Slacker semantics

Why superficiality, dependency and avoidance of commitment can be the right way to go

Ann Copestake

Natural Language and Information Processing Group
Computer Laboratory
University of Cambridge

May 2009

Outline

A paper in the EACL 2009 proceedings provides more detail on Argument Labelling and DMRS.

Outline

Introduction: logic and natural language

DELPH-IN and broad-coverage computational compositional semantics

Argument Labelling

Dependency MRS

Conclusions

A slacker history of compositional semantics

- Aristotle (c350 BCE) — syllogisms:
  Every P is D, and every S is a P; so every S is D
- Medieval logicians: *dictum de omni* and *dictum de nullo*
  Rex is a brown dog implies Rex is a dog
  Rex is not a dog implies Rex is not a brown dog

  BUT some patient respects some doctor and
  every doctor is a senator implies
  some patient respects some senator
- Frege (1879): modern logic. Solves earlier problems but treats natural language structure as misleading.
- Montague (1970, 1974): symbolic logic systematically generated from natural language fragment.

(Pietroski in Stanford Encyclopedia of Philosophy)

A slacker history of compositional semantics

- Aristotle (c350 BCE) — syllogisms:
  Every P is D, and every S is a P; so every S is D
- Medieval logicians: *dictum de omni* and *dictum de nullo*
  Rex is a brown dog implies Rex is a dog
  Rex is not a dog implies Rex is not a brown dog

  BUT some patient respects some doctor and
  every doctor is a senator implies
  some patient respects some senator
- Frege (1879): modern logic. Solves earlier problems but treats natural language structure as misleading.
- Montague (1970, 1974): symbolic logic systematically generated from natural language fragment.

(Pietroski in Stanford Encyclopedia of Philosophy)

A slacker history of compositional semantics

- Aristotle (c350 BCE) — syllogisms:
  Every P is D, and every S is a P; so every S is D
- Medieval logicians: *dictum de omni* and *dictum de nullo*
  Rex is a brown dog implies Rex is a dog
  Rex is not a dog implies Rex is not a brown dog

  BUT some patient respects some doctor and
  every doctor is a senator implies
  some patient respects some senator
- Frege (1879): modern logic. Solves earlier problems but treats natural language structure as misleading.
- Montague (1970, 1974): symbolic logic systematically generated from natural language fragment.

(Pietroski in Stanford Encyclopedia of Philosophy)

A slacker history of compositional semantics

- Aristotle (c350 BCE) — syllogisms:
  Every P is D, and every S is a P; so every S is D
- Medieval logicians: *dictum de omni* and *dictum de nullo*
  Rex is a brown dog implies Rex is a dog
  Rex is not a dog implies Rex is not a brown dog

  BUT some patient respects some doctor and
  every doctor is a senator implies
  some patient respects some senator
- Frege (1879): modern logic. Solves earlier problems but treats natural language structure as misleading.
- Montague (1970, 1974): symbolic logic systematically generated from natural language fragment.

(Pietroski in Stanford Encyclopedia of Philosophy)

A slacker history of compositional semantics

- Aristotle (c350 BCE) — syllogisms:
  Every P is D, and every S is a P; so every S is D
- Medieval logicians: *dictum de omni* and *dictum de nullo*
  Rex is a brown dog implies Rex is a dog
  Rex is not a dog implies Rex is not a brown dog

  BUT some patient respects some doctor and
  every doctor is a senator implies
  some patient respects some senator
- Frege (1879): modern logic. Solves earlier problems but treats natural language structure as misleading.
- Montague (1970, 1974): symbolic logic systematically generated from natural language fragment.

(Pietroski in Stanford Encyclopedia of Philosophy)

Reducing the gap between Frege and natural language

- Event semantics (Davidson, 1967). Also Hobbs (1985) “Ontological promiscuity”.
  $\text{chase}'(e, x, y) \wedge \text{quick}'(e)$
- Generalized quantifiers (Barwise and Cooper, 1981).
  $\text{some}(\lambda x[dog(x)], \lambda y[bark(y)]) \equiv$ some(x,dog(x),bark(x))
- Quantifier scope underspecification (Alshawi and Crouch, 1992).
- Flat semantics.
- Simplified composition:
  Full quantifier scope underspecification means NPs of type e, transitive verbs of type $\langle e, \langle e, t \rangle \rangle$
  - Lambda calculus (possibly with labels).
  - Algebraic approaches (Zeevat, 1989).

Reducing the gap between Frege and natural language

- Event semantics (Davidson, 1967). Also Hobbs (1985) “Ontological promiscuity”.
  $\text{chase}'(e, x, y) \wedge \text{quick}'(e)$
- Generalized quantifiers (Barwise and Cooper, 1981).
  $\text{some}(\lambda x[dog(x)], \lambda y[bark(y)]) \equiv \text{some(x,dog(x),bark(x))}$
- Quantifier scope underspecification (Alshawi and Crouch, 1992). Avoid commitment!
- Flat semantics.
- Simplified composition:
  Full quantifier scope underspecification means NPs of type e, transitive verbs of type $\langle e, \langle e, t\rangle\rangle$
  - Lambda calculus (possibly with labels).
  - Algebraic approaches (Zeevat, 1989).

Reducing the gap between Frege and natural language

- Event semantics (Davidson, 1967). Also Hobbs (1985) “Ontological promiscuity”.
  $\text{chase}'(e, x, y) \wedge \text{quick}'(e)$
- Generalized quantifiers (Barwise and Cooper, 1981).
  $\text{some}(\lambda x[dog(x)], \lambda y[bark(y)]) \equiv \text{some(x,dog(x),bark(x))}$
- Quantifier scope underspecification (Alshawi and Crouch, 1992).
- Flat semantics. Be superficial!
- Simplified composition:
  Full quantifier scope underspecification means NPs of type e, transitive verbs of type $\langle e, \langle e, t\rangle\rangle$
  - Lambda calculus (possibly with labels).
  - Algebraic approaches (Zeevat, 1989).

Minimal Recursion Semantic (MRS)

Some big dog chased every cat

l1:some(x,h1,h2), h1 qeq l2, l2:big(x), l2:dog(x),
l4:chase(e,x,y), l5:every(y,h3,h4), h3 qeq l6, l6:cat(y)

Elementary predications (EPs) and scope constraints (qeqs)

some(x, big(x) $\wedge$ dog(x), every(y, cat(y), chase(e,x)))
h1=l2, h3=l6, h2=l5, h4=l4

every(y, cat(y), some(x, big(x) $\wedge$ dog(x), chase(e,x)))
h1=l2, h3=l6, h2=l4, h4=l1

Minimal Recursion Semantic (MRS)

Some big dog chased every cat

l1:some(x,h1,h2), h1 qeq l2, l2:big(x), l2:dog(x),
l4:chase(e,x,y), l5:every(y,h3,h4), h3 qeq l6, l6:cat(y)

Elementary predications (EPs) and scope constraints (qeqs)

some(x, big(x) $\wedge$ dog(x), every(y, cat(y), chase(e,x)))
h1=l2, h3=l6, h2=l5, h4=l4

every(y, cat(y), some(x, big(x) $\wedge$ dog(x), chase(e,x)))
h1=l2, h3=l6, h2=l4, h4=l1

Minimal Recursion Semantic (MRS)

Some big dog chased every cat

l1:some(x,h1,h2), h1 qeq l2, l2:big(x), l2:dog(x),
l4:chase(e,x,y), l5:every(y,h3,h4), h3 qeq l6, l6:cat(y)

Elementary predications (EPs) and scope constraints (qeqs)

some(x, big(x) $\wedge$ dog(x), every(y, cat(y), chase(e,x)))
h1=l2, h3=l6, h2=l5, h4=l4

every(y, cat(y), some(x, big(x) $\wedge$ dog(x), chase(e,x)))
h1=l2, h3=l6, h2=l4, h4=l1

Minimal Recursion Semantic (MRS)

Some big dog chased every cat

l1:some(x,h1,h2), h1 qeq l2, l2:big(x), l2:dog(x),
l4:chase(e,x,y), l5:every(y,h3,h4), h3 qeq l6, l6:cat(y)

Elementary predications (EPs) and scope constraints (qeqs)

some(x, big(x) $\wedge$ dog(x), every(y, cat(y), chase(e,x)))
h1=l2, h3=l6, h2=l5, h4=l4

every(y, cat(y), some(x, big(x) $\wedge$ dog(x), chase(e,x)))
h1=l2, h3=l6, h2=l4, h4=l1

Minimal Recursion Semantic (MRS)

Some big dog chased every cat

l1:some(x,h1,h2), h1 qeq l2, l2:big(x), l2:dog(x),
l4:chase(e,x,y), l5:every(y,h3,h4), h3 qeq l6, l6:cat(y)

Elementary predications (EPs) and scope constraints (qeqs)

some(x, big(x) $\wedge$ dog(x), every(y, cat(y), chase(e,x)))
h1=l2, h3=l6, h2=l5, h4=l4

every(y, cat(y), some(x, big(x) $\wedge$ dog(x), chase(e,x)))
h1=l2, h3=l6, h2=l4, h4=l1

Minimal Recursion Semantic (MRS)

Some big dog chased every cat

l1:some(x,h1,h2), h1 qeq l2, l2:big(x), l2:dog(x),
l4:chase(e,x,y), l5:every(y,h3,h4), h3 qeq l6, l6:cat(y)

Elementary predications (EPs) and scope constraints (qeqs)

some(x, big(x) $\wedge$ dog(x), every(y, cat(y), chase(e,x)))
h1=l2, h3=l6, h2=l5, h4=l4

every(y, cat(y), some(x, big(x) $\wedge$ dog(x), chase(e,x)))
h1=l2, h3=l6, h2=l4, h4=l1

Minimal Recursion Semantic (MRS)

Some big dog chased every cat

l1:some(x,h1,h2), h1 qeq l2, l2:big(x), l2:dog(x),
l4:chase(e,x,y), l5:every(y,h3,h4), h3 qeq l6, l6:cat(y)

Elementary predications (EPs) and scope constraints (qeqs)

some(x, big(x) $\wedge$ dog(x), every(y, cat(y), chase(e,x)))
h1=l2, h3=l6, h2=l5, h4=l4

every(y, cat(y), some(x, big(x) $\wedge$ dog(x), chase(e,x)))
h1=l2, h3=l6, h2=l4, h4=l1

MRS

- Assumes events and generalised quantifiers.
- Full quantifier scope underspecification, flat, algebraic composition.
- Tense, number etc as sorts on variables.
- Robust MRS: very similar but further decomposition.
- Compatible with many frameworks: extensively demonstrated for HPSG (computational and non-computational work), also CG. RMRS with RASP (no subcategorization in the lexicon).
(cf also Gardent and Kallmeyer (2003) for TAG).

MRS composition: she chases some dog

dog [l4,x] l4:dog(x)
some [l8,x1] {[l9,x1]$_n$} l3:some(x1, h1, h2), h1 qeq l9
some dog $op_n(\mathrm{Det}, \mathrm{N})$
[l8,x] l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

chases [l2,e] {[l2,x2]$_{subj}$, [l2,x3]$_{obj}$}, l2:chase(e,x2,x3)
chases some dog $op_{obj}(\mathrm{V}, \mathrm{NP})$
[l2,e] {[l2,x12]$_{subj}$}, l2:chase(e,x2,x), l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

she [l0,y] l0:pron(y)
she chases some dog $op_{subj}(\mathrm{VP}, \mathrm{NP})$
[l2,e] l2:pron(y), l2:chase(e,y,x), l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

MRS composition: she chases some dog

dog [l4,x] l4:dog(x) hook

some [l8,x1] $\{[l9,x1]_n\}$ l3:some(x1, h1, h2), h1 qeq l9

some dog $op_n(\mathrm{Det}, \mathrm{N})$

[l8,x] l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

chases [l2,e] $\{[l2,x2]_{subj}, [l2,x3]_{obj}\}$, l2:chase(e,x2,x3)

chases some dog $op_{obj}(\mathrm{V}, \mathrm{NP})$

[l2,e] $\{[l2,x12]_{subj}\}$, l2:chase(e,x2,x), l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

she [l0,y] l0:pron(y)

she chases some dog $op_{subj}(\mathrm{VP}, \mathrm{NP})$

[l2,e] l2:pron(y), l2:chase(e,y,x), l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

MRS composition: she chases some dog

dog [l4,x] l4:dog(x)

some [l8,x1] {[l9,x1]$_n$} l3:some(x1, h1, h2), h1 qeq l9 slot

some dog $op_n(\mathrm{Det}, \mathrm{N})$

[l8,x] l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

chases [l2,e] {[l2,x2]$_{subj}$, [l2,x3]$_{obj}$}, l2:chase(e,x2,x3)

chases some dog $op_{obj}(\mathrm{V}, \mathrm{NP})$

[l2,e] {[l2,x12]$_{subj}$}, l2:chase(e,x2,x), l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

she [l0,y] l0:pron(y)

she chases some dog $op_{subj}(\mathrm{VP}, \mathrm{NP})$

[l2,e] l2:pron(y), l2:chase(e,y,x), l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

MRS composition: she chases some dog

dog [l4,x] l4:dog(x)

some [l8,x1] {[l9,x1]$_n$} l3:some(x1, h1, h2), h1 qeq l9

some dog $op_n(\mathrm{Det}, \mathrm{N})$ hook fills slot, x1=x, l9=l4

[l8,x] l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

chases [l2,e] {[l2,x2]$_{subj}$, [l2,x3]$_{obj}$}, l2:chase(e,x2,x3)

chases some dog $op_{obj}(\mathrm{V}, \mathrm{NP})$

[l2,e] {[l2,x12]$_{subj}$}, l2:chase(e,x2,x), l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

she [l0,y] l0:pron(y)

she chases some dog $op_{subj}(\mathrm{VP}, \mathrm{NP})$

[l2,e] l2:pron(y), l2:chase(e,y,x), l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

MRS composition: she chases some dog

dog [l4,x] l4:dog(x)
some [l8,x1] $\{[l9,x1]_n\}$ l3:some(x1, h1, h2), h1 qeq l9
some dog $op_n(\mathrm{Det}, \mathrm{N})$
[l8,x] l3:some(x,h1,h2), l4:dog(x), h1 qeq l4

chases [l2,e] $\{[l2,x2]_{subj}, [l2,x3]_{obj}\}$, l2:chase(e,x2,x3)
chases some dog $op_{obj}(\mathrm{V}, \mathrm{NP})$
[l2,e] $\{[l2,x12]_{subj}\}$, l2:chase(e,x2,x), l3:some(x,h1,h2),
l4:dog(x), h1 qeq l4
she [l0,y] l0:pron(y)
she chases some dog $op_{subj}(\mathrm{VP}, \mathrm{NP})$
[l2,e] l2:pron(y), l2:chase(e,y,x), l3:some(x,h1,h2), l4:dog(x),
h1 qeq l4

Natural compositional semantics?

- Still interpretable as a symbolic logic, but syntax of logic is closer to natural language.
- Compositional semantics as annotation, not replacement.
- Principle: capture all and only the information from syntax and productive morphology. So formalism must allow well-formed structures with that information alone.
- Composition which can be expressed as incremental specialisation (further specialisation for anaphora resolution, WSD, etc). Support for fully incremental left-to-right processing.
- Compatible with multiple approaches to syntax (including 'shallow' ones).
- Alternative? Natural Logic (Lakoff, 1970)?

Broad-coverage processing and computational semantics

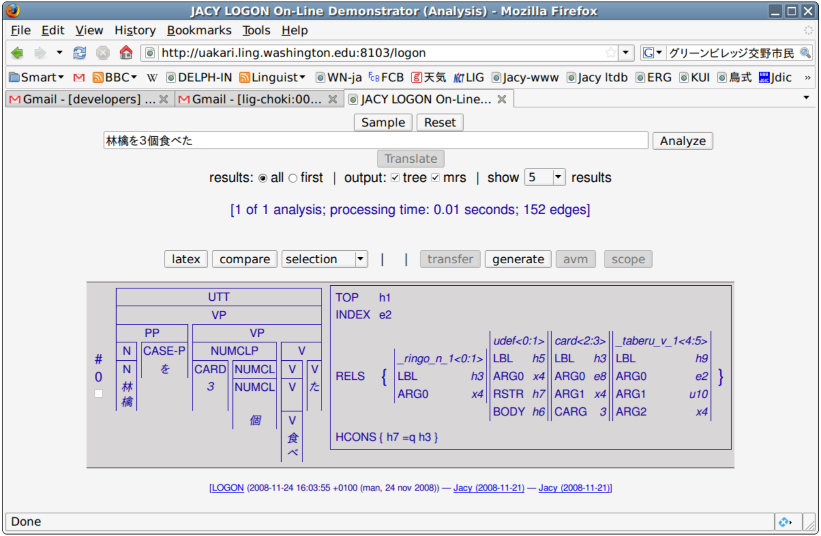
- High-throughput parsers with semantic output: CCG, RASP, ENJU, XLE ... ERG/PET (medium throughput) ...
- Effective statistical techniques for syntactic parse ranking.
- Limited resources:
  - No underlying knowledge base for disambiguation.
  - Limited lexical information available, even for syntax (e.g., multiword expressions).
- Must avoid semantics multiplying readings: several types of underspecification.
- Support inter-sentential anaphora/text structure.
- Inference, robust inference, semantic pattern matching.

Outline

DELPH-IN: Deep Linguistic Processing using HPSG

- Informal collaboration on tools and grammars: see `http://www.delph-in.net/`
- Large grammars for English, German and Japanese; medium/growing for Spanish, Norwegian, Portuguese, Korean, French. Many small grammars.
- Common semantic framework: MRS and Robust MRS. RMRS also from shallower parsing, chunking, POS tagging.
- Parsing and generation (realization), integrated shallower processing.
- Grammar Matrix: framework/starter kit for the development of grammars for diverse languages.

JACY example



JACY example

ringo	wo	3	ko	tabeta
apple	acc	3	classifier	eat past

[pro] ate three apples: default interpretation "I ate three apples"

l_3:_ringo_n_1(x_4),
l_5:udef(x_4, h_7, h_6),
l_3:card(e_8, x_4, 3),
l_9:_taberu_v_1($e_2\{\text{TENSE } \textit{past}, \text{PROG -}, \text{PERF -}, \text{SF } \textit{prop}\}$, u_{10}, x_4)
$h_7 =_q l_3$

Leading underscores: predicates correspond to lexeme.
No underscores: 'grammar' predicates (shared).
(Token/character positions not shown.)

A real example

Very few of the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

A real example

Very few of the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

modified quantifier

A real example

Very few of the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

partitive

A real example

Very few of the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

compound nominal

A real example

Very few of the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

reduced relative

A real example

Very few of the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

modified modifier

A real example

Very few of the Chinese construction companies consulted were even remotely interested in entering into such an arrangement with a local partner.

predeterminer

A real example

l_3:part_of(x_4{PERS *3*, NUM *pl*}, x_5{PERS *3*, NUM *pl*}),
l_6:udef_q(x_4, h_7, h_8),
l_3:_very_x_deg(e_9,e_{10}{SF *prop*}),
l_3:_few_a(e_{10}, x_4),
l_{11}:_the_q(x_5, h_{13}, h_{12}),
l_{14}:compound(e_{16}{SF *prop*, TENSE *untensed*, MOOD *indicative*, PROG -, PERF -}, x_5, x_{15}),
l_{17}:udef_q(x_{15}, h_{18}, h_{19}),
l_{20}:_chinese_a_1(e_{21}{SF *prop*, TENSE *untensed*, MOOD *indicative*}, x_{15}),
l_{20}:_construction_n(x_{15}),
l_{14}:_company_n(x_5),
l_3:_consult_v_1(e_{24}{SF *prop*, TENSE *untensed*, MOOD *indicative*, PROG -, PERF -}, p_{25}, x_4),
l_{27}:_even_a_1(e_{28},e_2{SF *prop*, TENSE *past*, MOOD *indicative*, PROG -, PERF -}),
l_{27}:_remotely_x_deg(e_{29}{SF *prop*, TENSE *untensed*, MOOD *indicative*, PROG -, PERF -}, e_2),
l_{27}:_interested_a_in(e_2, x_4, x_{30}{PERS *3*, NUM *sg*, GEND *n*}),
l_{31}:udef_q(x_{30}, h_{32}, h_{33}),
l_{34}:_enter_v_1(e_{35}{SF *prop*, TENSE *untensed*, MOOD *indicative*, PROG +, PERF -}, p_{36}),
l_{37}:nominalization(x_{30}, h_{34}),
l_{34}:_into_p(e_{38},e_{35}, x_{39}{PERS *3*, NUM *sg*, IND +}),
l_{40}:_such+a_q(x_{39}, h_{42}, h_{41}),
l_{43}:_arrangement_n_1(x_{39}),
l_{37}:_with_p($e_{44}x_{30}$, x_{45}{PERS *3*, NUM *sg*, IND +}),
l_{46}:_a_q(x_{45}, h_{48}, h_{47}),
l_{49}:_local_a_1(e_{50}{SF *prop*, TENSE *untensed*, MOOD *indicative*}, x_{45}),
l_{49}:_partner_n_1(x_{45}), $h_{48} =_q l_{49}$, $h_{42} =_q l_{43}$, $h_{32} =_q l_{37}$, $h_{18} =_q l_{20}$, $h_{13} =_q l_{14}$, $h_7 =_q l_3$

Compositional semantics in DELPH-IN

Meaning information that can be associated with syntax and morphology.

- Fully identified (for English): predicate-argument structure (nouns, adjectives, verbs), modifier scope (e.g., *probably*), many constructions (e.g., relative clauses, appositives, tag questions, pseudo-partitives), ...
- Partially identified/underspecified: quantifier scope, compound nouns, tense, aspect, massness, some sense extensions ...
- Possible additions: further (productive) derivational morphology and sense extension, underspecified distributivity, genericity ...
- In progress: tools for external mapping to deeper semantics, lexical semantics.

RMRS

Split off most of EP's arguments: relate to predicate via anchor

MRS:
l1:some(x,h1,h2), h1 qeq l2,
l2:dog(x),
l3:chase(e,x,y),
l4:every(y,h3,h4), h3 qeq l65,
l5:cat(y)

RMRS:
l1:a1:some, BV(a1,x), RSTR(a1,h1), BODY(a1,h2), h1 qeq l2,
l2:a2:dog(x),
l3:a3:chase(e), ARG1(a3,x), ARG2(a3,y),
l4:a4:every, BV(a4,y), RSTR(a4,h3), BODY(a4,h4), h3 qeq l5,
l5:a5:cat(y)

Allows omission or underspecification of arguments.

RMRS

Split off most of EP's arguments: relate to predicate via anchor

MRS:
l1:some(x,h1,h2), h1 qeq l2,
l2:dog(x),
l3:chase(e,x,y),
l4:every(y,h3,h4), h3 qeq l65,
l5:cat(y)

RMRS:
l1:a1:some, BV(a1,x), RSTR(a1,h1), BODY(a1,h2), h1 qeq l2,
l2:a2:dog(x),
l3:a3:chase(e), ARG1(a3,x), ARG2(a3,y),
l4:a4:every, BV(a4,y), RSTR(a4,h3), BODY(a4,h4), h3 qeq l5,
l5:a5:cat(y)

Allows omission or underspecification of arguments.

RMRS

Split off most of EP's arguments: relate to predicate via anchor

MRS:
l1:some(x,h1,h2), h1 qeq l2,
l2:dog(x),
l3:chase(e,x,y),
l4:every(y,h3,h4), h3 qeq l65,
l5:cat(y)

RMRS:
l1:a1:some, BV(a1,x), RSTR(a1,h1), BODY(a1,h2), h1 qeq l2,
l2:a2:dog(x),
l3:a3:chase(e), ARG1(a3,x), ARG2(a3,y),
l4:a4:every, BV(a4,y), RSTR(a4,h3), BODY(a4,h4), h3 qeq l5,
l5:a5:cat(y)

Allows omission or underspecification of arguments.

RMRS

Split off most of EP's arguments: relate to predicate via anchor

MRS:
l1:some(x,h1,h2), h1 qeq l2,
l2:dog(x),
l3:chase(e,x,y),
l4:every(y,h3,h4), h3 qeq l65,
l5:cat(y)

RMRS:
l1:a1:some, BV(a1,x), RSTR(a1,h1), BODY(a1,h2), h1 qeq l2,
l2:a2:dog(x),
l3:a3:chase(e), ARG1(a3,x), ARG2(a3,y),
l4:a4:every, BV(a4,y), RSTR(a4,h3), BODY(a4,h4), h3 qeq l5,
l5:a5:cat(y)

Allows omission or underspecification of arguments.

RMRS

Split off most of EP's arguments: relate to predicate via anchor

MRS:
l1:some(x,h1,h2), h1 qeq l2,
l2:dog(x),
l3:chase(e,x,y),
l4:every(y,h3,h4), h3 qeq l65,
l5:cat(y)

RMRS:
l1:a1:some, BV(a1,x), RSTR(a1,h1), BODY(a1,h2), h1 qeq l2,
l2:a2:dog(x),
l3:a3:chase(e), ARG1(a3,x), ARG2(a3,y),
l4:a4:every, BV(a4,y), RSTR(a4,h3), BODY(a4,h4), h3 qeq l5,
l5:a5:cat(y)

Allows omission or underspecification of arguments.

RMRS

Split off most of EP's arguments: relate to predicate via anchor

MRS:
l1:some(x,h1,h2), h1 qeq l2,
l2:dog(x),
l3:chase(e,x,y),
l4:every(y,h3,h4), h3 qeq l65,
l5:cat(y)

RMRS:
l1:a1:some, BV(a1,x), RSTR(a1,h1), BODY(a1,h2), h1 qeq l2,
l2:a2:dog(x),
l3:a3:chase(e), ARG1(a3,x), ARG2(a3,y),
l4:a4:every, BV(a4,y), RSTR(a4,h3), BODY(a4,h4), h3 qeq l5,
l5:a5:cat(y)

Allows omission or underspecification of arguments.

Semantics via incremental annotation (RMRS)

Most cats noisily chased a large dog

most_DAT cat_NN2 noisily_RR chase_VVD a_AT1 large_JJ dog_NN1

l1:a1:most_q
l2:a2:cat_n(x2)
l3:a3:noisy(e3)
l4:a4:chase(e4)
l5:a5:a(x5)
l6:a6:large(e6)
l7:a7:dog(x7)

Semantics via incremental annotation (RMRS)

Most cats noisily chased a large dog

most_DAT cat_NN2 noisily_RR chase_VVD a_AT1 large_JJ dog_NN1

l1:a1:most_q	a1:BV(x2)
l2:a2:cat_n(x2)	
l3:a3:noisy(e3)	
l4:a4:chase(e4)	
l5:a5:a(x5)	x5=x7
l6:a6:large(e6)	a6:ARG1(x7) l6=l7
l7:a7:dog(x7)	

Semantics via incremental annotation (RMRS)

Most cats noisily chased a large dog

most_DAT cat_NN2 noisily_RR chase_VVD a_AT1 large_JJ dog_NN1

l1:a1:most_q	a1:BV(x2)
l2:a2:cat_n(x2)	
l3:a3:noisy(e3)	l3=l4 e3=e4
l4:a4:chase(e4)	a4:ARG1(x2) a4:ARG2(x5)
l5:a5:a(x5)	x5=x7
l6:a6:large(e6)	a6:ARG1(x7) l6=l7
l7:a7:dog(x7)	

Semantics via incremental annotation (RMRS)

Most cats noisily chased a large dog

most_DAT cat_NN2 noisily_RR chase_VVD a_AT1 large_JJ dog_NN1

l1:a1:most_q	a1:BV(x2) a1:RSTR(h1) h1 qeq l2
l2:a2:cat_n(x2)	
l3:a3:noisy(e3)	l3=l4 e3=e4
l4:a4:chase(e4)	a4:ARG1(x2) a4:ARG2(x5)
l5:a5:a(x5)	x5=x7 a5:RSTR(h5) h5 qeq l6
l6:a6:large(e6)	a6:ARG1(x7) l6=l7
l7:a7:dog(x7)	

Semantics via incremental annotation (RMRS)

Most cats noisily chased a large dog

most_DAT cat_NN2 noisily_RR chase_VVD a_AT1 large_JJ dog_NN1

l1:a1:most_q	a1:BV(x2) a1:RSTR(h1) h1 qeq l2	a1:BODY(l5)
l2:a2:cat_n(x2)		
l3:a3:noisy(e3)	l3=l4 e3=e4	
l4:a4:chase(e4)	a4:ARG1(x2) a4:ARG2(x5)	
l5:a5:a(x5)	x5=x7 a5:RSTR(h5) h5 qeq l6	a1:BODY(l3)
l6:a6:large(e6)	a6:ARG1(x7) l6=l7	
l7:a7:dog(x7)		

Semantics via incremental annotation (RMRS)

Most cats noisily chased a large dog

most_DAT cat_NN2 noisily_RR chase_VVD a_AT1 large_JJ dog_NN1

l1:a1:most_q	a1:BV(x2) a1:RSTR(h1) h1 qeq l2	a1:BODY(l3)
l2:a2:cat_n(x2)		
l3:a3:noisy(e3)	l3=l4 e3=e4	
l4:a4:chase(e4)	a4:ARG1(x2) a4:ARG2(x5)	
l5:a5:a(x5)	x5=x7 a5:RSTR(h5) h5 qeq l6	a1:BODY(l1)
l6:a6:large(e6)	a6:ARG1(x7) l6=l7	
l7:a7:dog(x7)		

RMRS as semantic annotation of lexeme sequence

- Annotate most lexemes with random label, anchor, arg0. Note: null semantics for some words, e.g., infinitival *to*.
- Partially disambiguate lexeme with n, v etc.
- Add sortal information to arg0.
- Implicit conjunction: add equalities between labels.
- Ordinary arguments: add ARG relations (possibly underspecified e.g., ARGn) between anchors and arg0.
- Scopal arguments: add ARG relation plus qeq between anchors and labels.

Standoff annotation on original text via character positions.

Semantics and grammar engineering

- Ongoing extensive experimentation with details of the analysis (compare annotation).
- Highly empirical: working with real data for ongoing projects (simple examples for regression testing).
- Interactions can be complex: require implementation to investigate.
- Limitations of semantic literature:
  - base assumptions: ambiguity, lexical resources
  - sometimes ad-hoc or omitted syntax
  - few/no analyses: e.g., modification of quantifiers (*almost every*)
- If we do capture syntax/morphology, (R)MRS can be a basis for deeper semantics.

Outline

Introduction: logic and natural language

DELPH-IN and broad-coverage computational compositional semantics

Argument Labelling

Dependency MRS

Conclusions

Arguments in (R)MRS

- l:like_v_1(e,x,y) ≡ l:a:like_v_1(e), ARG1(a,x), ARG2(a,y)
- Arity may not vary. Different numbers of obligatory arguments requires different predicates: e.g., *leave* ‘depart’ vs ‘bequeath’.
- Argument labels: open class and prepositions have ARG0, ARG1, ARG2, ARG3 and (rarely) ARG4. Larger inventory for closed class and constructions (BV, RSTR etc).
- Lexical type controls linking of syntax and ARGs.
- ARG1 . . . ARGn ordering governed by the obliqueness hierarchy. Argument sequence contiguous. ARG1 is the subject of the base form (e.g., non-passivised).

Arguments in (R)MRS

- l:like_v_1(e,x,y) $\equiv$ l:a:like_v_1(e), ARG1(a,x), ARG2(a,y)
- Arity may not vary. Different numbers of obligatory arguments requires different predicates: e.g., *leave* 'depart' vs 'bequeath'.
- Argument labels: open class and prepositions have ARG0, ARG1, ARG2, ARG3 and (rarely) ARG4. Larger inventory for closed class and constructions (BV, RSTR etc).
- Lexical type controls linking of syntax and ARGs.
- ARG1 ... ARGn ordering governed by the obliqueness hierarchy. Argument sequence contiguous. ARG1 is the subject of the base form (e.g., non-passivised).

This is NOT the same as PropBank!

Argument labelling versus role assignment

- Roles (e.g., AGENT, GOAL, INSTRUMENT) are intended to be semantically meaningful.
- Assume this means that each role label implies a (default) entailment of one or more useful real world propositions).
- There seems no prospect of finding a small set of roles which can also be used to link predicates to arguments compositionally.
- A more modest aim: can we make ARG1 consistently agentive in verbs? (cf PropBank)

Causatives

(1) Kim boiled the water.

(2) The water boiled.

Can the grammar be set up so subject is ARG1 in causative but *water* is ARG2 in both?

Target RMRS representations (simplified):

(3) l:a:boil_v(e), a:ARG1(Kim), a:ARG2(x), water(x)

(4) l:a:boil_v(e), a:ARG2(x), water(x)

More causatives

(5) Michaela galloped the horse to the far end of the meadow, . . .

(6) With that Michaela nudged the horse with her heels and off the horse galloped.

(7) Michaela declared, “I shall call him Lightning because he runs as fast as lightning.” And with that, off she galloped.

Causative verbs of movement

- Option 1. Causative is obligatorily transitive. Then in *Michaela galloped*, Michaela is ARG2.
  Role labels of intransitive movement verbs would depend on knowing about the causative.
- Option 2. *gallop* has a causative intransitive form. *Michaela galloped* is ARG1, *the horse galloped* is ARG2 (but only in the case when it has a rider).
  Irresolvable ambiguity, plus losing a generalisation about movement.

True lexical anti-causatives?

Causative verbs of movement

- Option 1. Causative is obligatorily transitive. Then in *Michaela galloped*, Michaela is ARG2.
  Role labels of intransitive movement verbs would depend on knowing about the causative.
- Option 2. *gallop* has a causative intransitive form. *Michaela galloped* is ARG1, *the horse galloped* is ARG2 (but only in the case when it has a rider).
  Irresolvable ambiguity, plus losing a generalisation about movement.

True lexical anti-causatives?

Causative verbs of movement

- Option 1. Causative is obligatorily transitive. Then in *Michaela galloped*, Michaela is ARG2.
  Role labels of intransitive movement verbs would depend on knowing about the causative.
- Option 2. *gallop* has a causative intransitive form.
  Michaela galloped is ARG1, *the horse galloped* is ARG2 (but only in the case when it has a rider).
  Irresolvable ambiguity, plus losing a generalisation about movement.

True lexical anti-causatives?

Causative verbs of movement

- Option 1. Causative is obligatorily transitive. Then in *Michaela galloped*, Michaela is ARG2.
  Role labels of intransitive movement verbs would depend on knowing about the causative.
- Option 2. *gallop* has a causative intransitive form. *Michaela galloped* is ARG1, *the horse galloped* is ARG2 (but only in the case when it has a rider).
  Irresolvable ambiguity, plus losing a generalisation about movement.

True lexical anti-causatives?

Causative verbs of movement

- Option 1. Causative is obligatorily transitive. Then in *Michaela galloped*, Michaela is ARG2.
  Role labels of intransitive movement verbs would depend on knowing about the causative.
- Option 2. *gallop* has a causative intransitive form.
  Michaela galloped is ARG1, *the horse galloped* is ARG2 (but only in the case when it has a rider).
  Irresolvable ambiguity, plus losing a generalisation about movement.

True lexical anti-causatives?

The slacker alternative: systematic sense labelling

(8) Kim boiled the water.
l:a:boil_v_cause(e), a:ARG1(k), a:ARG2(x), water(x)

(9) The water boiled.
l:a:boil_v_1(e), a:ARG1(x), water(x)

- Inferences may be made about ARG1 and ARG2 for the _cause verbs.
- Identification of further classes can be done incrementally, supporting mapping of ARGs on a class-by-class basis (perhaps into FrameNet roles).
- Possible generalisation for all verbs (Dowty, 1991): ARG2 is not more agentive than an ARG1 (ARG1 has number of p-agt properties $\geq$ ARG2).

Outline

Introduction: logic and natural language

DELPH-IN and broad-coverage computational compositional semantics

Argument Labelling

Dependency MRS

Conclusions

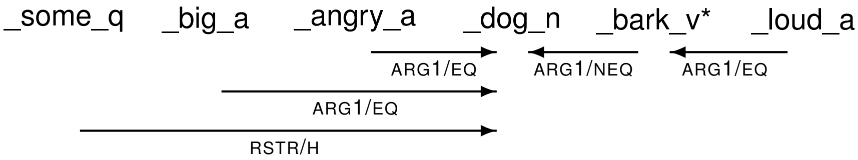
Semantic dependency representations

- Oepen: MRS elementary dependencies, a partial representation. Treebanking, features for parse ranking.
- Dependency MRS (DMRS) goals:
  - predicates with simple inventory of links, no variables
  - all information is retained so interconvertible with RMRS
  - structure is minimal (no redundancy)
  - applicable to different grammars, robust to changes in grammars
- No direct logical interpretation.

Semantic dependency representations

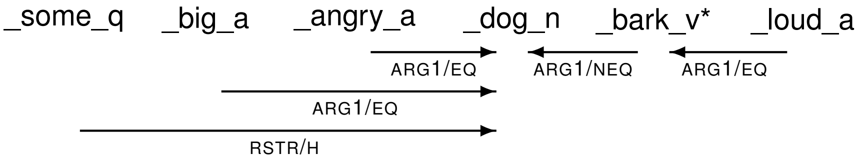
- Oepen: MRS elementary dependencies, a partial representation. Treebanking, features for parse ranking.
- Dependency MRS (DMRS) goals:
  - predicates with simple inventory of links, no variables
  - all information is retained so interconvertible with RMRS
  - structure is minimal (no redundancy)
  - applicable to different grammars, robust to changes in grammars
- No direct logical interpretation.

DMRS



l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

DMRS



l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

Characteristic variables

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

_some_q(x4,_big_a(e8,x4) ∧ _angry_a(e9, x4) ∧_dog_n(x4),
_bark_v(e2,x4) ∧_loud_a(e10,e2))

RMRS: EPs may have a distinguished argument.
Characteristic variable property: the distinguished argument of an RMRS EP (arg0) is unique to it.
Introduced into DELPH-IN grammars for grammar-internal reasons.

Characteristic variables

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

_some_q(x4,_big_a(e8,x4) ∧ _angry_a(e9, x4) ∧_dog_n(x4),
_bark_v(e2,x4) ∧_loud_a(e10,e2))

RMRS: EPs may have a distinguished argument.
Characteristic variable property: the distinguished argument of an RMRS EP (arg0) is unique to it.
Introduced into DELPH-IN grammars for grammar-internal reasons.

Characteristic variables

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

_some_q(x4,_big_a(e8,x4) ∧ _angry_a(e9, x4) ∧_dog_n(x4),
_bark_v(e2,x4) ∧_loud_a(e10,e2))

RMRS: EPs may have a distinguished argument.
Characteristic variable property: the distinguished argument of an RMRS EP (arg0) is unique to it.
Introduced into DELPH-IN grammars for grammar-internal reasons.

Adjectives and characteristic variables

- Use (and misuse) of event variables: e.g., Hobbs (1985), Asher (1993), Maienborn (2005).
- Long-standing use of event variables on adjectives in DELPH-IN grammars.
- Predicative uses without copula in semantics, tense as a property of the event variable.

  (10) She was angry.

  (11) pron(x), angry(e_{past}, x)
- Attributive adjective temporal modification in German.

  (12) Der im Fruehling gruene Rasen ist jetzt braun und ausgetrocknet.
  The in spring green lawn is now brown and dried-out.

RMRS to DMRS: RMRS graphs

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

1. label equality: EPs with equal labels
2. qeq graph: scopal argument in EP to label
   ltop: label of one of more EPs
3. variable graph: non-scopal arguments to characteristic variables

RMRS to DMRS: RMRS graphs

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

1. label equality: EPs with equal labels
2. qeq graph: scopal argument in EP to label
   ltop: label of one of more EPs
3. variable graph: non-scopal arguments to characteristic variables

RMRS to DMRS: RMRS graphs

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

1. label equality: EPs with equal labels
2. qeq graph: scopal argument in EP to label
   ltop: label of one of more EPs
3. variable graph: non-scopal arguments to characteristic variables

RMRS to DMRS: RMRS graphs

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

1. label equality: EPs with equal labels
2. qeq graph: scopal argument in EP to label
   ltop: label of one of more EPs
3. variable graph: non-scopal arguments to characteristic variables

RMRS to DMRS: RMRS graphs

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

1. label equality: EPs with equal labels
2. qeq graph: scopal argument in EP to label
   ltop: label of one of more EPs
3. variable graph: non-scopal arguments to characteristic variables

RMRS to DMRS: RMRS graphs

l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

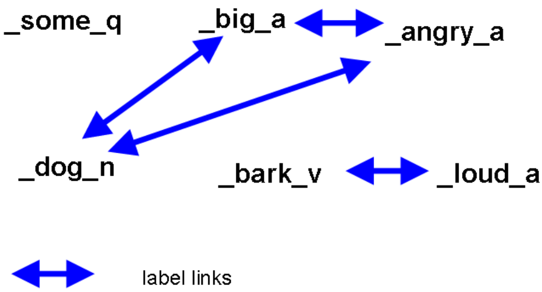
1. label equality: EPs with equal labels
2. qeq graph: scopal argument in EP to label
   ltop: label of one of more EPs
3. variable graph: non-scopal arguments to characteristic variables

RMRS to DMRS: RMRS graphs

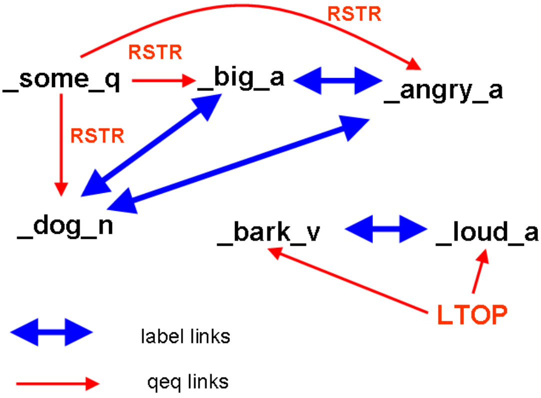
l1:a1:_some_q, BV(a1,x4), RSTR(a1,h5), BODY(a1,h6),
h5 qeq l2,
l2:a2:_big_a(e8), ARG1(a2,x4),
l2:a3:_angry_a(e9), ARG1(a3,x4),
l2:a4:_dog_n(x4),
l4:a5:_bark_v(e2), ARG1(a5,x4),
l4:a6:_loud_a(e10), ARG1(a6,e2)

1. label equality: EPs with equal labels
2. qeq graph: scopal argument in EP to label
   ltop: label of one of more EPs
3. variable graph: non-scopal arguments to characteristic variables

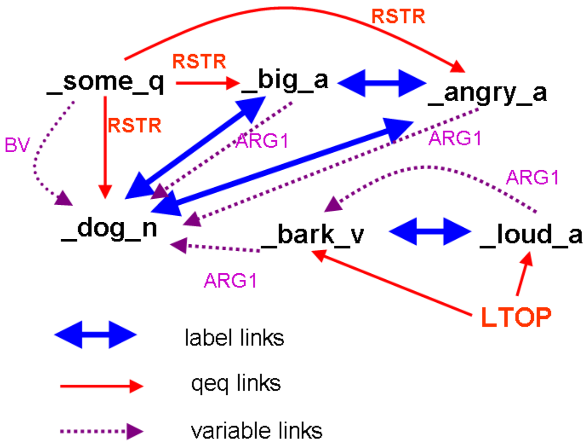
RMRS label equality graph



Label equality and qeq graph

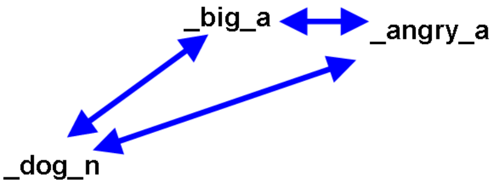


Label equality, qeq and variable graph



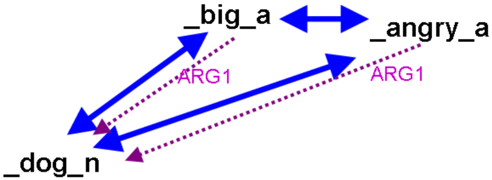
Redundant link problem

Label equalities give $n(n-1)/2$ binary links.



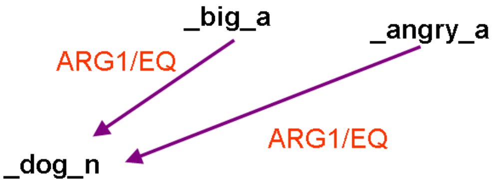
Variable links

Variable links relate an EP argument to a unique EP because of the characteristic variable property.

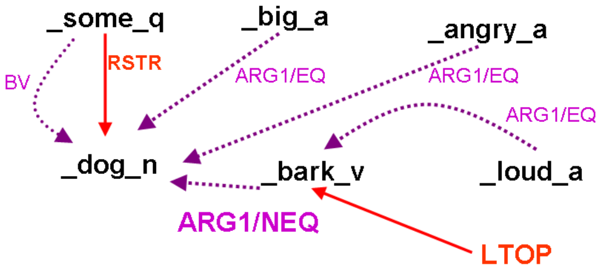


Merged links

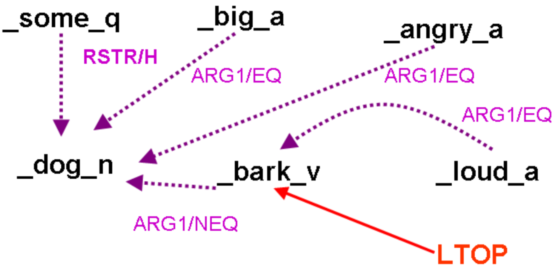
Use variable graph to decide on canonical links.



Merged links on full graph



DMRS


_some_q
_big_a
_angry_a
RSTR/H
ARG1/EQ
ARG1/EQ
ARG1/EQ
_dog_n
_bark_v
_loud_a
ARG1/NEQ
LTOP

Semantics of relative clauses

Two pieces of semantics associated with relative clause attachment:

1. Modified noun as filler of gap in the relative clause.
2. Relative clause conjoined with noun (hence part of quantifier RSTR).

The cat which barked slept.

_the_q _cat_n_1 _bark_v_1 _sleep_v_1

RSTR/H ARG1/EQ

ARG1/NEQ

Relative clauses and the EQ link

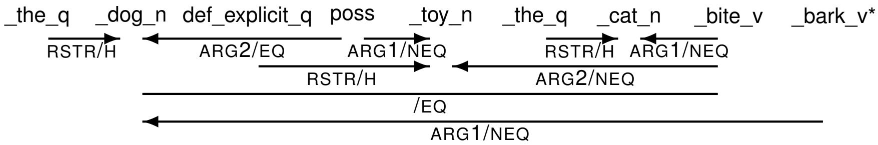
who the cat bit: gap is in main verb of relative clause.

$[\ l, e\ ]\ \{[\ l, y\ ]_{mod}\}\ [\ cat(z), l{:}bite(e,z,y)\ ]$

whose toy the cat bit: gap not in main verb of rel. clause

$[\ l, e\ ]\ \{[\ l, x\ ]_{mod}\}\ [\ poss(x,y), toy(y), cat(z), l{:}bite(e,z,y)\ ]$

The dog whose toy the cat bit barked.



Relative clauses and the EQ link

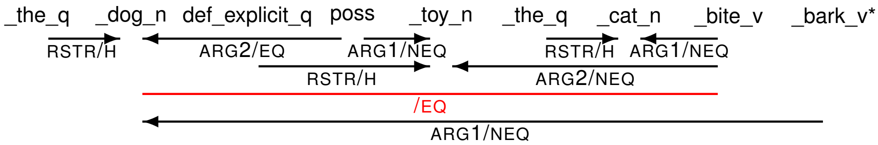
who the cat bit: gap is in main verb of relative clause.

$[\ l, e\ ]\ \{[\ l, y\ ]_{mod}\}\ [\ cat(z), l{:}bite(e,z,y)\ ]$

whose toy the cat bit: gap not in main verb of rel. clause

$[\ l, e\ ]\ \{[\ l, x\ ]_{mod}\}\ [\ poss(x,y), toy(y), cat(z), l{:}bite(e,z,y)\ ]$

The dog whose toy the cat bit barked.



DMRS and grammatical relations

(13) Not all those who wrote opposed the proposal.

PARC	pron form(pro3, those) adjunct(pro3, write) adjunct type(write, relative) pron form(pro4, who) pron type(pro4, relative) pron rel(write, pro4) topic rel(write, pro4)
GR	(cmod who those wrote) (ncsubj wrote those)
Stanford	nsubj(wrote, those) rel(wrote, who) rcmod(those, wrote)

MRS treatment uses several construction predicates: ‘those people who wrote’.
No predicate from relative clause *who* because of reduced relatives *the people consulted objected*.

Outline

Conclusions

- Compositional semantics: annotation to make those aspects of meaning conveyed by syntax and morphology more accessible to subsequent processing.
- Be superficial and avoid commitment! For more natural semantics.
- Be dependent! For more readable semantics.
- DELPH-IN: Open Source grammars for multiple languages, sharing many assumptions about semantics.
- Grammar engineering perspective: perhaps shallow, but full coverage and cross-linguistic.

Going deeper . . .

1. Deeper compositional semantics: specifying semantics of quantifiers, constructions, tense . . . so that (R)MRS can be converted to alternative (deeper) representations.
2. Symbolic lexical semantics: e.g., word classes, mapping to semantic roles, mapping to WordNet.
3. Distributional lexical semantics combined with DMRS: combined compositional and distributional techniques.
4. Paraphrase and inference test sets for evaluation and regression testing (FraCaS and RTE are a start).

Going broader . . .

1. Distributional techniques require very large scale corpus processing to collect vectors (although, psychologically unrealistic . . .)
2. Semantic search is becoming realistic, but limited utility without lexical semantics.
3. RMRS and DMRS can be produced from fast processing technology. Deep grammars motivate the structures.
4. Construction semantics: so far only really investigated in deep grammars. Essential for properly integrated compositional and distributional semantics.

Acknowledgements

Thanks to: Dan Flickinger, Alex Lascarides, Emily Bender, Stephan Oepen, Francis Bond, other DELPH-INites and Cambridge NLIPers.