

Knowledge Engineering for NLP

February 11, 2008

Discourse status, Optional arguments

Clause types, Illocutionary force

Overview

- Discourse status
- Optional arguments
 - Semantic classification
 - Syntactic classification
 - Typological claims
- Analysis of optional arguments
- Clausal semantics
 - Why?
 - What's a clause?
 - Our strategy

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

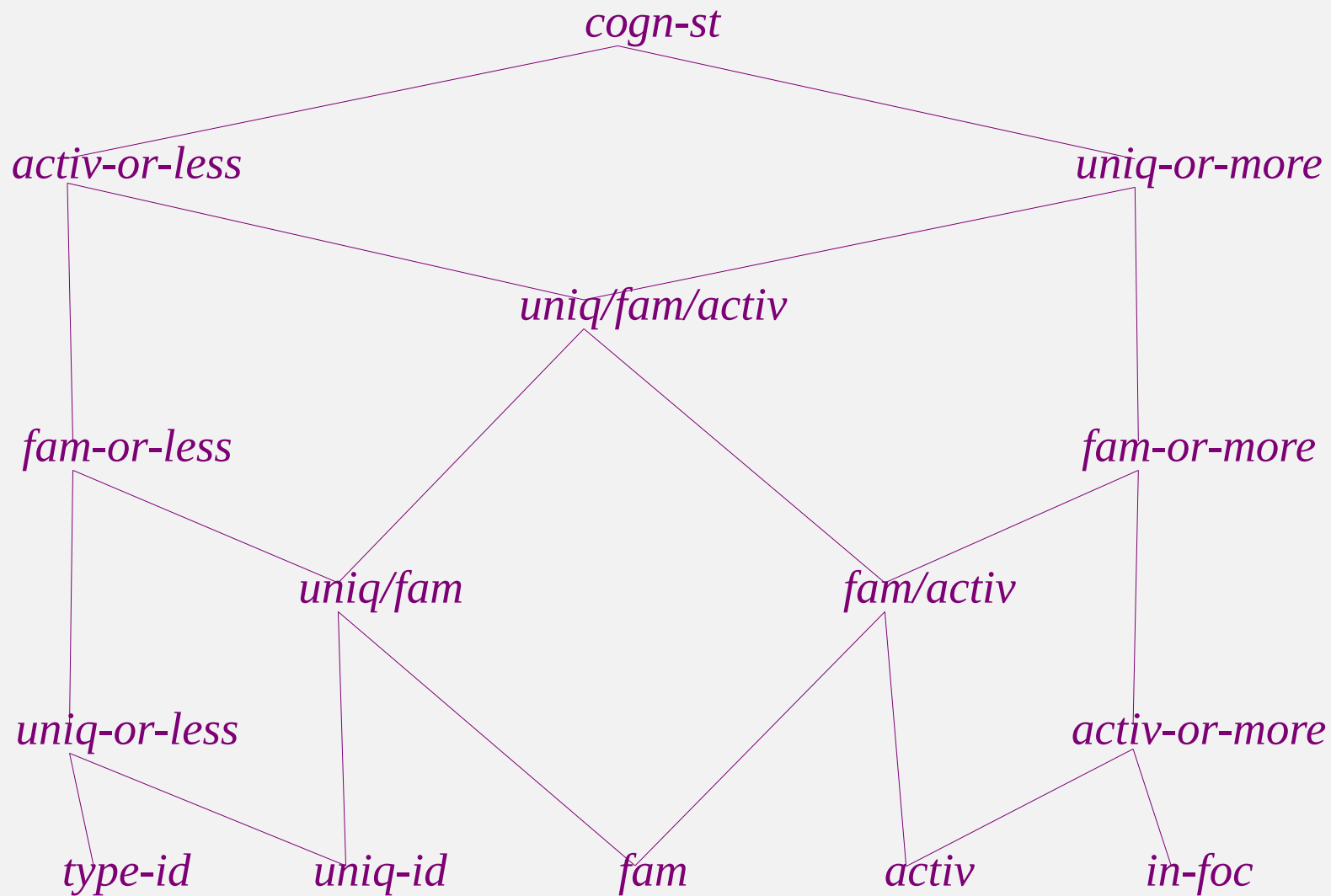
Discourse status: What's that? (2/2)

- The binary distinction “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
<i>a N</i>	indefinite <i>this N</i>	<i>the N</i>	<i>that N</i>	<i>that, this</i> <i>this N</i>	<i>it</i>

Borthen & Haugereid's proposal (1/3)



Borthen & Haugereid's proposal (2/3)

SS.LOC.CONT.REF-PROP	<i>ref-prop</i>	
	INDEX	<i>ref</i>
		PER <i>per</i>
		NUM <i>num</i>
		GEND <i>gend</i>
	COGN-ST	<i>cogn-st</i>
	SPECI	<i>bool</i>
	PART	<i>bool</i>
	UNIV	<i>bool</i>

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

$$\left[\begin{array}{c} \text{HOOK.INDEX} \end{array} \left[\begin{array}{cc} \text{PNG} & \left[\begin{array}{cc} \text{PER} & \textit{person} \\ \text{NUM} & \textit{number} \\ \text{GEND} & \textit{gender} \end{array} \right] \\ \text{COG-ST} & \textit{cog-st} \\ \text{SPECI} & \textit{bool} \end{array} \right] \right]$$

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 unspecified
 - Those that only allow DNI specify [OPT-CS activ-or-more]

Overview

- Discourse status
- Optional arguments
 - Semantic classification
 - Syntactic classification
 - Typological claims
- Analysis of optional arguments
- Clausal semantics
 - Why?
 - What's a clause?
 - Our strategy

Why clausal semantics?

- Illocutionary force correlates with syntactic form
- MRS representations should include all semantic information that is syntactically marked
- Illocutionary force is ‘predicated of’ situations — i.e., the rest of the MRS

Aside: Perlocutionary, Locutionary, Illocutionary

- locutionary act: The act of saying something
- illocutionary act: The act of asking, asserting, commanding, etc. by saying something
- perlocutionary act: The act of getting someone to do or believe something by asking, asserting, etc. something

What's a clause?

- Syntactically complete
- Expresses some illocutionary force
- Contrasts with fragments, some of which can also carry illocutionary force.
- Marking of illocutionary force is often associated with the clause as a whole or with its head verb.
- Clauses can be matrix or embedded.
- Embedded clauses carry illocutionary force, too.

Our general strategy (1/2)

- Represent illocutionary force with a feature of events, called 'SF'.
- Possible values of SF: command, prop-or-ques, proposition, question
- Non-branching rules at the top of the tree set SF depending on syntactic features. OR: Subject-attaching rules constrain SF.
- Further up the tree, complementizers, selecting heads, or unary constructions can constrain things further.

Our general strategy (2/2)

- Allow lexical items to introduce information about possible clausal semantics
 - English inverted auxiliaries are [INV +], English yes-no rule is sensitive to this.
 - English imperative clauses (a kind of head-opt-subj phrase) require [FORM base] heads.
- How do we avoid getting a parse of *Did Kim leave?* which is just [SF prop-or-ques]?

Overview

- Discourse status
- Optional arguments
 - Semantic classification
 - Syntactic classification
 - Typological claims
- Analysis of optional arguments
- Clausal semantics
 - Why?
 - What's a clause?
 - Our strategy