Motivation and Contents

Not a coherent grammatical framework: wide range of different kinds of DG
- just as there are wide ranges of “generative syntax”
- Different core ideas than phrase structure grammar

Dependency grammar is important for those interested in CL:
- Increasing interest in dependency-based approaches to syntactic parsing in recent years

Introduction

Basic Concepts of Dependency Syntax

The basic idea:
- Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.
- In the (translated) words of Lucien Tesnière [Tesnière (1959)]:
  - The sentence is an organized whole, the constituent elements of which are words. [1.2] 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 perceives connections, the totality of which forms the structure of the sentence. [1.3] The structural connections establish dependency relations between the words. Each connection in principle unites a superior term and an inferior term. [2.1] The superior term receives the name governor. The inferior term receives the name subordinate. Thus, in the sentence Alfred parle […] parle is the governor and Alfred the subordinate. [2.2]

Overview: constituency

(1) Small birds sing loud songs

What you might be more used to seeing:

A corresponding dependency tree representation [Hudson (2000)]:

What are Dependency Relations?

- DG is based on dependency relations between words:
  - A → B means A governs B or B depends on A …
  - Dependency relations can refer to syntactic properties, semantic properties, or a combination of the two
    - Some variants of DG separate syntactic and semantic relations by representing different layers of dependency structures
    - These relations are generally syntactic functions: subject, object/complement, adjunct, etc.
    - Subject/Agent: John fished.
    - Object/Patient: Mary hit John.
- PSG is based on groupings, or constituents
  - Grammatical relations are not usually seen as primitives, but as being derived from structure
For the sentence *John loves Mary*, we have the relations:

- loves \( \rightarrow \text{subj} \) John
- loves \( \rightarrow \text{obj} \) Mary

Both *John* and *Mary* depend on *loves*, which makes *loves* the head, or root, of the sentence (i.e., there is no word that governs *loves*)

The structure of a sentence consists of the set of pairwise relations among words.
Economic news had little effect on financial markets.

Dependency structures explicitly represent
- head-dependent relations (directed arcs),
- functional categories (arc labels),
- possibly some structural categories (parts-of-speech).

Phrase structures explicitly represent
- phrases (nonterminal nodes),
- structural categories (nonterminal labels),
- possibly some functional categories (grammatical functions).

Hybrid representations may combine all elements.

Theoretical Frameworks
- Word Grammar (WG) [Hudson(1984), Hudson(1990)]
- Functional Generative Description (FGD) [Szall et al.(1986)Szall, Hajicova and Paneva]
- Dependency Unification Grammar (DUG) [Hellwig(1986), Hellwig(2003)]
- Meaning-Text Theory (MTT) [Mel'cuk(1988)]
- Functional Dependency Grammar (FDG) [Tapanainen and Järvinen(1997), Järvinen and Tapanainen(1998)]

Some Theoretical Issues
- Dependency structure sufficient as well as necessary?
- Mono-stratal or multi-stratal syntactic representations?
- What is the nature of lexical elements (nodes)?
  - Morphemes?
  - Word forms?
  - Multi-word units?
- What is the nature of dependency types (arc labels)?
  - Grammatical functions?
  - Semantic roles?
- What are the criteria for identifying heads and dependents?
- What are the formal properties of dependency structures?
Criteria for Heads and Dependents

- Criteria for a syntactic relation between a head H and a dependent D in a construction C [Zwicky (1983), Hudson (1990)]:
  1. H determines the syntactic category of C; H can replace C.
  2. H determines the semantic category of C; D specifies H.
  3. H is obligatory; D may be optional.
  4. H selects D and determines whether D is obligatory.
  5. The form of D depends on H (agreement or government).
  6. The linear position of D is specified with reference to H.

- Issues:
  - Syntactic (and morphological) versus semantic criteria
  - Exocentric versus endocentric constructions

Some Clear Cases

<table>
<thead>
<tr>
<th>Construction</th>
<th>Head</th>
<th>Dependent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exocentric</td>
<td>Verb</td>
<td>Subject (sbj)</td>
</tr>
<tr>
<td></td>
<td>Verb</td>
<td>Object (obj)</td>
</tr>
<tr>
<td>Endocentric</td>
<td>Verb</td>
<td>Adverbial (vmod)</td>
</tr>
<tr>
<td></td>
<td>Noun</td>
<td>Attribute (nmod)</td>
</tr>
</tbody>
</table>

I can see that they rely on this and that.

Economic news suddenly affected financial markets.

Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation

I can see that they rely on this and that.

Punctuation

I can see that they rely on this and that.
Some Tricky Cases

- Complex verb groups (auxiliary ↔ main verb)
- Subordinate clauses (complementizer ↔ verb)
- Coordination (coordinator ↔ conjuncts)
- Prepositional phrases (preposition ↔ nominal)
- Punctuation

I can see that they rely on this and that.

I can see that they rely on this and that.

I can see that they rely on this and that.

I can see that they rely on this and that.
## Dependency Graphs

- A dependency structure can be defined as a directed graph $G$, consisting of:
  - a set $V$ of nodes,
  - a set $E$ of arcs (edges),
  - a linear precedence order $<$ on $V$ (not in every theory)
- Labeled graphs:
  - Nodes in $V$ are labeled with word forms (and annotation).
  - Arcs in $E$ are labeled with dependency types.
- Notational conventions ($i, j \in V$):
  - $i \rightarrow j \equiv (i, j) \in E$
  - $i \rightarrow^* j \equiv i = j \lor \exists k : i \rightarrow k, k \rightarrow^* j$

## Formal Conditions on Dependency Graphs

- $G$ is (weakly) **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 $i \rightarrow^* k$, for any $k$ such that $i < k < j$ or $j < k < i$.

## Projectivity

- Most theoretical frameworks do **not** assume projectivity.
- Non-projective structures are needed to account for:
  - long-distance dependencies,
  - free word order.

## Grammaticality

### Obtaining word order constraints

Non-projectivity in principle could allow any word order
- This would clearly overgenerate for most languages

Some DGs use projectivity constraints [Hudson(1990)]:

1. with great difficulty
2. $^*$great with difficulty

$^*$great with difficulty is ruled out because branches would have to cross in that case
- In general, this is too strong of a constraint
Obtaining word order constraints

More often, DGs specify word order separately from dependencies.
- e.g., \(\text{Seq(det}(w), \text{adj}(w), \text{n}(w))\):
- determiner precedes adjective which precedes noun

Important concept in DG is \textit{valency}: ability of a word to take arguments.
Possible lexicon fragment:

\[
\begin{array}{c}
\text{sink}_1 \\
\text{sink}_2 \\
\text{give} \\
\end{array}
\begin{array}{c}
\text{Slot}_1 \\
\text{Slot}_2 \\
\text{Slot}_3 \\
\end{array}
\begin{array}{c}
\text{ACT}(\text{nom}) \\
\text{PAT}(\text{nom}) \\
\text{PAT(\text{acc})} \\
\text{ADDR(\text{dat})} \\
\end{array}
\]

To determine grammaticality:
- Words have valency requirements that must be satisfied.
- Constraints apply to the valencies to see if a sentence is valid.

Example lexical entry from a different framework [Duchier(1999)]:

\[
\begin{array}{ccc}
\text{string} & \text{loves} \\
\text{cat} & V \\
\text{agr} & \langle \text{sing 3 nom} \rangle \\
\text{comps} & \{\text{subject, object}\} \\
\end{array}
\]

Grammaticality

Using layers of dependencies

Different frameworks allow for differing layers of dependencies.
- e.g., FGD distinguishes tectogrammatical & analytical layers.

Example from MTT [Melčuk(1988)]:
Mutual dependence: verb selects subject (& other arguments), but verb form depends on the subject:

(4)

a. The child is playing.
   b. The children are playing.

One solution:
- Dependence of \textit{child/children} on the verb is syntactic.
- Dependence of the verb(form) on the subject is morphological.

Likewise, \textit{clean} could depend both on the verb \textit{wash} & on the noun \textit{dish}:

(5) Wash the dish \textit{clean}.

One solution:
- Dependence of \textit{clean} on \textit{wash} is syntactic (cf. case).
- Dependence of \textit{clean} on \textit{dish} is semantic (cf. gender).

(6) My \textit{našli} \textit{zal} pust-ym.
We found the hall\textit{masc} empty\textit{masc.sg.inst}
## Advantages and Disadvantages of DG

### Advantages:
- Close connection to semantic representation
- More flexible structure for, e.g., non-constituent coordination
- Easier to capture some typological regularities
- Large body of computational work on dependency parsing

### Disadvantages:
- No constituents makes analyzing coordination difficult
- No distinction between modifying a constituent vs. an individual word
- Harder to capture things like, e.g., subject-object asymmetries

---

### References

- **Järvinen, Timo and Pasi Tapanainen (1988).** Towards an Implementable Dependency Grammar.

---

### In Sylvain Kahane and Alain Polguère (eds.), Proceedings of the Workshop on Processing of Dependency-Based Grammars. pp. 1–10.
- **Tapanainen, Pasi and Timo Järvinen (1997).** A non-projective dependency parser.