Knowledge Engineering for NLP

February 2, 2008

Modification, Discourse status, Optional arguments

Overview

- Modification
- Discourse status
- Optional arguments
    - Semantic classification
    - Syntactic classification
    - Typological claims
- Analysis of optional arguments
- Precision grammars and corpus data

Modification: Syntax

- Modifiers select the heads they modify via the MOD feature (inside HEAD).
- The value of MOD is a list of *synsems*.
- Head-modifier rules are cross-classified according to order (head-adj, adj-head) and the intersective/scopal distinction.
- You might already have head-modifier rules in your grammar (probably just instances in rules.tdl which inherit directly from types in matrix.tdl).

Intersective modifiers

- Adjoined via a ‘head-compositional’ PSR (syntactic head is semantic head)
- ARG1 is MOD’s INDEX (*individual*)
- LTOP = MOD’s LTOP (constraint on rule)

Scopal modifiers

- Serve as semantic head daughters

  What does this mean in tdl?

- Identify their own INDEX with their MOD’s INDEX (why?)
- Take a handle-valued ARG1
- Insert a qeq between their ARG1 and their MOD’s LTOP (why?)

Scopal modifiers: examples

- Kim did not read every book.
- Kim probably read every book.
- The most likely winner of every medal was disqualified.

Gate keeping

- The phrase structure rules for intersective and scopal modifiers need to be different.
- Use subtypes of *local* to constrain which rule gets used.

    No other use for subtypes of *local*

    Modifiers constrain LOCAL inside their MOD value

Scopal mod phrase

```
scopal-mod-phrase := head-mod-phrase-simple &
  [ NON-HEAD-DTR.SYNSEM.LOCAL [
       CAT.HEAD.MOD < [ LOCAL scopal-mod ] >,
       CONT.HOOK #hook ],
    C-CONT [ HOOK #hook,
             HCONS <! !> ] ].
```

Intersective mod phrase

```
isect-mod-phrase := head-mod-phrase-simple &
                    head-compositional &
  [ HEAD-DTR.SYNSEM.LOCAL.CONT [
        HOOK.LTOP #hand ],
    NON-HEAD-DTR.SYNSEM.LOCAL [
        CAT.HEAD.MOD < [ LOCAL intersective-mod ] >,
CONT.HOOK.LTOP #hand ],
    C-CONT.HCONS <! !> ].
```

Modification: Your tasks

- Instantiate adj-head-int or head-adj-int rules, as necessary
- Create lexical types for adjectives and adverbs (inheriting from matrix types)
- Create lexical rules for agreeing adjectives (as necessary)
- Constrain other lexical types to not serve as modifiers

Overview

- Modification
- Discourse status
- Optional arguments
    - Semantic classification
    - Syntactic classification
    - Typological claims
- Analysis of optional arguments
- Precision grammars and corpus data

Discourse status: What’s that? (1/2)

- A property of referents, describing their relationship to the common ground of a conversation.
- Tends to be reflected syntactically in markers of ‘definiteness’ as well as demonstratives and constraints on the availability of types of NPs in particular constructions.
- Closely related to information structure:
  - Classification parts of a sentence into topic and comment
  - Sentential focus

Discourse status: What’s that? (2/2)

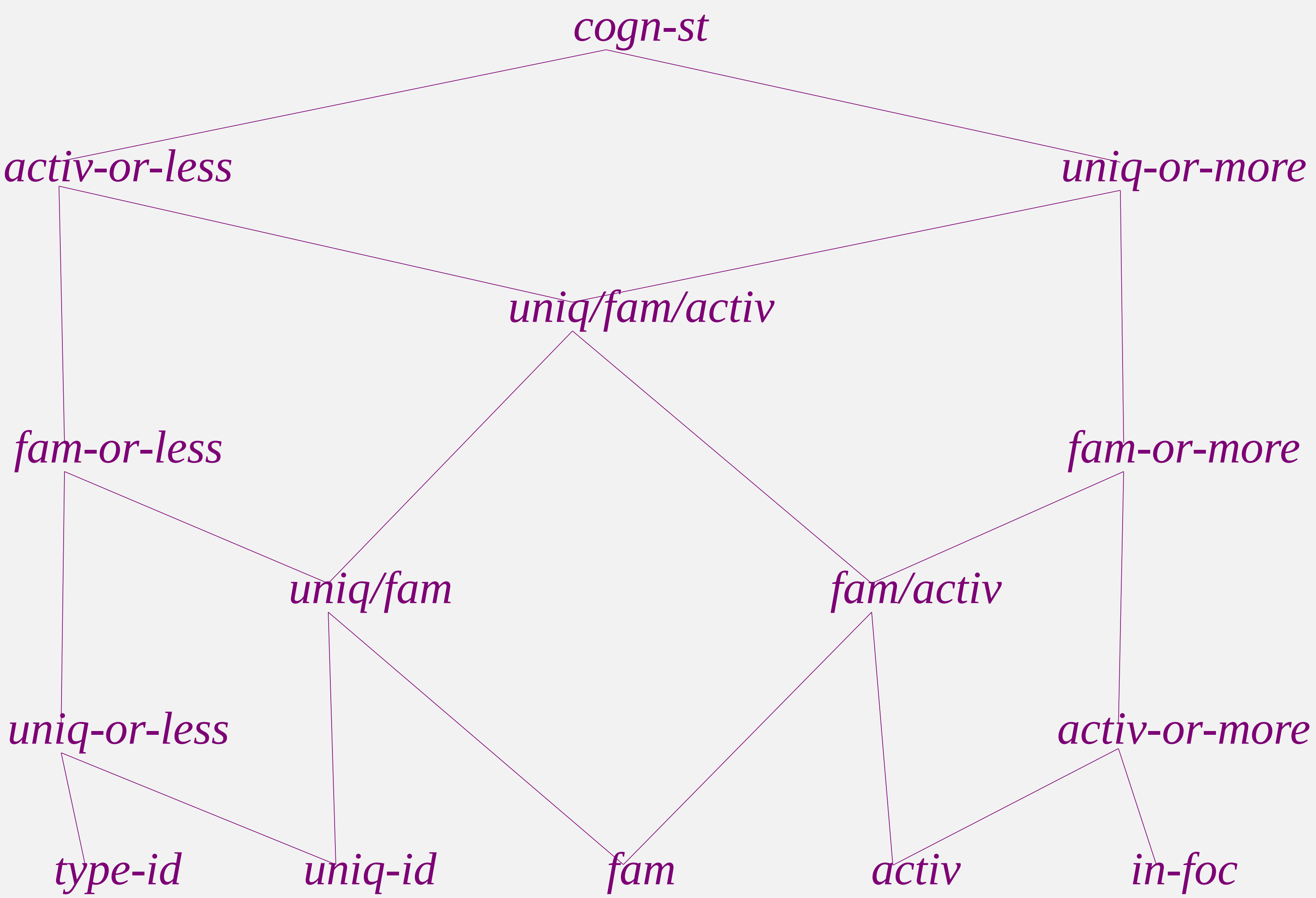
- The binary disctinction “definite/indefinite” is not sufficient to capture this.
- Furthermore, discourse status can be broken down into hearer-oriented “cognitive status” and speaker-oriented “specificity.”

Givenness hierarchy

(Gundel et al 1993, Prince 1981)

Type id <	Referential <	Uniq. id. <	Familiar <	Activated <	In focus
a N	indefinite	*the N*	*that N*	*that, this*	*it*
	this N			*this N*	

Borthen & Haugereid's proposal (1/3)



Borthen & Haugereid's proposal (2/3)

$$\left[\text{SS.LOC.CONT.REF-PROP} \quad \left[\begin{array}{ll} \textit{ref-prop} & \\ \text{INDEX} & \left[\begin{array}{ll} \textit{ref} & \\ \text{PER} & \textit{per} \\ \text{NUM} & \textit{num} \\ \text{GEND} & \textit{gend} \end{array}\right] \\ \text{COGN-ST} & \textit{cogn-st} \\ \text{SPECI} & \textit{bool} \\ \text{PART} & \textit{bool} \\ \text{UNIV} & \textit{bool} \end{array}\right]\right]$$

Borthen & Haugereid's proposal (3/3)

- SPECI indicates specificity (speaker-oriented)
- Compatible with both "definite" and "indefinite" NPs:
  - *The best student won.*
  - *The next customer will receive a reward.*
- Corresponds to overt syntactic phenomena in at least Norwegian (specificity adjectives) and Turkish (accusative case precludes specific interpretation).

First-pass Matrix-based proposal

$$\begin{bmatrix} \text{HOOK.INDEX} & \begin{bmatrix} \text{PNG} & \begin{bmatrix} \text{PER} & \textit{person} \\ \text{NUM} & \textit{number} \\ \text{GEND} & \textit{gender} \end{bmatrix} \\ \text{COG-ST} & \textit{cog-st} \\ \text{SPECI} & \textit{bool} \end{bmatrix} \end{bmatrix}$$

Optional arguments

- There are many cases in which an argument may be semantically present but syntactically absent.
- Semantically, these cases can be categorized by how the missing argument is interpreted.
- Syntactically, these cases can be categorized by how the missing argument is licensed.

Semantic classification

- Indefinite null instantiation: *I ate.*

  The referent of the missing argument is indefinite, not (necessarily) recoverable from context.

- Definite null instantiation: *I told you already.*

  The referent of the missing argument is definite, i.e., it should be recoverable from context.

- Constructional null instantiation: *Eat!*, *I told Kim to eat*, *Stir until completely mixed.*

  The referent of the missing argument is determined by the syntactic construction (definite, indefinite, linked to other argument in the sentence).

Syntactic classification

- Lexical: The potential for an argument to be missing is determined by the lexical type/entry of the selecting head.
  - *eat* allows indefinite null instantiation of its object
  - *devour* does not.
- Systematic: Arguments (perhaps of a certain syntactic type, such as NP or a particular grammatical function) in general can be missing.
  - Japanese-style any argument pro-drop
  - Spanish-style subject pro-drop.

Syntactic classification (2/2)

- By hypothesis, systematic pro-drop is given the definite interpretation (i.e., it corresponds to one use of overt pronouns in other languages).
- Pronoun incorporation: Verbal affixes are actually interpreted as pronouns. I would expect these cases to involve definite null instantiation.

  If the affixes are also required when the object is expressed by a full NP, this is not pronoun incorporation.

Lining up syntactic and semantic classifications

- Claim 1: A language with systematic pro-drop will allow definite interpretations of all dropped arguments.
- Claim 2: A language with systematic pro-drop will also allow indefinite interpretations of some dropped arguments, corresponding roughly to where a language without systematic pro-drop would allow indefinite null instantiation.
- Claim 3: Indefinite null instantiation of subjects involves special verb marking (e.g., impersonal passives).
- Claim 4: It follows from these hypotheses that there is no need for lexically licensed definite null instantiation in languages with Japanese-style pro-drop.

Example (Japanese)

Tabeta

Ate

‘I/you/he... ate.’/‘I/you/he... ate it.’

- Japanese has systematic pro-drop of all arguments.
- It also appears to have lexically licensed INI.
- Thus *Tabeta* is ambiguous, and we would like to be able to translate it into two different English strings.
- Nonetheless, it would be nice to avoid assigning two different tree structures, and rather provide an underspecified semantic representation.

Proposed analysis in the Matrix: Overview (1/2)

- Constructional null instantiation covered by analysis of imperatives, raising, etc.
- Distinction between definite and indefinite null instantiation handled by a feature on indices representing definiteness.
  - Pronouns and arguments subject to DNI are [COG-ST in-foc & [SPECI +]].
  - Arguments subject to INI (and possibly indefinite NPs) are [COG-ST type-id & [SPECI –]].
- Caveat: There are also interesting questions about whether we want quantifiers associated with these positions, but that's for a later time...

Proposed analysis in the Matrix: Overview (2/2)

- Posit opt-comp and opt-subj rules parallel to the bare-np rules.
- Use a feature [OPT bool] to code lexically licensed null instantiation (leaving it underspecified in languages where there is systematic pro-drop).
- Use a second feature [OPT-CS cog-st] to allow lexical items to specify whether any given optional argument would be interpreted as definite or indefinite in case of null instantiation.

The feature OPT

- OPT and OPT-CS will both be features of *synsems*.
- However, nothing constrains its own OPT value (that is, no phrases are inherently optional or non-optional, independent of which head they are dependent on).
- Rather, heads constrain certain arguments to be [OPT –], which blocks the optional complement/subject rules from applying, since these look for argument which are (compatible with) [OPT +].

The feature OPT-CS (1/2)

- OPT-CS is a ‘junk slot’ to allow a lexical head to store information about how an argument will be interpreted if it is unexpressed.
- The opt-comp rule will identify the OPT-CS and HOOK.INDEX.COG-ST values of any argument it caches out as unrealized.

The feature OPT-CS (2/2)

- Because the HOOK.INDEX of every argument is identified with some ARGn position in the head’s key relation, this information will be encoded in the semantics.
- Note that we’re not positing pronoun relations or associated quantifier relations for these dropped objects. This point is debatable, especially if your language appears to have incorporated pronouns.

The Matrix opt-comp type

```
basic-head-opt-comp-phrase := head-valence-phrase & head-only &
                              head-compositional &
  [ INFLECTED #infl,
    SYNSEM canonical-synsem &
    [ ..CAT [ VAL [ SUBJ #subj, COMPS #comps, SPR #spr, SPEC #spec ],
              MC #mc, POSTHEAD #ph ],
      MODIFIED #mod ],
    HEAD-DTR [ INFLECTED #infl & +,
               ..CAT [ VAL [ SUBJ #subj, SPR #spr, SPEC #spec,
                             COMPS < unexpressed &
                                   [ OPT +, OPT-CS #def,
                                     ..INDEX.COG-ST #def ] . #comps >],
                                     MC #mc, POSTHEAD #ph ],
               ..CONT.HOOK.INDEX event,
               MODIFIED #mod ],
    C-CONT [ RELS <! !>, HCONS <! !> ] ].
```

For a language with systematic pro-drop

- Allow definite null instantiation (pro-drop) everywhere.
- Also allow indefinite null instantiation if lexically specified.
- Same head-opt-comp-rule
- Two types of lexical entry:
    - Those that allow both INI and DNI leave OPT-CS undespecified
    - Those that only allow DNI specify [OPT-CS activ-or-more]

Discourse status: Your tasks

- Instantiate opt-subj and opt-comp rules
- Create verb subtypes for lexically constrained argument optionality
- Possibly modify agreement lexical rules (in case of optional agreement correlated with possibility of pro-drop)

  ... this is “incorporated pronouns”
- Add lexical types and lexical entries for demonstrative adjectives or determiners

Overview

- Modification
- Discourse status
- Optional arguments
  - Semantic classification
  - Syntactic classification
  - Typological claims
- Analysis of optional arguments
- Precision grammars and corpus data

Precision Grammars and Corpus Data

- Theoretical motivation
- Methodology
- Results
- Your grammars
- Precision grammars and NLP

Theoretical motivation (1/2)

- Corpora as a sole source of data are inadequate because:

  They are limited in size and may not reflect the full range of grammatical constructions.

  They contain errors due to processing and reflect other extragrammatical factors.

  They can only provide positive (attested) examples, and not contrasting negative ones.

Theoretical motivation (2/2)

- Intuitions as data are inadequate because:

  Grammaticality is neither homogeneous nor categorical.

  Grammaticality judgments are frequently formed in unnatural context vacuums.

  Social/cultural biases color judgments.

  Relying solely on intuitions limits linguists to only the data they have the imagination to think up.

Combine the two types of data for better results!

- Grammar engineering provides a sophisticated way of doing so.
- Precision grammars encode a sharp notion of grammaticality.
- Use grammar as a representation of intuitions.
- Use the corpus as a source of further data to explore.
- Process the corpus with the grammar...

Methodology

- Randomly select 20,000 strings (‘sentence tokens’) from the BNC written component.
- Strip punctuation, tag for part-of-speech, tokenize proper names and number expressions, normalize to American spelling.
- Select those strings with full lexical span (32%).
- Process these strings with the ERG to isolate those that can’t presently be parsed.
- Use treebanking technology/methodology to validate parses.
- Propose paraphrases of the unparseable strings until the ERG is able to parse one.

Results: Grammar coverage

- 57% of strings parsed.
- 83% of parsed strings assigned a correct (preferred) parse, perhaps among others.
- Average ambiguity for 10-20 word strings: 64 parses.

Results: Causes of parse failure

Cause of parse failure	Frequency	Category
Missing lexical entry	41%	grammar
Missing construction	39%	grammar
Fragment	4%	grammar
Preprocessor error	4%	neither
Parser resource limitations	4%	neither
Ungrammatical string	6%	corpus
Extragrammatical string	2%	corpus

Missing lexical entries (1/2)

- Incomplete categorization of existing lexical items

  table as a verb

  ‘universal grinder’

- Syntactically-marked MWEs

  take off, verb + *up*

  off screen, at arm’s length

  High frequency: verb-particles constitute 1.6% of BNC word tokens

Missing lexical entries (2/2)

- Drawbacks to introspection alone: subtle gaps like transitive *suffer*

- Drawbacks to corpus data alone: *tell* in the ‘discover’ sense:

  @Not sure how you can tell.

  Can/could you tell?

  Are you able to tell?

  *They might/ought to tell.

  How might you tell?

  *How ought they to tell?

Missing constructions (1/4)

- $^{@}$*However pissed off* we might get from time to time...
- ERG specifically disallowed this.
- → Corpus data as a check on introspection.
- Further corpus investigations surprised ys.

Missing constructions (2/4)

- [@]He’s a good player and a *hell of a* nice guy, too.
- Baldwin et al present this as a semantic puzzle:

  Apparent syntactic attachment to NP/N$'$ because of definiteness restrictions

  Semantic attachment to adjective (intensifier)
- Still complex, but less mysterious, in a world where definiteness is encoded as a feature of indices.

Missing constructions (3/4)

- @The price of train tickets can vary from *the reasonable* to *the ridiculous*.
- Exocentric NPs not limited to classes of people.
- What adjectives can appear here, and with what kinds of referents?

Missing constructions (4/4)

- @This sort of response was also noted in the sample task for *criterion 2*.
- ‘Title’ (common noun) + series element
- Frequent in corpora (like dates, number names, quotatives)
- Not usually remarked on in syntactic theory

Extragrammatical strings

- Prime example: Structural markup:

  $^{@}$There are five of these general arrest conditions: (a) the name of. . .

- Preprocessing requires interface to grammar:

  $^{@}$(I) The Mrs Simpson could never be Queen.

  $^{@}$(I) rarely took notes during the thousands of informal conversational interviews.

Summary

- Methodology goes beyond merely using the corpus for inspiration.

  encoding intuitions in the grammar

  use the grammar to process the corpus, twice: filter out ‘easy’ cases, investigate where in a string the problems are

- Provides detailed feedback to grammar developers

- Turns up previously unnoted constructions, which might be too low frequency to be found otherwise

How about your grammars?

- Role of corpora so far?
- How to get from current state to something that could turn up unexpected constructions?

Precision grammars in NLP

- Baldwin et al: Notion of grammaticality cuts down on spurious ambiguity and crucial in avoiding ill-formed output in generation
- Elsewhere: Value of elaborated semantic representations
- Cost: Could grammar development ever become cheaper than treebank development?

Summary

- Theoretical motivation
- Methodology
- Results
- Your grammars
- Precision grammars and NLP