Slot Grammar

Zahra Solgi

Universität Tübingen

June 18, 2016
Slot Grammar

Overview

Slot Grammar?
Slot Grammar?

what is the use of that?
Slot Grammar

Overview

Slot Grammar?

what is the use of that?

what are ESG, FSG, SSG, ISG, BPSG, GSG?
Slot Grammar

- SG: provides a convenient means for writing practical, broad-coverage grammars for natural language
Slot Grammar

Overview

- SG: provides a convenient means for writing practical, broad-coverage grammars for natural language

- An SG is a *Dependency tree* which reduces the differences between grammars of different languages
Using Slot Grammar

- the input is segmented by a "Sentences" separator
- the output of SG analysis is a pars tree
Using Slot Grammar

overview

comparison between Dependency grammar and constituency grammar

Phrase structure grammar

b. They killed the man with a gun.

Dependency grammar
Dependency grammar vs. constituency grammar

```
We are trying to understand the difference.
```

Dependency

```
We are trying to understand the difference.
```

Constituency (BPS)
Deep parsing

- Surface structure is determined by modifier structure
- Deep structure is determined by logical argument frames
Using Slot Grammar
overview

- subj → subject
- obj → direct object
- iobj → indirect object
- comp → predicate complement
- objprep → object of preposition
- ndet → NP determiner
- :
ESG parse tree: Chandeliers look great but nowadays do not usually use these items from which their name is derived.
Using Slot Grammar

Complement slots – determined by the properties of their headword (e.g. verb which take subjects and objects)

Adjunct slots – determined by the part of speech of their headword (e.g. verbs can take an adverb)
parse Nodes

- Headword
- ID
- Logical argument frame
- Features
- Modifier structure

```
comp(p)          from(12,17,13)          prep wh
  objprep(n)      which(13,11,u)      noun pron wh
   ndet           their(14)           det sg possdet
   subj(n)        name(15,u,u)        noun cn sg langunit
   nrel            be(16,15,17)      verb vfin vpres sg
data(17,u,15,12)  derive(17,u,15,12) verb ven vpass
```
SG lexical entries

- Morpholexical Analysis: how to determine frames?
  - Look up word in the provided SG Lexicon and match its use in context to a sense frame specified in the Lexicon

\[
talk < v \text{(obj n (p about)) (comp (p to with))} \\
< v \text{obj1 (compl (p into))} \\
< n \text{nsubj (nobj n (p about))} \\
\text{(ncomp (p to with))}
\]

- Lexical entries
  - Part of speech – e.g. noun, verb, adjective, etc.
  - Complement slot frame
  - Features – syntactic features or semantic types e.g. object, property, event, living being
  - Numerical score – rate sense frames
  - Subject area – e.g. computers, medicine
  - Support verb construction
IBM improvement on SG lexicon

- Match noun frames with verb frames
  - E.g. encode a relationship between “celebration and “celebrate”
  - Helps match questions to answers
- Augmentation of ESG base lexicon (using WordNet)
  - Increases number of entries
  - Indicates semantic types
- Noun-verb correspondences
  - E.g. verb defer has indicated noun-forms deferral, deference, etc.
- Chunk Lexicons
  - Handle multiword entries (e.g. “Sing a Song of Sixpence”).
- LAT Reward Features
  - Aid in identification of answer types
syntactic analysis

1. Combine tokens into syntactic constituents
2. Bottom-up, left-right organization of constituents into slots
3. Subtrees build phrases
4. Phrases are scored according to lexical use of constituents, rules in grammar
Predicate argument

- Simplifies and generalizes result of ESG parse
  1. Elements change exact semantic meaning but in general are not essential to its core meaning.
  2. Does not process original text. Instead modifies the output of the ESG parse.

- I heard that Edison invented the phonograph in 1877.
- I heard that Edison invented a phonograph in 1877.
- I heard — Edison invented the phonograph in 1877.
- I heard that Edison was inventing the phonograph in 1877.
- I heard that the phonograph was invented by Edison in 1877.

- E.g, Have different meanings,, generate different ESG parse trees
  1. Exact semantic meaning irrelevant since they all contain the same evidence to answer a question “Who invented the phonograph?”
Slot filling for *Mary gave John a book*

give \((e, x, y, z)\) means \(e\) is an event where \(x\) gives \(y\) to \(z\)
Logical Representation

analysis

Slot filling for *Mary gave a book to John*
Ingredients of Slot Grammar analysis Structure
Extrposition

phrase (X, H, Sense, Features, SlotFrame, Ext, Modes)

- the argument Ext is used to hold Extraposed slots
  - i.e: slots that can be filled by left-extraposed phrases like who in who did Alicetry to find.
- The list Ext consists of internal form and one element

Ingredients of shells:
- a declaration of extraposer slots
- extraposed filler rules

Extraposer slots are slots that allow extrapositions out of their fillers
The following method for handling coordination was outlined in (MacCord 1980) then was implemented in the recent adaptation of Slot Grammar to logic programing.

The form of coordination in a phrase:

LM Preconj LC Conj RC RM
Coordination

The substrings indicated are interpreted as follows:

- **Conj** → coordinating conjunction (like *and* or *or*) or a punctuation like comma
- **Preconj** → optional *preconjunction* that can accompany *Conj* *(both for *and*)
- **LC and RC** → *left* and *right conjuncts* respectively. (one single phrase)
- **LM and RM** → (optional) *left* and *right common modifier*, respectively. (several phrases)
Let’s make an Example:

**The man** sees **and** **probably hears** the car.

**John sees** **and** **Mary hears** the car.
We have given an overview of ESG analysis and indicated the central role of slots and slot frames.

▶ conclusion and Future Work

1. expansion of the semantic type system and its use in parsing; 
2. incorporation of word-sense disambiguation, probably with senses of less granularity than in WordNet. 
3. indication in parse trees of scoping of generalized quantifiers and focusing adverbs, etc. 
4. development of specialized lexicons and methods for handling very large lexicons 
5. continued improvement of coverage of SG via regression testing.