Dependency Grammar

Lilja Øvrelid
INF5830
Fall 2015

With thanks to Markus Dickinson, Sandra Kübler and Joakim Nivre

Overview

- INF5830 so far
  - general methodology
  - statistical, data-driven approaches
  - words, frequencies
- The next four (or so) weeks
  - theoretical background and practical experience with two NLP tasks
  - “deeper processing”: syntactic and semantic analysis
    - data-driven dependency parsing
    - semantic role labeling (SRL)
  - experimental methodology
  - supervised machine learning
  - classification
  - evaluation
Part I: Data-driven dependency parsing

- Dependency grammar (today)
- Dependency parsing (next week)
- Project A released next week
- Experimental methodology (next Thursday)
- Project A (written report due Oct. 23rd):
  - training and evaluation of parsers for several languages
  - MaltParser: freely available software for data-driven dependency parsing

Part II: Semantic Role Labeling (SRL)

- Semantic roles, theoretical (Oct 19th)
- Semantic role labeling, practical/computational (Oct 26th)
- Project B (written report due Nov. 6th)
  - CoNLL 2008: syntactic and semantic parsing of English
  - solve part of this task: semantic argument classification
  - feature engineering (using syntactic analysis)
  - supervised machine learning
  - evaluation
Lectures and groups

- Curriculum: largely research literature
  - “Classics” from linguistics, e.g. Zwicky
  - Computational linguistics research literature, e.g. Nivre, Gildea & Jurafsky
- Lectures: introduction to topics, synthesis of curriculum
- Group teaching:
  - focused reading groups (please prepare!!)
  - practical sessions related to obligatory assignments

Today’s lecture

- Very brief repetition of basic principles of syntax:
  - form vs function
  - constituents and phrases
  - context-free grammars
- Dependency Grammar
  - basic concepts: head, dependent
  - comparison to constituent structure
  - theoretical issues: head criteria, tricky constructions
  - formal properties
Syntax

▷ Syntax: study of the structure of sentences
▷ “Who does what to whom?”
▷ Wealth of theories: some differences, a lot in common
  ▷ Government and Binding (GB)
  ▷ Minimalist Program (MP)
  ▷ Head-driven phrase structure grammar (HPSG)
  ▷ Lexical Functional Grammar (LFG)
  ▷ Categorial Grammar
  ▷ Dependency Grammar
  ▷ ...

Why bother?

▷ Theoretical syntacticians concerned with grammaticality
  ▷ *President the new Supreme Court justice
  ▷ The President nominated a new Supreme Court justice
▷ Relevant for some NLP applications:
  ▷ text generation
  ▷ grammar checking
▷ But mostly want systems that are robust and can handle realistic (noisy) language
Why bother?

- Parsing provides “scaffolding” for semantic analysis
- Direct, down-stream usage of syntactic information
  - opinion mining
  - information extraction
  - syntax-informed statistical machine translation
  - sentence compression
  - etc.

Constituents

- The words in a sentence are organized into groupings
- function as a whole
- relate to other words as a unit
  - The dog ate my homework
- linguistic tests of constituency
Form and function

- **Syntactic form** - constituents are described using parts of speech and phrases
  - phrases - larger constituents above word level
  - phrases named after the **head** - central, obligatory member
  - e.g. NP, VP, PP
- **Syntactic function** - constituents are described by their role in the sentence as a whole
  - Subject
  - (Direct and Indirect) Object
  - Adverbial

Arguments vs. adjuncts

- Subconstituents which are not heads: arguments or adjuncts
  - arguments: selected by the head and complete the meaning
  - adjuncts: not selected by the head and refine the meaning
- Different PoS may take argument(s):
  - John *invited* Mary to the event
  - John’s *invitation* of Mary to the event caused quite a stir
  - Mary found the book *under* the couch
- Adjuncts are not obligatory and may often iterate
  - John ran on Sunday / with Mary / in the park
Phrase structure grammars

- Capture constituent status and ordering
- Formal model: context-free grammar
  1. \( S \rightarrow NP \ VP \)
  2. \( NP \rightarrow D \ N \)
  3. \( VP \rightarrow V \ NP \)
- Syntactic structure as phrase structure \texttt{trees}

Now: Dependency Grammar

- An alternative to phrase structure representations
- Syntactic functions are central
- Claimed to be closer to semantic analysis

\begin{dependency}
    obj
    | nmod \\
    | sbj \\
    | \\
    Small birds sing loud songs
\end{dependency}
Dependency Grammar

- 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
- We will base a lot of our discussion on [Mel’čuk 1988]

Dependency grammar is important for those interested in CL:
- Increasing interest in dependency-based approaches to syntactic parsing in recent years (e.g., CoNLL-X shared task, 2006)
- Downstream applications: machine translation, question answering, ontology learning, sentiment analysis, etc.

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:

```
S
  NP
    Small
    birds
  VP
    sing
    NP
      loud
      songs
```

**Overview: dependency**

The corresponding dependency tree representations [Hudson 2000]:

```
obj
  nmod
  nmod
  Small
  birds
  sing
  loud
  songs
```
```
Constituency vs. Relations

- DG is based on relationships between words, i.e., dependency relations
  - A $\rightarrow$ 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
  - These relations are generally things like subject, object/complement, (pre-/post-)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

Simple relation example

For the sentence John loves Mary, we have the relations:

- loves $\rightarrow$ subj John
- loves $\rightarrow$ 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, then, consists of the set of pairwise relations among words.
Economic news had little effect on financial markets.

**Terminology**

<table>
<thead>
<tr>
<th>Superior</th>
<th>Inferior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>Dependent</td>
</tr>
<tr>
<td>Governor</td>
<td>Modifier</td>
</tr>
<tr>
<td>Regent</td>
<td>Subordinate</td>
</tr>
</tbody>
</table>

Dependency Grammar 21(37)

Dependency Grammar 22(37)
Comparison

- 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.

Some Theoretical Issues

- 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?
Syntactic heads

- Central concept in syntactic theory
- Seminal paper *Heads* [Zwicky 1985]:
  - The intuition to be captured with the notion HEAD is that in certain syntactic constructs one constituent in some sense 'characterizes' or 'dominates' the whole
    1. Det + N *those penguins*
    2. V + NP *control those penguins*
    3. Aux + VP *must control those penguins*
    4. P + NP *toward those penguins*
    5. NP + VP *we control those penguins*
    6. Comp + S *that we control those penguins*
- semantic argument ("kind of thing"), subcategorisand (lexical property), morphosyntactic locus (bearer of inflection)

Criteria for Heads and Dependents

- Criteria for a syntactic relation between a head $H$ and a dependent $D$ in a construction $C$ [Zwicky 1985, 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
Criteria for Heads and Dependents

- Endocentric constructions: dependents are optional
  - *Economic news had little effect on* [financial] *markets*
- Exocentric constructions: head cannot readily replace the whole
  - *Economic news had little effect on financial* [markets]
- head-complement relations are exocentric
- head-modifier relations are endocentric

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></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>

```
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

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 \((i, j \in V)\):
  - \( i \to j \equiv (i, j) \in E \)
Formal Properties of Dependency Graphs

- **antisymmetric**: if $A \rightarrow B$, then $B \not\rightarrow A$
  - If $A$ governs $B$, $B$ does not govern $A$
  - cf. *box lunch* ($\text{lunch} \rightarrow \text{box}$) vs. *lunch box* ($\text{box} \rightarrow \text{lunch}$)

- **antireflexive**: if $A \rightarrow B$, then $B \neq A$
  - No word can govern itself.

- **antitransitive**: if $A \rightarrow B$ and $B \rightarrow C$, then $A \not\rightarrow C$
  - These are *direct* dependency relations
  - cf. *a usually reliable source*: $\text{source} \rightarrow \text{reliable} \& \text{reliable} \rightarrow \text{usually}$, but $\text{source} \not\rightarrow \text{usually}$

- **labeled**: $\forall \rightarrow$, $\rightarrow$ has a label ($r$)

---

Formal Conditions on Dependency Graphs

- **Intuitions:**
  - Syntactic structure is complete (**Connectedness**).
  - Syntactic structure is hierarchical (**Acyclicity**).
  - Every word has at most one syntactic head (**Single-Head**).

- Connectedness can be enforced by adding a special root node.

```
root Economic news had little effect on financial markets
       ^   ^  ^   ^  ^    ^
      nmod  sbj    nmod nmod nmod
     ^     ^     ^     ^
    pred   obj pc
```

**Dependency Grammar**

31(37)
Formal Conditions on Dependency Graphs

- **G** is (weakly) connected:
  - For every node \( i \) there is a node \( j \) such that \( i \to j \) or \( j \to i \).

- **G** is acyclic:
  - If \( i \to j \) then not \( j \to *i \).

- **G** obeys the single-head constraint:
  - If \( i \to j \), then not \( k \to j \), for any \( k \neq i \).

Projectivity

Projectivity

- A head (A) and a dependent (B) must be adjacent: A is adjacent to B provided that every word between A and B is a subordinate of A.

- A projective graph: If \( i \to j \) then \( i \to *k \), for any \( k \) such that \( i < k < j \) or \( j < k < i \)

(2) with great difficulty

(3) *great with difficulty

- with \( \to \) difficulty
- difficulty \( \to \) great

*great with difficulty* is ruled out because branches would have to cross in that case.
Projectivity

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

```
What       did     economic    news     have     little    effect     on     ?
  pc       p       vg       subj     nmod     nmod     nmod
```

Advantages and Disadvantages of DG

Advantages:
- Close connection to semantic representation
- Easier to capture some typological regularities
- Vast & expanding body of computational work on dependency parsing

Disadvantages:
- No constituents makes analyzing coordination difficult
- No distinction between modifying a constituent vs. an individual word
Thursday:
- Reading group, in-class exercises
- Zwicky and Nivre
- Please prepare!

Monday 5th:
- Dependency Parsing

References
  *English Word Grammar*. Blackwell.
  Dependency grammar course notes. 
- Lucien Tesnière. 1959. 
  *Éléments de syntaxe structurale*. Editions Klincksieck.