Dependencies & Constituencies

L715: Seminar on: Data manipulation for parser improvement

Dept. of Linguistics, Indiana University
Fall 2011

Introduction

Before looking at dependency & constituency conversions (among other conversions), we want to explore these notions

- Theoretical syntax in the US (and on English) has largely been focused on constituency-based models
- At the same time, many theoretical traditions (especially studying freer order languages) rooted in dependencies

For parsing, 1990s-2000s focused on constituency parsing

- In the last 5-10 years, dependency parsing has taken off, due to its efficiency and connection to semantic roles
- See Cer et al. (2010) for some discussion on the advantages of each type of parsing

What is a dependency?

Gerdes and Kahane (2011)

A (labeled) binary relation between two items (normally, two words)

Dependencies have been defined:

- in terms of acceptable two-word fragments
- in terms of "directed co-occurrence"
- in terms of constituencies (with heads)
  - Tests for headedness involve replaceability, optionality, positioning, etc. (Zwicky 1985; Hudson 1990)

One must be clear what the nature of the dependencies are: morphological, syntactic, semantic? (cf., e.g., Mel'čuk 1988)

What is a constituent?

Constituencies are chunks of items (again, words), capturing some notion of regularity

- "the only goal being the description of possible orders with few rules" (Gerdes and Kahane 2011, p. 19)

There are also tests for constituency—stand-alone, movement, coordination, etc.

The notion of a constituent often involves some notion of a head (e.g., X-bar theory)

Multiple constituencies?

Given different criteria, Gerdes and Kahane (2011) suggest looking at multiple constituencies in a sentence, e.g.,

- She waits [on us]
- She [waits on] us

This leads them to derive dependencies and constituences from multiple fragmentations

- bears some similarity to data-oriented parsing (DOP) (Bod et al. 2003)

What are the (actual) differences?

Rambow (2010)

Annotation is the result of a scientific activity

- assigning abstract linguistic classes and structures as determined by someform of linguistic theory

Three details to keep in mind:

1. Representation type: mathematical object (e.g., dependency trees and phrase structure trees)
2. Syntactic content: linguistic (morphological & syntactic) facts of the analysis
3. Syntactic theory: how the syntactic content is represented in the chosen formalism

Representation Types

- **Dependency Tree (DT)**: "a tree in which all nodes are labeled with words ... or empty strings."
- **Phrase Structure Tree (PST)**: "a tree in which all and only the leaf nodes are labeled with words or empty strings, and internal nodes are labeled with nonterminal strings"

Rambow (2010) stresses that these are the *only* parts to the definitions

Representation Types

Both DTs and PSTs can vary in the following:

- Unordered trees
  - DT more likely to be unordered, sometimes only at a "deep" level (but PSTs, too)
- Empty categories
  - PSTs more likely to have empty strings (but DTs, too)
- Discontinuous Constituents or Non-projectivity
  - DTs more likely (but PSTs, too)
- Labeled Arcs
  - DTs more likely to be labeled (but PSTs, too)

Syntactic Content

Syntactic content is the "*empirical matter* which linguistic theory attempts to represent or explain"

In terms of syntactic content:

- **Syntactic dependencies**: relation between words, generally subsuming grammatical functions
  - e.g., subject, object, temporal-adjunct
- **Syntactic phrase structure**: "recursive representation using sets of one or more linguistic units ... such that at each level, each set (constituent) acts as a unit syntactically"

The syntactic content of *either* can be expressed via DTs or PSTs . . .

Syntactic Content (2)

Consider the Penn Treebank (PTB) encoding of predicate-argument structure

- PTB uses PSTs, but encodes grammatical functions, e.g., NP-SBJ
- This thus encodes both syntactic phrase structure and dependencies

Consider converting DTs to PSTs:

- Each node is both a preterminal (X^0) a node covering a constituent that it is the head of (XP)
  - The XP contains "all terminals included in the subtree it [X] heads"
  - Intermediate projections can be encoded via features and arc labels

Syntactic Theory

"The choice of representation type does not determine the representation for a given sentence."

- Many possible syntactic theories

Syntactic theory: "defines a coherent mapping for a well-defined set of content to the chosen representation type"

- Choices are independent
- Can omit some syntactic content, e.g., PTB does not distinguish raising and control

"*every* representation needs a theory for us to extract its meaning!"

Conversions

Converting between DTs and PSTs means that we want the same content

- We can only convert information which is present in one treebank into the other
  - May have to use heuristics to recover semi-latent information
- Converting between representations expressing the same content may be challenging
  - Treebank representations will often be based on different theoretical traditions

Form-function relations

Tsarfaty and Sima'an (2008)

For morphologically-rich languages, there is a need to factor out configuration from parsing models

- "the position of a constituent is a *form* manifestation of its grammatical *function*"
- The goal is to allow for better:
  - phrase-structure variability
  - morphological-syntactic interaction

Idea: separate generation into phases:

1. *Relational* phase: build a relational network among categories (cf. dependencies)
2. *Realizational* phase: "projected relations are realized in a certain surface configuration" (cf. constituencies)

For our purposes, we are interested in what this illustrates about the roles of constituencies & dependencies

Structural modeling

Constituent parsers parameterize in ways which generalize over structure/configurations

- e.g., PCFGs model configurations
- May not work as well for less-configurational relations

Even for dependency parsing, relatively free word-order languages do worse

Background on modeling form & function

Head-driven PCFG parsing (Collins, Charniak) features nodes enriched with head information

- Allows for better modeling of dependencies
- However, there is a configurational bias:
  - encodes the position of a non-head relative to a head
  - implicit assumption: "the relation between the position of a constituent and its grammatical function is fully predictable."
- grammatical functions are secondary notions, derived from constituencies

But: properties of a constituent "may emerge from different surface forms dominated by it"

Relational Grammar

Relational grammars (RGs) took grammatical relations (e.g., Subject, Object) as grammatical primitives

Primitive elements:

- set of nodes representing linguistic elements
- set of names of grammatical relations ($gr_1, ...gr_n$)

One can then define:

- Arcs: $gr_i(A, B)$
- Relational Network (RN): set of arcs sharing a single head

Distinctions of RGs:

- RNs use clause-level categories—instead of only surface forms
- Auxiliary verbs & particles do not receive arcs
- RNs are unordered

Separation

"Separation Hypothesis" (Beard 1988):

- 'form' (e.g., a particular morpheme) and 'function' do not have to be in a one-to-one correspondence
  - "The problem of modeling certain surface phenomena then boils down to modeling [potentially complex] form and function correlations"

Viewpoint:

- *form*: position of a constituent in a phrase
- *function*: articulated grammatical function

Task: statistically learn how positions of constituents relate to grammaticl relations

- "Configuration" parameters determine which syntactic position grammatical relations are realized in

Morphosyntactic representations

View of tree-internal nodes: *Morphosyntactic Paradigms*

- Structured representation of morphological & syntactic properties
- Morphological features are percolated from surface forms
  - specify head (POS) information
  - specify structural features, e.g., vertical markovization

"Given the grammatical relation an element bears, it is statistically feasible to learn the morphosyntactic properties by which it is realized."

Three levels of the generative model

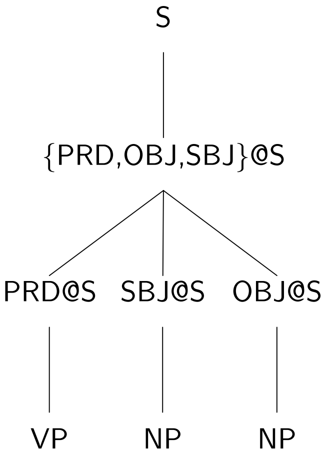
Let $S_p \rightarrow \langle S_{c_1}\text{-}gr_1 \ldots S_{c_n}\text{-}gr_n \rangle$ be a CF rule

- S_p = morphosyntactic representation of parent constituent
- $gr_1 \ldots gr_n$ = grammatical relations forming RN
- $\langle S_{c_1} \ldots S_{c_n} \rangle$ = morphosyntactic represenations of child constituents

Grammar generates a rule in 3 phases:

- **Projection** (generate set of grammatical relations): $S_p \rightarrow \{gr_i\}_{i=1}^{n} @ S_p$
- **Configuration** (order the grammatical relations): $\{gr_i\}_{i=1}^{n} @ S_p \rightarrow \langle gr_1 @ S_p \ldots gr_n @ S_p \rangle$
- **Realization** (generate morphosyntactic representations) $\{gr_i @ S_p \rightarrow S_{c_i}\}_{i=1}^{n}$

Example tree



Three levels of the generative model (2)

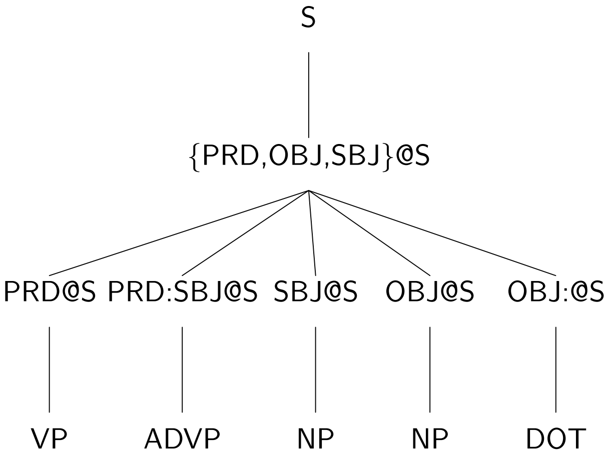
- Functional elements (auxiliarys, particles) are not generated
- Modifiers & periphrastics are allowed in configuration phase in-between other elements

Altered grammar:

- **Projection:**
  $S_p \rightarrow \{gr_i\}_{i=1}^{n} @ S_p$
- **Configuration:**
  $\{gr_i\}_{i=1}^{n} @ S_p \rightarrow \langle gr_0 : gr_1 @ S_p ... gr_n : gr_{n+1} @ S_p \rangle$
- **Realization:**
  $\{gr_i @ S_p \rightarrow S_{c_i}\}_{i=1}^{n}$
  $\{gr_i : gr_{i+1} @ S_p \rightarrow \langle S_{c_{i_1}} ... S_{c_{i_{m_i}}}\}_{i=1}^{n}$

Also: modifications can have more than one constituent in realization phase

Example tree



Details, details ...

See the paper for more details on

- interaction with morphological marking
  - Realizations reflect grammatical function and not configuration, allowing morphology to correlate with GRs, not linear position
- treatment of conjunction
- the probabilistic model
- the grammar

Experimental results

- Models include ones where morphological representation includes **Def**initeness and **Acc**usativity, in addition to the **Base**
- Likewise, *Head*, *Parent*, and *ParentHead* models are used to enrich representations

Consider subset of their Table 1 results (**BaseDefAcc**/*ParentHead* model):

PCFG: 72.77/73.01

RR-PCFG: 76.15/76.43

Morphological information is better incorporated into RR models

Usefulness of separation

"trends ... suggest that the *RR*-models are more powerful in exploiting different sources of information"

- Captures generalizations based on grammatical function
- Allows for better integration of morphological features

For our purposes, it is important to note:

- how form & function are separate grammatical components
- how both can be modeled in correlated, but distinct ways

References

Bod, Rens, Remko Scha and Khalil Sima'an (eds.) (2003). *Data-Oriented Parsing*. University of Chicago Press.

Cer, Daniel, Marie-Catherine de Marneffe, Daniel Jurafsky and Christopher D. Manning (2010). Parsing to Stanford Dependencies: Trade-offs between speed and accuracy. In *Proceedings of LREC-10*. Malta.

Gerdes, Kim and Sylvain Kahane (2011). Defining dependencies (and constituents). In *Proceedings of the International Conference on Dependency Linguistics*. Barcelona, Spain, pp. 17–27.

Hudson, Richard A. (1990). *English Word Grammar*. Blackwell.

Mel'čuk, Igor (1988). *Dependency Syntax: Theory and Practice*. State University of New York Press.

Rambow, Owen (2010). The Simple Truth about Dependency and Phrase Structure Representations: An Opinion Piece. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 337–340.

Tsarfaty, Reut and Khalil Sima'an (2008). Relational-Realizational Parsing. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, pp. 889–896.

Zwicky, A. M. (1985). Heads. *Journal of Linguistics* 21, 1–29.