Knowledge Engineering for NLP

March 3, 2008

The LOGON MT architecture and Machine Translation

Overview

- MRS and MT: Some history
- The Grammar Matrix and massively multilingual MT
- The LOGON architecture
  - Processing steps
  - Anatomy of a transfer rule
  - Types and translation

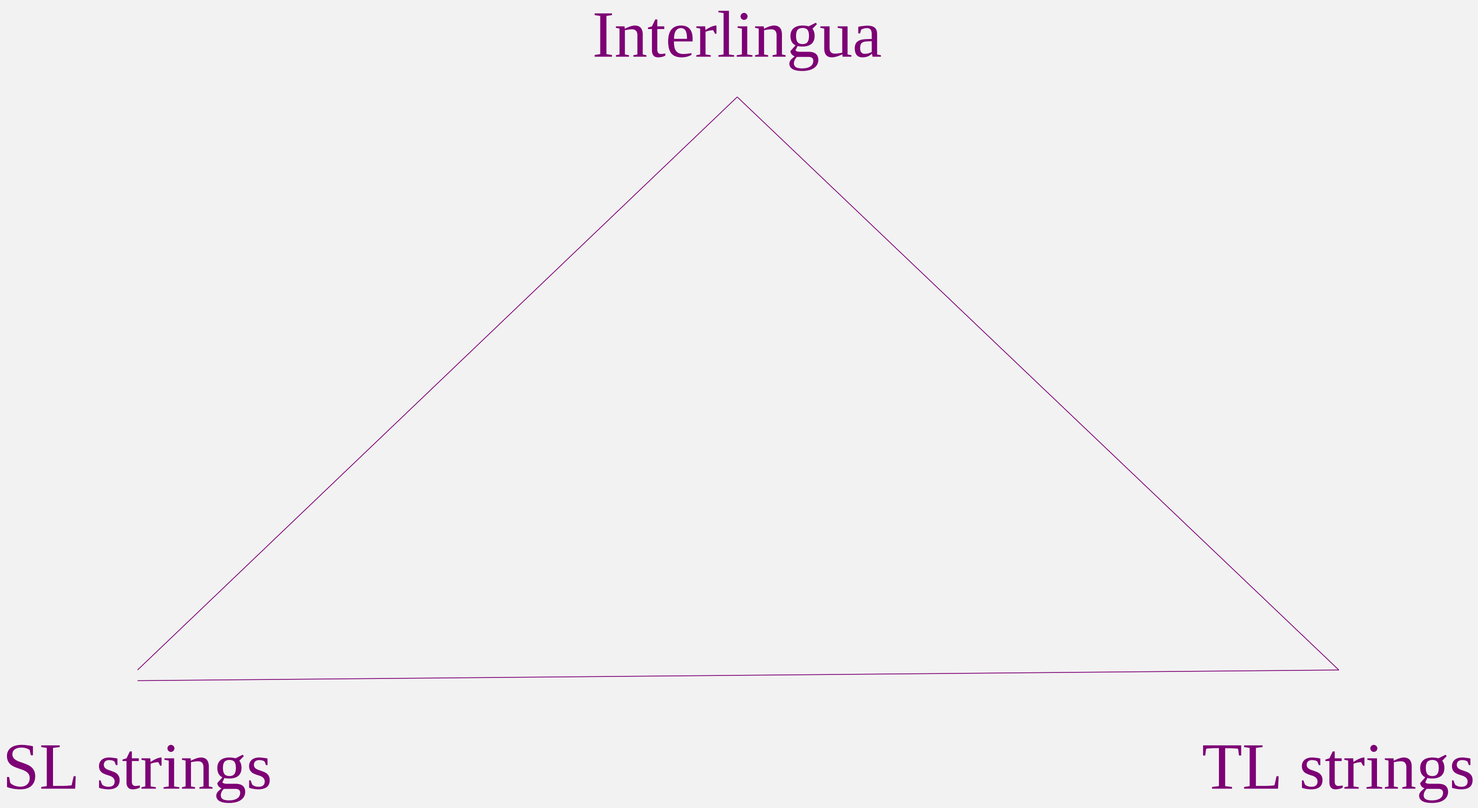
MRS and MT: Some history (1)

- Copestake et al 1995: Original motivation for MRS included MT applications
- Resolving scope ambiguities is hard, and usually not necessary
- Logical form equivalence is undecidable even in FOPL (Shieber 1993)
  - *young black cat* ↔ *gato negro y feroz* (Spanish)
  - *young black bull* ↔ *novillo negro*
- ⇒ Logical forms with less syntactic complexity
- ⇒ Underspecification wherever possible

MRS and MT: Some history (2)

- MRS originally developed in the context of Verb*Mobil* but not deployed fully for transfer-based MT in that project.
- In 2003, LOGON picks up the thread and builds first MRS-based MT system. (Norwegian → English; tourism brochures)
- Input is LFG, with MRSs projected from f-structure.
- Output is generated by the English Resource Grammar (HPSG).

ObMT Triangle



Interlingua

SL strings

TL strings

Is MRS an interlingua?

- Could MRS be used to encode an interlingua?
- Could our grammars produce such MRS-encoded interlingua?

Massively multilingual MT

- Problem of combinatory explosion ($n \times\ n$) :
- 2 languages: 2 sets of transfer rules
- 4 languages: 16 sets of transfer rules
- 24 languages: 576 sets of transfer rules
- 6000 languages: 36,000,000 sets of transfer rules

What are the alternatives?

- Design an interlingua, and create two grammars for each language:
    - strings ↔ ordinary MRS
    - ordinary MRS ↔ interlingua
- Hybrid interlingua/transfer-based model
    - partial lexical interlingua OR TRANSGRAPH
    - TL-side ‘accommodation’ transfer grammars—O(n)
    - transfer matrix to capture generalizations
- How far will approach 2 scale?
- How much mismatch is there?

MRS ‘Harmonization’ helps

- Just because it’s not an interlingua doesn’t mean they grammars can’t be brought closer together.
- Example 1: Demonstratives (adjective and determiner demonstratives should be more similar now).
- Example 2: COG-ST et al, reduction in quantifier-rel inventory
- Further potential for harmonization: Pronouns v. pro-drop
- Other examples?

LOGON Procesing steps

- Parse in source language
  - Visualization tools for parses and MRSs
- Apply source language’s transfer grammar to produce new MRSs
  - Visualization tools for transfer outputs
- Generate in target language from new MRSs
  - Visualization tools for MRS inputs
  - Compare to MRS produced by parsing expected output
  - Generator chart

Anatomy of a transfer rule (1)

- Quadruple: [CONTEXT:] INPUT [!FILTER] → OUTPUT
- Each item above is a (partial) MRS.
- Rules apply to complete MRSs to produce partially rewritten MRSs.
- Resource sensistive: INPUT is consumed in producing OUTPUT.
- CONTEXT: Additional propertes beyond the INPUT which must be satisfied.
- FILTER: Negative constraints; contexts in which the rule should NOT apply.

Anatomy of a transfer rule (2)

- Rules can be obligatory or optional.
- Optional rules introduce non-determinism in the transfer process.
- Pairing each optional rule with one obligatory rule cuts down the transfer search space.
- Rules can also be grouped into sets for ‘extrinsic’ ordering (which we probably won’t need).

Types and Translation

- Many transfer rules share most of their properties, differing only in lexical predicates/other small details
- → Define types of transfer rules, with particular instances, analogous to lexical types and lexical entries.
- Types mentioned in transfer rules will unify with compatible types in actual MRSs.
- In addition, the generator (it seems) will allow some unification of different (but compatible) types for feature values.

Example type

```
monotonic_mtr := mrs_transfer_rule &
[ CONTEXT.HOOK.LTOP #h,
  INPUT.HOOK.LTOP #h,
  OUTPUT.HOOK.LTOP #h ].
```

Example transfer rule instance

```
exist-way-transfer-rule := monotonic_omtr &
  [ INPUT.RELS <! [ PRED "_can_v_rel",
                    LBL #ltop,
                    ARG1 #emd ] !>,
    OUTPUT.RELS <! [ PRED "_way_n_rel",
                     ARG0 #way,
                     ARG1 #emd ],
                   [ PRED "_exist_v_rel",
                     ARG1 #way,
                     LBL #ltop ] !> ].
```

What about features of indices? (1)

- Can’t change value from input to output while maintaining identity of index with other positions.
- Person and number can be harmonized (in principle at least) by extending hierarchies on both sides.
- You might think you want to keep gender on pronouns and throw it out on nouns, but that only works on closely related languages.

What about features of indices? (2)

- The best (long-term) solution for gender is probably to do anaphora resolution on the SL and then fill in the appropriate gender on the TL side according to the antecedent.
- For now: global variable %transfer-properties-filter% (in lkb/mt.lsp) lists features to strip.

Practicalities

- Grammar clean-up
- Transfer matrix
- Generate based on inputs from English and Italian

Overview

- MRS and MT: Some history
- The Grammar Matrix and massively multilingual MT
- The LOGON architecture
    - Processing steps
    - Anatomy of a transfer rule
    - Types and translation