believe→in, in→system
- Semantics-oriented: believe→system, system→in

How about the analysis of “had forced” on the previous slide?
Dependency Graphs

Dependency representation of a sentence, consisting of its lexical elements linked by binary asymmetric relations, is a (labelled or unlabelled) directed graph.

Properties of Dependency Graphs:

- **Connected** – Every node is related to at least one other node, and (through transitivity) to the root node.

- **Single-headed** – Every node is a dependent of at most one head.

- **Acyclic** – The graph should not contain cycles of directed arcs.

Graph is a rooted tree with a single root node that is not dependent on any other node.
Dependency Graphs: Projectivity

There may also be constraints relating dependency structures to linear realizations:

- A graph is **projective** w.r.t. the linear order of its nodes if, $\forall$ arcs $h \rightarrow d$ and node $w$, $w$ occurs between $h$ and $d$ in linear order only if $w$ is **dominated** by $h$ (i.e., in the transitive closure of the arc relation).

A dependency graph is projective if, when all the words are written in linear order, the edges can be drawn above the words without crossing.
A Projective Dependency Graph

(9) I heard Cecilia teach the horses to sing.

Choose any word in the sentence and test that the graph satisfies this definition.
Dependency Graphs: Projectivity

• “Nonprojective” dependency graph: $w$ can occur between $h$ and $d$ in linear order without being dominated by $h$. (Consider $w=\text{"hoord"}$ below, between $h=\text{"leren"}$ and $d=\text{"Cecilia"}$.)

\[(10)\quad \text{dat ik Cecilia de paarden hoord leren zingen.}\]

“Nonprojective” means “non-context free”.

Most practical dependency parsers assume projectivity, though most dependency-based linguistic theories do not.
Class exercise: Dependency Analysis

The sequence

(11) Time flies like an arrow

has three interpretations corresponding to:

1. Time/NN flies/VBZ like/IN an/DT arrow/NN

2. Time/VB flies/NNS like/IN an/DT arrow/NN

3. Time/NN flies/NNS like/VBZ an/DT arrow/NN

Draw a dependency structure for each interpretation. Make sure that it is connected, single-headed and acyclic.
You can use the following surface-oriented dependency relations:

- nsubj for the nominal subject of a verb
- dobj for its direct object
- nmod for the pre- or post-modifier of a noun
- pobj for the object of a preposition (i.e., relating the preposition with the head of its object)

Is the dependency structure for each interpretation projective or non-projective?