English Resource Semantics

Dan Flickinger, Ann Copestake & Woodley Packard

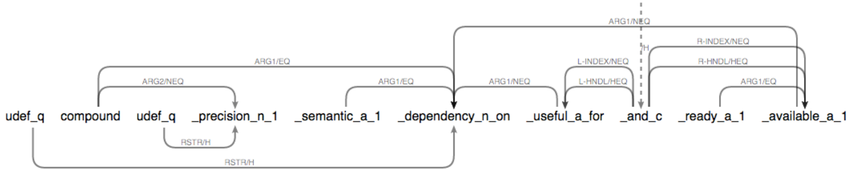
Stanford University, University of Cambridge & University of Washington

24 May 2016

Outline

What is an ERS?

- A rich, spanning, compositionally produced representation of sentence meaning, including ‘who did what to whom’, grammatically required coreference, and grammatically constrained scope information.
- *Precision semantic dependencies are useful and readily available.*



What can I get from an ERS?

- High-precision semantic relations, including long-distance dependencies
- (Partial) information about the scope of scopal operators
- Information about tense, number and similar features

Where does an ERS come from?

- The English Resource Grammar A hand-crafted, broad-coverage, open source, HPSG grammar for English [Flickinger, 2000, Flickinger, 2011]
- Developed over 23 years, against text from varied genres:
  - Meeting scheduling dialogues, tourism brochures, customer email, Wikipedia articles on compling, newspaper text (WSJ), online forum posts, & more
  - But not genre- or domain- dependent.
- Efficient parsing algorithms + maxent parse selection, trained on grammar-derived treebanks [Callmeier, 2002, Oepen et al., 2004, Toutanova et al., 2005]

How can I get ERS?

- ERG-based parsing
  - With PET (included in the LOGON distribution) [Callmeier, 2002]
  - With ACE `http://sweaglesw.org/linguistics/ace/`
- Interactive single-sentence and batch parsing
- Software components with APIs for inclusion in NLP systems

How can I get ERS?

- ERS-annotated sembanks (manually-verified analyses)

Treebank	Sents	Words	Domain
DeepBank	36918	760K	Wall Street Journal newspaper text (as in PTB)
LOGON	11559	160K	Tourism brochures
Verbmobil	11407	90K	Transcribed dialogues
WeScience	9265	170K	100 articles on Comp Ling in Wikipedia
SemCor	2634	50K	Subset of Brown corpus, word-sense-annotated
E-commerce	5413	50K	Customer emails
Misc	3423	40K	Test suites, essay, online forum
Total	80619	1320K	

`http://www.delph-in.net/redwoods`

How can I get ERS?

- In a wide variety of formats:
  - MRS [Copestake, 2002, Copestake et al., 2005]
    Underspecified logical form with variables
  - Dependency MRS (DMRS) [Copestake, 2009]
    Variable-free semantic dependency graph including scope
  - Elementary Dependency Structures (EDS) [Oepen and Lønning, 2006]
    Variable-free semantic dependency graph without scope
  - Bilexical semantic dependencies (DM) [Ivanova et al., 2012]
    Only word-to-word dependencies
- For this tutorial, we will mostly use 'standard' MRS, and sometimes DMRS

Goals of this tutorial

- Set up the ERG-based parsing stack, including preprocessing
- Access ERG Redwoods/DeepBank treebanks in the various export formats
- Interpret ERS representations

Outline

Installation of parser and grammar

- Install VirtualBox from `VirtualBox.org`
- Download the Ubuntu+ERS appliance file from UW
- Run VirtualBox, and from File menu, choose “Import Appliance”
- Choose the ERS appliance file to start the import wizard
- When finished with the wizard, start the new virtual machine

Contents of the package

- ACE parser/generator
- English Resource Grammar (ERG)
- Linguistic User Interface (LUI)
- Full-Forest Treebanker

Running the parser interactively

- In a terminal window in VirtualBox, start the parser:
  `ace -g erg/erg.dat -1lTf`
- Type a simple test sentence, and hit Enter:
  Most fierce dogs chase cats.
- A separate parse tree window pops up.
- Right-click within the parse tree window, and choose “Indexed MRS” to see a compressed view of the ERS.

Alternatively, to get the ERS as a string written to the terminal:

- In the terminal window, start the parser without LUI:
  `ace -g erg/erg.dat -1Tf`
- Type a sentence, and hit Enter:
  Most fierce dogs chase cats.
- The ‘native’ or ‘simple’ ERS output appears in the terminal window

Running the parser in batch mode

- Create a file “mysents.txt” containing a small set of sentences, with one sentence per line
- Run the parser with this filename as an additional argument, and store the results in a file called “myoutput.txt”

  ```
  ace -g erg.dat -1T mysents.txt > myoutput.txt
  ```
- Open the file “myoutput.txt” to see the results of the batch parsing

Example sentence 1

Most house cats are easy for dogs to chase.

$$\langle\, h_1, e_3,$$

$$\left|\begin{array}{l} h_4\text{:\_most\_q}(x_5,\, h_6,\, h_7), \\ h_8\text{:compound}(e_{10},\, x_5,\, x_9), \\ h_{11}\text{:udef\_q}(x_9,\, h_{12},\, h_{13}), \\ h_{14}\text{:\_house\_n\_of}(x_9,\, i_{15}), \\ h_8\text{:\_cat\_n\_1}(x_5), \\ h_2\text{:\_easy\_a\_for}(e_3,\, h_{16},\, x_{17}), \\ h_{18}\text{:udef\_q}(x_{17},\, h_{19},\, h_{20}), \\ h_{21}\text{:\_dog\_n\_1}(x_{17}), \\ h_{22}\text{:\_chase\_v\_1}(e_{23},\, x_{17},\, x_5) \end{array}\right|$$

$$\{\, h_1 =_q h_2,\, h_6 =_q h_8,\, h_{12} =_q h_{14},\, h_{16} =_q h_{22},\, h_{19} =_q h_{21} \,\}\,\rangle$$

Example sentence 2

Which book did the guy who left give to his neighbor?

$$\langle\, h_1, e_3,$$
$$h_4\text{:\_which\_q}(x_5, h_6, h_7),$$
$$h_8\text{:\_book\_n\_of}(x_5, i_9),$$
$$h_{10}\text{:\_the\_q}(x_{12}, h_{13}, h_{11}),$$
$$h_{14}\text{:\_guy\_n\_1}(x_{12}),$$
$$h_{14}\text{:\_leave\_v\_1}(e_{15}, x_{12}, i_{16}),$$
$$h_2\text{:\_give\_v\_1}(e_3, x_{12}, x_5, x_{17}),$$
$$h_{18}\text{:def\_explicit\_q}(x_{17}, h_{20}, h_{19}),$$
$$h_{21}\text{:poss}(e_{23}, x_{17}, x_{22}),$$
$$h_{24}\text{:pronoun\_q}(x_{22}, h_{25}, h_{26}),$$
$$h_{27}\text{:pron}(x_{22}),$$
$$h_{21}\text{:\_neighbor\_n\_1}(x_{17})$$
$$\{\, h_1 =_q h_2,\, h_6 =_q h_8,\, h_{13} =_q h_{14},\, h_{20} =_q h_{21},\, h_{25} =_q h_{27} \,\}\,\rangle$$

Disambiguation alternatives

- Automatic one-best, using maxent model:
  Have the parser only produce the one most likely analysis for each input.
- Manual selection, using ACE Treebanker:
  Have the parser produce all analyses, with the forest presented via discriminats which enable manual selection of the intended analysis.

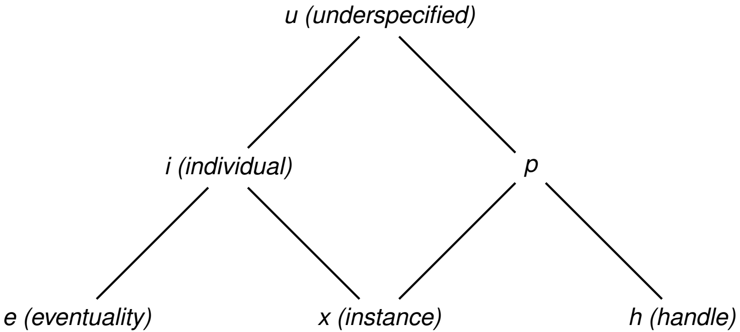
Introduction to ERS formalism

The cat sleeps.

$$\langle\, h_1, e_3, \left| \begin{array}{l} h_4\text{:\_the\_q}(x_6,\, h_7,\, h_5), \\ h_8\text{:\_cat\_n\_1}(x_6), \\ h_2\text{:\_sleep\_v\_1}(e_3,\, x_6) \end{array} \right| \{\, h_1 =_q h_2,\, h_7 =_q h_8 \,\} \,\rangle$$

- Top handle
- Bag of elementary predications
- Scope constraints

ERS variable types



Properties of variables

- Number, person, gender, and individuation on instances

  h_8:_cat_n_1(ARG0 x_6\{PERS *3*, NUM *sg*, GEND *n*, IND *+*\})

- Sentence force, tense, mood, and aspect on eventualities

  h_2:_sleep_v_1(
  ARG0 e_3\{SF *prop*, TENSE *pres*, MOOD *indicative*, PROG -, PERF -\},
  ARG1 x_6)

Elementary predications

Every predication contains

- Predicate name
- Label of type *handle*
- Intrinsic argument of type *individual* as value of ARG0 (except that the ARG0 of quantifiers is not intrinsic)

Predications may contain additional arguments, as values of attributes normally called ARG1, ARG2, ..., though quantifiers and conjunctions, among others, use a richer inventory of attribute names.

Predicates

Surface vs. abstract:

- Naming conventions for **surface** predicates (from lexical entries)
  - Leading underscore
  - Underscore-separated fields
    _lemma_pos_sense
    - *lemma* is orthography of the base form of word in lexicon
    - *pos* draws coarse-grained sense distinction
    - *sense* draws finer-grained sense distinction
- **Abstract** predicates are introduced either via construction, or in decomposed semantics of lexical entries.

Abstract predicate example: Noun-noun compounds

The **police dog** *barked.*

$\langle$,,

| h_8:**compound**(e_{10}, x_6, x_9),
| h_{14}:_police_n_1(x_9),
| h_8:_dog_n_1(x_6) |

{ } $\rangle$

Parameterized predications

Words for named entities introduce in their semantic predication a parameter as the value of a distinguished attribute CARG

We admire Kim greatly.

h_{13}:named(x_9, *Kim*)

Scopal arguments

A predication may have a handle as the value of one of its argument attributes, with a corresponding element in the HCONS list identifying the label of the highest-scoping predication of the argument phrase.

We **know** *that the cat* **didn't sleep**.

$$
\langle\, h_1, e_3,
\left|\begin{array}{l}
h_4\text{:pron}(x_5), \\
h_6\text{:pronoun\_q}(x_5,\, h_7,\, h_8), \\
h_2\text{:\_know\_v\_1}(e_3,\, x_5,\, \mathbf{h}_9), \\
h_{10}\text{:\_the\_q}(x_{12},\, h_{13},\, h_{11}), \\
h_{14}\text{:\_cat\_n\_1}(x_{12}), \\
\mathbf{h}_{15}\text{:neg}(e_{17},\, h_{16}), \\
h_{18}\text{:\_sleep\_v\_1}(e_{19},\, x_{12})
\end{array}\right|
\{\, h_1 =_q h_2,\ h_7 =_q h_4,\ \mathbf{h}_9 =_q \mathbf{h}_{15},\ h_{13} =_q h_{14},\ h_{16} =_q h_{18} \,\}\,\rangle
$$

Basic assumptions for well-formed ERS

- Every predication has a unique ‘intrinsic’ ARG0 (not quantifiers)
- Every instance variable is bound by a quantifier
- Scope resolution results in a set of one or more trees (which can be treated as conventional logical forms)

Outline

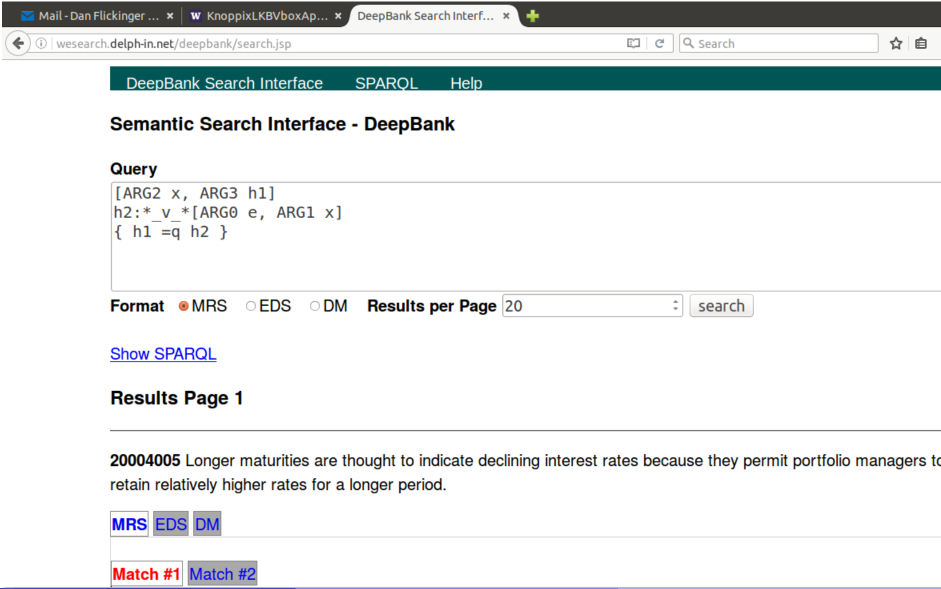
Introduction to the treebanks

- Several collections of text in a variety of domains
- 80,000 sentences, 1.3 million words
- Each sentence parsed with ERG to produce candidate analyses
- Manually disambiguated via syntactic or semantic *discriminants* [Carter, 1997, Oepen et al., 2004]
- Each correct analysis stored with its semantic representation

Semantic search via fingerprints

- Identify elements of ERS to match in treebank
- Regular expressions over predicate names
- Returns sentences and their ERS (in multiple views)
- Useful for exploring ERS in support of feature design

Fingerprint search example: ‘Object’ Control



Fingerprint formalism

- Partial descriptions of ERSs automatically expanded to SPARQL queries for efficient search over RDF encoding of the sembank [Kouylekov and Oepen, 2014].
- Queries consist of one or more EP descriptions, separated by white space, plus optionally HCONS lists
- EP descriptions consist of one or more of:
  - Identifier (label, e.g. `h0`)
  - (Lucene-style pattern over) predicate symbol (e.g. `*_v_*`)
  - List of argument roles with (typed) value identifiers (e.g. `[ARG1 x2]`)
- Repeated identifiers across EPs indicate required reentrancies in the matched ERSs

For more information

- Fuller description of query language:
  `http://sdp.delph-in.net/2015/search.html`
- Sample fingerprints in ERG Semantic Documentation phenomenon pages
  `http://moin.delph-in.net/ErgSemantics`
- Further examples later in this tutorial

Available output formats

- Standard MRS
- Simple MRS
- DMRS
- EDS
- Bi-lexical dependencies
- Direct ERS output from ACE

Standard MRS (terse)

The jungle lion was chasing a small giraffe.

$$\langle\, h_1, e_3,$$
$$\left|\begin{array}{l} h_4\text{:\_the\_q}(x_6,\, h_7,\, h_5), \\ h_8\text{:compound}(e_{10},\, x_6,\, x_9), \\ h_{11}\text{:udef\_q}(x_9,\, h_{12},\, h_{13}), \\ h_{14}\text{:\_jungle\_n\_1}(x_9), \\ h_8\text{:\_lion\_n\_1}(x_6), \\ h_2\text{:\_chase\_v\_1}(e_3,\, x_6,\, x_{15}), \\ h_{16}\text{:\_a\_q}(x_{15},\, h_{18},\, h_{17}), \\ h_{19}\text{:\_small\_a\_1}(e_{20},\, x_{15}), \\ h_{19}\text{:\_giraffe\_n\_1}(x_{15}) \end{array}\right|$$
$$\{\, h_1 =_q h_2,\, h_7 =_q h_8,\, h_{12} =_q h_{14},\, h_{18} =_q h_{19} \,\}\,\rangle$$

Standard MRS with argument roles

The jungle lion was chasing a small giraffe.

$$\langle\, h_1, e_3,$$
$$\left|\begin{array}{l}
h_4\text{:_the_q}(\text{ARG0 } x_6, \text{ RSTR } h_7, \text{ BODY } h_5),\\
h_8\text{:compound}(\text{ARG0 } e_{10}, \text{ ARG1 } x_6, \text{ ARG2 } x_9),\\
h_{11}\text{:udef_q}(\text{ARG0 } x_9, \text{ RSTR } h_{12}, \text{ BODY } h_{13}),\\
h_{14}\text{:_jungle_n_1}(\text{ARG0 } x_9),\\
h_8\text{:_lion_n_1}(\text{ARG0 } x_6),\\
h_2\text{:_chase_v_1}(\textbf{ARG0 } e_3, \textbf{ ARG1 } x_6, \textbf{ ARG2 } x_{15}),\\
h_{16}\text{:_a_q}(\text{ARG0 } x_{15}, \text{ RSTR } h_{18}, \text{ BODY } h_{17}),\\
h_{19}\text{:_small_a_1}(\text{ARG0 } e_{20}, \text{ ARG1 } x_{15}),\\
h_{19}\text{:_giraffe_n_1}(\text{ARG0 } x_{15})
\end{array}\right|$$
$$\{\, h_1 =_q h_2,\ h_7 =_q h_8,\ h_{12} =_q h_{14},\ h_{18} =_q h_{19} \,\}\,\rangle$$

Standard MRS with argument roles and properties

The jungle lion was chasing a small giraffe.

$\langle\, h_1, e_3,$

h_4:_the_q(ARG0 x_6, ,
RSTR h_7, BODY h_5)
h_8:compound(ARG0 e_{10}\{SF *prop*, TENSE *untensed*, MOOD *indic*, PROG -, PERF -\},
ARG1 x_6, ARG2 x_9\{IND +\})
h_{11}:udef_q(ARG0 x_9, RSTR h_{12}, BODY h_{13}),
h_{14}:_jungle_n_1(ARG0 x_9),
h_8:_lion_n_1(ARG0 x_6\{**PERS** *3*, **NUM** *sg*, **IND** +\}),
h_2:_chase_v_1(ARG0 e_3\{**SF** *prop*, **TENSE** *past*, **MOOD** *indic*, **PROG** +, **PERF** -\},
ARG1 x_6, ARG2 x_{15}\{PERS *3*, NUM *sg*, IND +\})
h_{16}:_a_q(ARG0 x_{15}, RSTR h_{18}, BODY h_{17}),
h_{19}:_small_a_1(ARG0 e_{20}\{SF *prop*, TENSE *untensed*, MOOD *indic*\}, ARG1 x_{15}),
h_{19}:_giraffe_n_1(ARG0 x_{15})

$\{\, h_1 =_q h_2,\ h_7 =_q h_8,\ h_{12} =_q h_{14},\ h_{18} =_q h_{19} \,\}\,\rangle$

Standard MRS also with character positions

The jungle lion was chasing a small giraffe.

$$\langle\, h_1, e_3,$$

$$\left|\begin{array}{l}
h_4\text{:_the_q}\langle \mathbf{0}:\mathbf{3}\rangle(\text{ARG0 } x_6,\ \text{RSTR } h_7,\ \text{BODY } h_5),\\
h_8\text{:compound}\langle 4:15\rangle(\text{ARG0 } e_{10},\ \text{ARG1 } x_6,\ \text{ARG2 } x_9),\\
h_{11}\text{:udef_q}\langle 4:10\rangle(\text{ARG0 } x_9,\ \text{RSTR } h_{12},\ \text{BODY } h_{13}),\\
h_{14}\text{:_jungle_n_1}\langle 4:10\rangle(\text{ARG0 } x_9),\\
h_8\text{:_lion_n_1}\langle \mathbf{11}:\mathbf{15}\rangle(\text{ARG0 } x_6),\\
h_2\text{:_chase_v_1}\langle \mathbf{20}:\mathbf{27}\rangle(\text{ARG0 } e_3,\ \text{ARG1 } x_6,\ \text{ARG2 } x_{15}),\\
h_{16}\text{:_a_q}\langle 28:29\rangle(\text{ARG0 } x_{15},\ \text{RSTR } h_{18},\ \text{BODY } h_{17}),\\
h_{19}\text{:_small_a_1}\langle 30:35\rangle(\text{ARG0 } e_{20},\ \text{ARG1 } x_{15}),\\
h_{19}\text{:_giraffe_n_1}\langle 36:44\rangle(\text{ARG0 } x_{15})
\end{array}\right|$$

$$\{\, h_1 =_q h_2,\ h_7 =_q h_8,\ h_{12} =_q h_{14},\ h_{18} =_q h_{19} \,\}\,\rangle$$

Simple MRS

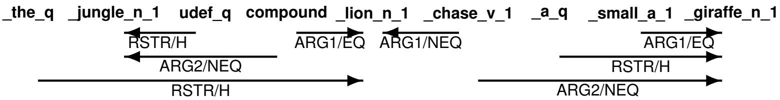
The jungle lion was chasing a small giraffe.

```
[ LTOP  h1 h
  INDEX e3 e
  RELS ⟨ [ LBL h4 h      [ LBL  h8 h       [ LBL  h11 h     [ LBL  h14 h          [ LBL  h8
           PRED _the_q ,   PRED compound ,   PRED udef_q ,    PRED _jungle_n_1 ,    PRED _lion_n_1
           ARG0 x6 x       ARG0 e10 e        ARG0 x9          ARG0 x9 ]             ARG0 x6 ]
           RSTR h7 h       ARG1 x6           RSTR h12 h
           BODY h5 h ]     ARG2 x9 x ]       BODY h13 h ]

         [ LBL  h2 h        [ LBL  h16 h    [ LBL  h19 h          [ LBL  h19
           PRED _chase_v_1 ,  PRED _a_q ,     PRED _small_a_1 ,     PRED _giraffe_n_1
           ARG0 e3            ARG0 x15        ARG0 e20 e            ARG0 x15 ]  ⟩
           ARG1 x6            RSTR h18 h      ARG1 x15 ]
           ARG2 x15 x ]       BODY h17 h ]
  HCONS ⟨ [ HARG h1 ,  [ HARG h7 ,  [ HARG h12 ,  [ HARG h18
            LARG h2 ]    LARG h8 ]    LARG h14 ]    LARG h19 ] ⟩ ]
```

DMRS

The jungle lion was chasing a small giraffe.



EDS: Elementary Dependency Structures

- Reduction to core predicate-argument graph [Oepen et al., 2002];
- 'semantic network': formally (if not linguistically) similar to AMR.

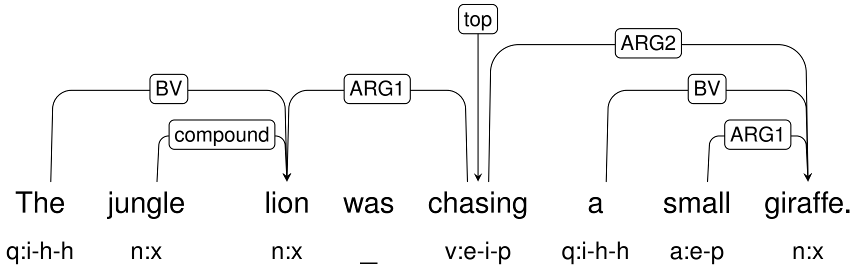
The jungle lion was chasing a small giraffe.

```
(e3 / _chase_v_1
 :ARG1 (x6 / _lion_n_1
        :ARG1-of (e10 / compound
                  :ARG2 (x9 / _jungle_n_1
                         :BV-of (_2 / udef_q)))
        :BV-of (_1 / _the_q))
 :ARG2 (x15 / _giraffe_n_1
        :ARG1-of (e20 / _small_a_1)
        :BV-of (_3 / _a_q)))
```

DM: *Bi-Lexical* Semantic Dependencies

- Lossy reduction of EDS graph: use only surface tokens as nodes;
- construction semantics as edge labels; coarse argument frames;

→ Oepen et al. on Friday: *Comparability of Linguistic Graph Banks*.



ERS output directly from ACE parser

The jungle lion was chasing a small giraffe.

[LTOP: h0
INDEX: e2 [e SF: prop TENSE: past MOOD: indicative PROG: + PERF: -]
RELS: < [_the_q_rel<0:3> LBL: h4 ARG0: x3 [x PERS: 3 NUM: sg IND: +] RSTR: h5 BODY: h6]
[compound_rel<4:15> LBL: h7 ARG0: e8 [e SF: prop TENSE: untensed MOOD: indicative PROG: - PERF: -] ARG1: x3 ARG2: x9 [x IND: +]]
[udef_q_rel<4:10> LBL: h10 ARG0: x9 RSTR: h11 BODY: h12]
["_jungle_n_1_rel"<4:10> LBL: h13 ARG0: x9]
["_lion_n_1_rel"<11:15> LBL: h7 ARG0: x3]
["_chase_v_1_rel"<20:27> LBL: h1 ARG0: e2 ARG1: x3 ARG2: x14 [x PERS: 3 NUM: sg IND: +]]
[_a_q_rel<28:29> LBL: h15 ARG0: x14 RSTR: h16 BODY: h17]
["_small_a_1_rel"<30:35> LBL: h18 ARG0: e19 [e SF: prop TENSE: untensed MOOD: indicative] ARG1: x14]
["_giraffe_n_1_rel"<36:44> LBL: h18 ARG0: x14] >
HCONS: < h0 qeq h1 h5 qeq h7 h11 qeq h13 h16 qeq h18 >]

DMRS XML output

The jungle lion was chasing a small giraffe.


```
<dmrs>
<node nodeid=’10001’ cfrom=’0’ cto=’3’><gpred>_the_q</gpred><sortinfo cvarsort=’x’ pers=’3’ num=’sg’ ind=’plus’/></node>
<node nodeid=’10002’ cfrom=’4’ cto=’15’><gpred>compound</gpred><sortinfo cvarsort=’e’ sf=’prop’ tense=’untensed’
 mood=’indicative’ prog=’minus’ perf=’minus’/></node>
<node nodeid=’10003’ cfrom=’4’ cto=’10’><gpred>udef_q</gpred><sortinfo cvarsort=’x’ ind=’plus’/></node>
<node nodeid=’10004’ cfrom=’4’ cto=’10’><gpred>_jungle_n_1</gpred><sortinfo cvarsort=’x’ ind=’plus’/></node>
<node nodeid=’10005’ cfrom=’11’ cto=’15’><gpred>_lion_n_1</gpred><sortinfo cvarsort=’x’ pers=’3’ num=’sg’ ind=’plus’/></node>
<node nodeid=’10006’ cfrom=’20’ cto=’27’><gpred>_chase_v_1</gpred><sortinfo cvarsort=’e’ sf=’prop’ tense=’past’
 mood=’indicative’ prog=’plus’ perf=’minus’/></node>
<node nodeid=’10007’ cfrom=’28’ cto=’29’><gpred>_a_q</gpred><sortinfo cvarsort=’x’ pers=’3’ num=’sg’ ind=’plus’/></node>
<node nodeid=’10008’ cfrom=’30’ cto=’35’><gpred>_small_a_1</gpred><sortinfo cvarsort=’e’ sf=’prop’ tense=’untensed’
 mood=’indicative’/></node>
<node nodeid=’10009’ cfrom=’36’ cto=’44’><gpred>_giraffe_n_1</gpred><sortinfo cvarsort=’x’ pers=’3’ num=’sg’ ind=’plus’/></node>
<link from=’10001’ to=’10002’><rargname>RSTR</rargname><post>H</post></link>
<link from=’10001’ to=’10005’><rargname>RSTR</rargname><post>H</post></link>
<link from=’10002’ to=’10001’><rargname>ARG1</rargname><post>NEQ</post></link>
<link from=’10002’ to=’10003’><rargname>ARG2</rargname><post>NEQ</post></link>
<link from=’10002’ to=’10005’><rargname>NIL</rargname><post>EQ</post></link>
<link from=’10003’ to=’10004’><rargname>RSTR</rargname><post>H</post></link>
<link from=’10006’ to=’10001’><rargname>ARG1</rargname><post>NEQ</post></link>
<link from=’10006’ to=’10007’><rargname>ARG2</rargname><post>NEQ</post></link>
<link from=’10007’ to=’10008’><rargname>RSTR</rargname><post>H</post></link>
<link from=’10007’ to=’10009’><rargname>RSTR</rargname><post>H</post></link>
<link from=’10008’ to=’10007’><rargname>ARG1</rargname><post>NEQ</post></link>
<link from=’10008’ to=’10009’><rargname>NIL</rargname><post>EQ</post></link>
</dmrs>
```

Inspection and conversion tools

- LUI: inspection
- pyDelphin: conversion and inspection

Interactive disambiguation

Instructions for using ACE Treebanker

- Batch parse a set of sentences
- Invoke the Treebanker with the resulting set of parse forests
- Select a sentence for disambiguation
- Click on each discriminant which is true for the intended analysis
- When the single correct tree remains alone, click “Save”

Outline

Sample linguistic analyses

- For individual phenomena, illustrate how they are represented in ERS
- In aggregate, give a sense of the richness of ERS
- Further documentation for many phenomena available at `http://moin.delph-in.net/ErgSemantics`

Not all surface words are directly reflected in the ERS

- *It does seem as though Kim will both go and rely on Sandy.*

$$
\langle\, h_1, e_3,\ \left|\begin{array}{l}
h_2\text{:\_seem\_v\_to}(e_3, h_4, i_5),\\
h_6\text{:proper\_q}(x_8, h_7, h_9),\\
h_{10}\text{:named}(x_8, \mathit{Kim}),\\
h_{11}\text{:\_go\_v\_1}(e_{12}, x_8),\\
h_{13}\text{:\_and\_c}(e_{14}, h_{11}, e_{12}, h_{15}, e_{16}),\\
h_{15}\text{:\_rely\_v\_on}(e_{16}, x_8, x_{17}),\\
h_{18}\text{:proper\_q}(x_{17}, h_{19}, h_{20}),\\
h_{21}\text{:named}(x_{17}, \mathit{Sandy})
\end{array}\right|\ \{\, h_{19} =_q h_{21},\, h_7 =_q h_{10},\, h_4 =_q h_{13},\, h_1 =_q h_2 \,\}\,\rangle
$$

Sentential negation analyzed in terms of the scopal operator **neg**

The dog didn't bark.

$$\langle\, h_1, e_3,\ \left|\begin{array}{l} h_4{:}\_\mathrm{the\_q}(x_6,\, h_7,\, h_5), \\ h_8{:}\_\mathrm{dog\_n\_1}(x_6), \\ h_2{:}\mathrm{neg}(e_{10},\, h_9), \\ h_{11}{:}\_\mathrm{bark\_v\_1}(e_3,\, x_6) \end{array}\right|\ \{\, h_9 =_q h_{11},\ h_7 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Contracted negation (*didn't, won't*) and independent *not* normalized

- *The dog did not bark.*

$$\langle\, h_1, e_3,$$
$$\left|\begin{array}{l} h_4\text{:\_the\_q}(x_6, h_7, h_5), \\ h_8\text{:\_dog\_n\_1}(x_6), \\ h_2\text{:neg}(e_{10}, h_9), \\ h_{11}\text{:\_bark\_v\_1}(e_3, x_6) \end{array}\right|$$
$$\{\, h_9 =_q h_{11},\ h_7 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Scope of negation fixed by position in parse tree

Sandy knows that Kim probably didn't leave.

$$\langle\, h_1, e_3,$$

$$\left|\begin{array}{l} h_4\text{:proper\_q}(x_6, h_5, h_7), \\ h_8\text{:named}(x_6, \textit{Sandy}), \\ h_2\text{:\_know\_v\_1}(e_3, x_6, h_9), \\ h_{10}\text{:proper\_q}(x_{12}, h_{11}, h_{13}), \\ h_{14}\text{:named}(x_{12}, \textit{Kim}), \\ h_{15}\text{:\_probable\_a\_1}(e_{16}, h_{17}), \\ h_{18}\text{:neg}(e_{20}, h_{19}), \\ h_{21}\text{:\_leave\_v\_1}(e_{22}, x_{12}, p_{23}) \end{array}\right|$$

$$\{\, h_{19} =_q h_{21},\ h_{17} =_q h_{18},\ h_{11} =_q h_{14},\ h_9 =_q h_{15},\ h_5 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

NP negation treated as generalized quantifier

- The body of this quantifier is not fixed by its position in the parse tree

Kim probably saw no dog.

$$\langle\, h_1, e_3,$$
$$\left|\begin{array}{l} h_4\text{:proper\_q}(x_6,\ h_5,\ h_7), \\ h_8\text{:named}(x_6,\ \mathit{Kim}), \\ h_2\text{:\_probable\_a\_1}(e_9,\ h_{10}), \\ h_{11}\text{:\_see\_v\_1}(e_3,\ x_6,\ x_{12}), \\ h_{13}\text{:\_no\_q}(x_{12},\ h_{15},\ h_{14}), \\ h_{16}\text{:\_dog\_n\_1}(x_{12}) \end{array}\right|$$
$$\{\, h_{15} =_q h_{16},\ h_{10} =_q h_{11},\ h_5 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Morphological negation unanalyzed (for now)

That dog is invisible.

$$\langle\, h_1, e_3,\; \left|\begin{array}{l} h_4\text{:\_that\_q\_dem}(x_6, h_7, h_5), \\ h_8\text{:\_dog\_n\_1}(x_6), \\ h_2\text{:\_invisible\_a\_to}(e_3, x_6, i_9) \end{array}\right| \;\{\, h_7 =_q h_8,\; h_1 =_q h_2 \,\}\,\rangle$$

Lexically negative verbs not decomposed

The dog failed to bark.

$$\langle\, h_1, e_3,$$
$$\left|\begin{array}{l} h_4\text{:\_the\_q}(x_6, h_7, h_5), \\ h_8\text{:\_dog\_n\_1}(x_6), \\ h_2\text{:\_fail\_v\_1}(e_3, h_9), \\ h_{10}\text{:\_bark\_v\_1}(e_{11}, x_6) \end{array}\right|$$
$$\{\, h_9 =_q h_{10},\ h_7 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Negation interacts with the analysis of sentence fragments

Not this year.

$$\langle\, h_1, e_3, \left|\begin{array}{l} h_2\text{:unknown}(e_3, u_4), \\ h_2\text{:neg}(e_6, h_5), \\ h_7\text{:loc\_nonsp}(e_8, e_3, x_9), \\ h_{10}\text{:\_this\_q\_dem}(x_9, h_{12}, h_{11}), \\ h_{13}\text{:\_year\_n\_1}(x_9) \end{array}\right| \\ \{\, h_{12} =_q h_{13},\ h_5 =_q h_7,\ h_1 =_q h_2 \,\}\,\rangle$$

Negation fingerprints

```
neg[ARG1 h1]
h2:[ARG0 e]
{ h1 =q h2 }
```

Some predicates establish required coreference relations

Kim persuaded Sandy to leave.

$$\langle\, h_1, e_3,$$

$$\begin{vmatrix} h_4\text{:proper\_q}(x_6, h_5, h_7), \\ h_8\text{:named}(x_6, \textit{Kim}), \\ h_2\text{:\_persuade\_v\_of}(e_3, x_6, x_{10}, h_9), \\ h_{11}\text{:proper\_q}(x_{10}, h_{12}, h_{13}), \\ h_{14}\text{:named}(x_{10}, \textit{Sandy}), \\ h_{15}\text{:\_leave\_v\_1}(e_{16}, x_{10}, p_{17}) \end{vmatrix}$$

$$\{\, h_{12} =_q h_{14},\ h_9 =_q h_{15},\ h_5 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Which arguments are shared is predicate-specific

Kim promised Sandy to leave.

$$\langle\, h_1, e_3,$$
$$\left|\begin{array}{l} h_4\text{:proper\_q}(x_6, h_5, h_7), \\ h_8\text{:named}(x_6, \textit{Kim}), \\ h_2\text{:\_promise\_v\_1}(e_3, x_6, x_{10}, h_9), \\ h_{11}\text{:proper\_q}(x_{10}, h_{12}, h_{13}), \\ h_{14}\text{:named}(x_{10}, \textit{Sandy}), \\ h_{15}\text{:\_leave\_v\_1}(e_{16}, x_6, p_{17}) \end{array}\right|$$
$$\{\, h_{12} =_q h_{14},\ h_9 =_q h_{15},\ h_5 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Not just verbs can be control predicates

Kim is happy to leave.

$$\langle\, h_1, e_3,$$
$$\left|\begin{array}{l} h_4\text{:proper\_q}(x_6,\ h_5,\ h_7), \\ h_8\text{:named}(x_6,\ \mathit{Kim}), \\ h_2\text{:\_happy\_a\_with}(e_3,\ x_6,\ h_9), \\ h_{10}\text{:\_leave\_v\_1}(e_{11},\ x_6,\ p_{12}) \end{array}\right|$$
$$\{\, h_9 =_q h_{10},\ h_5 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Control predicates involve diverse syntactic frames; normalized at the semantic level

Kim prevented Sandy from leaving.

$$\langle\, h_1, e_3,$$
$$\begin{array}{|l|}
h_4\text{:proper_q}(x_6, h_5, h_7), \\
h_8\text{:named}(x_6, \textit{Kim}), \\
h_2\text{:_prevent_v_from}(e_3, x_6, x_{10}, h_9), \\
h_{11}\text{:proper_q}(x_{10}, h_{12}, h_{13}), \\
h_{14}\text{:named}(x_{10}, \textit{Sandy}), \\
h_{15}\text{:_leave_v_1}(e_{16}, x_{10}, p_{17})
\end{array}$$
$$\{\, h_{12} =_q h_{14},\ h_9 =_q h_{15},\ h_5 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Control fingerprints

Example: Subject control. [NB: This is a very general search!]

```
[ARG0 e1, ARG1 x2, ARG3 h3]
h4:[ARG0 e5, ARG1 x2]
{ h3 =q h4 }
```

Lexically Mediated

Complex examples are easy to find.

$$\langle\, h_1, e_3,$$
$$\left|\begin{array}{l} h_4\text{:udef\_q}(x_6,\ h_5,\ h_7), \\ h_8\text{:\_complex\_a\_1}(e_9,\ x_6), \\ h_8\text{:\_example\_n\_of}(x_6,\ i_{10}), \\ h_2\text{:\_easy\_a\_for}(e_3,\ h_{11},\ i_{12}), \\ h_{13}\text{:\_find\_v\_1}(e_{14},\ i_{12},\ x_6) \end{array}\right|$$
$$\{\, h_{11} =_q h_{13},\ h_5 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Relative clauses

The cat whose collar you thought I found escaped.

$\langle h_1, e_3,$

h_4:_the_q(x_6, h_7, h_5),
h_8:_cat_n_1(x_6),
h_9:def_explicit_q(x_{11}, h_{12}, h_{10}),
h_{13}:poss(e_{14}, x_{11}, x_6),
h_{15}:_collar_n_1(x_{11}),
h_{16}:pron(x_{17}),
h_{18}:pronoun_q(x_{17}, h_{19}, h_{20}),
h_8:_think_v_1(e_{21}, x_{17}, h_{23}, i_{22}),
h_{24}:pron(x_{25}),
h_{26}:pronoun_q(x_{25}, h_{27}, h_{28}),
h_{29}:_find_v_1(e_{30}, x_{25}, x_{11}),
h_2:_escape_v_1(e_3, x_6, p_{31})

$\{ h_{27} =_q h_{24}, h_{23} =_q h_{29}, h_{19} =_q h_{16}, h_{12} =_q h_{15}, h_7 =_q h_8, h_1 =_q h_2 \} \rangle$

Right Node Raising

PCBs move into and go out of the machine automatically.

$$\langle\, h_1, e_{10},$$
$$\begin{array}{|l|}
h_4\text{:udef_q}(x_6,\, h_5,\, h_7), \\
h_8\text{:_pcbs/nns_u_unknown}(x_6), \\
h_9\text{:_move_v_1}(e_{10},\, x_6), \\
h_9\text{:_into_p}(e_{11},\, e_{10},\, x_{12}), \\
h_2\text{:_and_c}(e_3,\, h_9,\, e_{10},\, h_{14},\, e_{13}), \\
h_{14}\text{:_go_v_1}(e_{13},\, x_6), \\
h_{14}\text{:_out+of_p_dir}(e_{15},\, e_{13},\, x_{12}), \\
h_{16}\text{:_the_q}(x_{12},\, h_{18},\, h_{17}), \\
h_{19}\text{:_machine_n_1}(x_{12}), \\
h_2\text{:_automatic_a_1}(e_{20},\, e_3)
\end{array}$$
$$\{\, h_{18} =_q h_{19},\, h_5 =_q h_8,\, h_1 =_q h_2 \,\}\,\rangle$$

Fingerprints?

- Long-Distance Dependencies do not constitute a semantic phenomenon
- There are no characteristic patterns in the ERS reflecting them
- Rather, dependencies which are long-distance in the syntax appear ordinary in the ERS

Outline

Parser settings

- Root symbols
- Preprocessing
- Unknown-word handling
- Disambiguation models
- Resource limits

Robust processing: Three methods

- Csaw:
  Using probabilistic context-free grammar trained on ERG best-one analyses of 50 million sentences from English Wikipedia
  (Based on previous work on Jigsaw by Yi Zhang)
- Bridging:
  Using very general binary bridging constructions added to the ERG which build non-licensed phrases
- Mal-rules:
  Using error-specific constructions added to the ERG to admit words or phrases which are predicatbly ill-formed, with correct semantics

Efficiency vs. Precision in Parsing

- Parameters to control resource limits
  - **Time**: maximum number of seconds to use per sentence
  - **Memory**: maximum number of bytes to use for building the packed parse forest and for unpacking
  - **Number of analyses**: only unpack part of the forest
- Ubertagging
  Prune the candidate lexical items for each token in a sentence before invoking the parser, using a statistical model trained on Redwoods and DeepBank
  [Dridan, 2013]

Outline

Efficiency vs. Precision

- Word senses for finer-grained semantic representations
- More derivational morphology (e.g. semi-productive deverbal nouns)
- Support for coreference within and across sentence boundaries

Information Structure

- Addition of ICONS attribute for constraints on pairs of individuals
- Now used for structurally imposed constraints on *topic* and *focus*
- Passivized subjects (topic) and “topicalized” phrases (focus)
- [Song and Bender, 2012]

Outline

Sample applications using ERS

- Scope of negation
- Logic to English (generation)
- Robot blocks world

Task

- *SEM2012 Task 1: Identify negation *cues* and their associated *scopes* [Morante and Blanco, 2012]
- Ex: {The German} was sent for but professed to {know} ⟨nothing⟩ {of the matter}.
- Relevant for sentiment analysis, IE, MT, and many other applications

Contribution of ERS

- Operator scope is a first-class notion in ERS
- Scopes discontinuous in the surface string form subgraphs of ERS
- Characterization links facilitate mapping out to string-based annotations

Challenges

- Shared task notions of negation and scope don't directly match those in ERS
- Target annotations include semantically empty elements
- Dialect differences (early 1900s British English v. contemporary American English)

Approach

- Use cue detection from [Read et al., 2012]
- Map cue identified in string to EP in ERS
- 'Crawl' the ERS graph from the cue, according to the type of cue and type of EP encountered
- Use EP characterization and syntactic parse tree to map scope to substrings
- Fall back to [Read et al., 2012] if no parse or top ranked parse has a score of < 0.5

Approach

{The German} was sent for but professed to {know} ⟨nothing⟩ {of the matter}.

$$\langle\, h_1, e_3,$$
$$h_4\text{:\_the\_q}(x_6, h_7, h_5),$$
$$h_8\text{:named}(x_6, \textit{German}),$$
$$h_9\text{:\_send\_v\_for}(e_{10}, i_{11}, x_6),$$
$$h_9\text{:parg\_d}(e_{12}, e_{10}, x_6),$$
$$h_2\text{:\_but\_c}(e_3, h_9, e_{10}, h_{14}, e_{13}),$$
$$h_{14}\text{:\_profess\_v\_to}(e_{13}, x_6, h_{15}),$$
$$h_{16}\text{:\_know\_v\_1}(e_{17}, x_6, x_{18}),$$
$$h_{19}\text{:thing}(x_{18}),$$
$$h_{20}\text{:\_no\_q}(x_{18}, h_{21}, h_{22}),$$
$$h_{19}\text{:\_of\_p}(e_{23}, x_{18}, x_{24}),$$
$$h_{25}\text{:\_the\_q}(x_{24}, h_{27}, h_{26}),$$
$$h_{28}\text{:\_matter\_n\_of}(x_{24}, i_{29})$$
$$\{\, h_{27} =_q h_{28},\, h_{21} =_q h_{19},\, h_{15} =_q h_{16},\, h_7 =_q h_8,\, h_1 =_q h_2 \,\}\,\rangle$$

Approach

{The German} was sent for but professed to {know} ⟨nothing⟩ {of the matter}.

$$
\langle h_1, e_3, \left|
\begin{array}{l}
h_4\text{:\_the\_q}(x_6, h_7, h_5),\\
h_8\text{:named}(x_6, \textit{German}),\\
h_9\text{:\_send\_v\_for}(e_{10}, i_{11}, x_6),\\
h_9\text{:parg\_d}(e_{12}, e_{10}, x_6),\\
h_2\text{:\_but\_c}(e_3, h_9, e_{10}, h_{14}, e_{13}),\\
h_{14}\text{:\_profess\_v\_to}(e_{13}, x_6, h_{15}),\\
h_{16}\text{:\_know\_v\_1}(e_{17}, x_6, x_{18}),\\
h_{19}\text{:thing}(x_{18}),\\
h_{20}\text{:\_no\_q}(x_{18}, h_{21}, h_{22}),\\
h_{19}\text{:\_of\_p}(e_{23}, x_{18}, x_{24}),\\
h_{25}\text{:\_the\_q}(x_{24}, h_{27}, h_{26}),\\
h_{28}\text{:\_matter\_n\_of}(x_{24}, i_{29})
\end{array}
\right|
\{ h_{27} =_q h_{28}, h_{21} =_q h_{19}, h_{15} =_q h_{16}, h_7 =_q h_8, h_1 =_q h_2 \} \rangle
$$

Approach

{The German} was sent for but professed to {know} ⟨nothing⟩ {of the matter}.

$$\langle\, h_1, e_3,$$

$$\left|\begin{array}{l} h_4\text{:\_the\_q}(x_6,\ h_7,\ h_5), \\ h_8\text{:named}(x_6,\ German), \\ h_9\text{:\_send\_v\_for}(e_{10},\ i_{11},\ x_6), \\ h_9\text{:parg\_d}(e_{12},\ e_{10},\ x_6), \\ h_2\text{:\_but\_c}(e_3,\ h_9,\ e_{10},\ h_{14},\ e_{13}), \\ h_{14}\text{:\_profess\_v\_to}(e_{13},\ x_6,\ h_{15}), \\ h_{16}\text{:\_know\_v\_1}(e_{17},\ x_6,\ x_{18}), \\ h_{19}\text{:thing}(x_{18}), \\ h_{20}\text{:\_no\_q}(x_{18},\ h_{21},\ h_{22}), \\ h_{19}\text{:\_of\_p}(e_{23},\ x_{18},\ x_{24}), \\ h_{25}\text{:\_the\_q}(x_{24},\ h_{27},\ h_{26}), \\ h_{28}\text{:\_matter\_n\_of}(x_{24},\ i_{29}) \end{array}\right|$$

$$\{\, h_{27} =_q h_{28},\ h_{21} =_q h_{19},\ h_{15} =_q h_{16},\ h_7 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Approach

{The German} was sent for but professed to {know} ⟨nothing⟩ {of the matter}.

$$
\langle\, h_1, e_3,
$$
$$
\left|\begin{array}{l}
h_4{:}\text{\_the\_q}(x_6,\, h_7,\, h_5), \\
h_8{:}\text{named}(x_6,\, \textit{German}), \\
h_9{:}\text{\_send\_v\_for}(e_{10},\, i_{11},\, x_6), \\
h_9{:}\text{parg\_d}(e_{12},\, e_{10},\, x_6), \\
h_2{:}\text{\_but\_c}(e_3,\, h_9,\, e_{10},\, h_{14},\, e_{13}), \\
h_{14}{:}\text{\_profess\_v\_to}(e_{13},\, x_6,\, h_{15}), \\
h_{16}{:}\text{\_know\_v\_1}(e_{17},\, x_6,\, x_{18}), \\
h_{19}{:}\text{thing}(x_{18}), \\
h_{20}{:}\text{\_no\_q}(x_{18},\, h_{21},\, h_{22}), \\
h_{19}{:}\text{\_of\_p}(e_{23},\, x_{18},\, x_{24}), \\
h_{25}{:}\text{\_the\_q}(x_{24},\, h_{27},\, h_{26}), \\
h_{28}{:}\text{\_matter\_n\_of}(x_{24},\, i_{29})
\end{array}\right|
$$
$$
\{\, h_{27} =_q h_{28},\, h_{21} =_q h_{19},\, h_{15} =_q h_{16},\, h_7 =_q h_8,\, h_1 =_q h_2 \,\}\,\rangle
$$

Approach

{The German} was sent for but professed to {know} ⟨nothing⟩ {of the matter}.

$$\langle\, h_1, e_3,\ \Big|\ h_4{:}\_the\_q(x_6, h_7, h_5),\ h_8{:}named(x_6, German),\ h_9{:}\_send\_v\_for(e_{10}, i_{11}, x_6),\ h_9{:}parg\_d(e_{12}, e_{10}, x_6),\ h_2{:}\_but\_c(e_3, h_9, e_{10}, h_{14}, e_{13}),\ h_{14}{:}\_profess\_v\_to(e_{13}, x_6, h_{15}),\ h_{16}{:}\_know\_v\_1(e_{17}, x_6, x_{18}),\ h_{19}{:}thing(x_{18}),\ h_{20}{:}\_no\_q(x_{18}, h_{21}, h_{22}),\ h_{19}{:}\_of\_p(e_{23}, x_{18}, x_{24}),\ h_{25}{:}\_the\_q(x_{24}, h_{27}, h_{26}),\ h_{28}{:}\_matter\_n\_of(x_{24}, i_{29})\ \Big|\ \{\, h_{27} =_q h_{28},\ h_{21} =_q h_{19},\ h_{15} =_q h_{16},\ h_7 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle$$

Approach

{The German} was sent for but professed to {know} ⟨nothing⟩ {of the matter}.

$$
\langle\, h_1, e_3,\
\left|\begin{array}{l}
h_4\text{:\_the\_q}(x_6,\ h_7,\ h_5),\\
h_8\text{:named}(x_6,\ \textit{German}),\\
h_9\text{:\_send\_v\_for}(e_{10},\ i_{11},\ x_6),\\
h_9\text{:parg\_d}(e_{12},\ e_{10},\ x_6),\\
h_2\text{:\_but\_c}(e_3,\ h_9,\ e_{10},\ h_{14},\ e_{13}),\\
h_{14}\text{:\_profess\_v\_to}(e_{13},\ x_6,\ h_{15}),\\
h_{16}\text{:\_know\_v\_1}(e_{17},\ x_6,\ x_{18}),\\
h_{19}\text{:thing}(x_{18}),\\
h_{20}\text{:\_no\_q}(x_{18},\ h_{21},\ h_{22}),\\
h_{19}\text{:\_of\_p}(e_{23},\ x_{18},\ x_{24}),\\
h_{25}\text{:\_the\_q}(x_{24},\ h_{27},\ h_{26}),\\
h_{28}\text{:\_matter\_n\_of}(x_{24},\ i_{29})
\end{array}\right|
\{\, h_{27} =_q h_{28},\ h_{21} =_q h_{19},\ h_{15} =_q h_{16},\ h_7 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle
$$

Approach

{The German} was sent for but professed to {know} ⟨nothing⟩ {of the matter}.

$$
\begin{array}{l}
\langle\, h_1, e_3, \\
\quad h_4\text{:\_the\_q}\langle 0\!:\!3\rangle(x_6, h_7, h_5), \\
\quad h_8\text{:named}\langle 4\!:\!10\rangle(x_6, \mathit{German}), \\
\quad h_9\text{:\_send\_v\_for}\langle 15\!:\!19\rangle(e_{10}, i_{11}, x_6), \\
\quad h_9\text{:parg\_d}\langle 15\!:\!19\rangle(e_{12}, e_{10}, x_6), \\
\quad h_2\text{:\_but\_c}\langle 24\!:\!27\rangle(e_3, h_9, e_{10}, h_{14}, e_{13}), \\
\quad h_{14}\text{:\_profess\_v\_to}\langle 28\!:\!37\rangle(e_{13}, x_6, h_{15}), \\
\quad h_{16}\text{:\_know\_v\_1}\langle 41\!:\!45\rangle(e_{17}, x_6, x_{18}), \\
\quad h_{19}\text{:thing}\langle 46\!:\!53\rangle(x_{18}), \\
\quad h_{20}\text{:\_no\_q}\langle 46\!:\!53\rangle(x_{18}, h_{21}, h_{22}), \\
\quad h_{19}\text{:\_of\_p}\langle 54\!:\!56\rangle(e_{23}, x_{18}, x_{24}), \\
\quad h_{25}\text{:\_the\_q}\langle 57\!:\!60\rangle(x_{24}, h_{27}, h_{26}), \\
\quad h_{28}\text{:\_matter\_n\_of}\langle 61\!:\!68\rangle(x_{24}, i_{29}) \\
\{\, h_{27} =_q h_{28},\ h_{21} =_q h_{19},\ h_{15} =_q h_{16},\ h_7 =_q h_8,\ h_1 =_q h_2 \,\}\,\rangle
\end{array}
$$

Results

- As of 2014, state of the art for this task

	Scopes			Tokens		
Method	**Prec**	**Rec**	**F_1**	**Prec**	**Rec**	**F_1**
Read et al 2012	87.4	61.5	72.2	82.0	**88.8**	85.3
ERS Crawler	**87.8**	43.4	58.1	78.8	66.7	72.2
Combined System	87.6	**62.7**	**73.1**	**82.6**	88.5	**85.4**

Data/software for reproducibility:
`http://www.delph-in.net/crawler/`

Task: Generate English from First-Order Logic

- Online course on introductory logic
  Textbook: Barker-Plummer, Barwise and Etchemendy, *Language, Proof, and Logic, 2nd Edition*
- Students are presented with an English statement
- Their task: Produce an equivalent first-order logic expression
- Our task: Generate English paraphrases of an FOL
  - Produce English for auto-generated course FOL to start task
  - Restate student's incorrect FOL as English for instruction

Our method

- Convert FOL to skeletal ERS (Python script)
- Inflate skeletal ERS to full ERS using ACE ‘transfer’ rules
- Apply richer set of transfer rules using ACE to produce paraphrase ERSs
- Generate from each of these paraphrase ERSs using ACE
- Select one of these outputs to present to the student

Example: FOL to English

First, convert FOL to skeletal ERS via Python script:

`large(a)&large(b)`

```
[ LTOP: h1
  INDEX: e1
  RELS: < [ "name" LBL: h3 ARG0: x1 CARG: "A" ]
          [ "large" LBL: h4 ARG0: e2 ARG1: x1 ]
          [ "name" LBL: h5 ARG0: x2 CARG: "B" ]
          [ "large" LBL: h6 ARG0: e3 ARG1: x2 ]
          [ "and" LBL: h2 ARG0: e1 L-INDEX: e2 R-INDEX: e3 ] > ]
```

‘Inflated’ ERS for `large(a)&large(b)`

Next, apply transfer rules to fill in missing elements (quantifiers, variable properties, ERS predicate names, handle constraints):

```
[ LTOP: h20
  INDEX: e13 [ e SORT: collective SF: prop TENSE: pres PERF: - ]
  RELS: < [ named LBL: h5 ARG0: x10 [ x PERS: 3 NUM: sg ] CARG: "A" ]
          [ named LBL: h9 ARG0: x11 [ x PERS: 3 NUM: sg ] CARG: "B" ]
          [ proper_q LBL: h2 ARG0: x10 RSTR: h3 BODY: h4 ]
          [ proper_q LBL: h6 ARG0: x11 RSTR: h7 BODY: h8 ]
          [ _and_c LBL: h12 ARG0: e13 L-INDEX: e14 R-INDEX: e15 L-HNDL: h16 R-HNDL: h17 ]
          [ _large_a_1 LBL: h18 ARG0: e14 [ e SF: prop TENSE: pres PERF: - ] ARG1: x10 ]
          [ _large_a_1 LBL: h19 ARG0: e15 [ e SF: prop TENSE: pres PERF: - ] ARG1: x11 ] >
  HCONS: < h3 qeq h5 h7 qeq h9 h16 qeq h18 h17 qeq h19 > ]
```

Paraphrase transfer rules

Then apply paraphrase transfer rules to produce multiple ERSs, and present each ERS to the generator.

Example rule for *B is large and C is large* → *B and C are large*

```
coord_subject_rule := openproof_omtr &
[ CONTEXT.RELS < [ PRED named, ARG0 x3 ],
                 [ PRED named, ARG0 x6 ] >,
  INPUT.RELS   < [ PRED _and_c, ARG0 e10, L-INDEX e2, R-INDEX e5 ],
                 [ PRED pred1, ARG0 e2, ARG1 x3 ],
                 [ PRED pred1, ARG0 e5, ARG1 x6 ] >
  OUTPUT.RELS  < [ PRED _and_c, ARG0 x10, L-INDEX x3, R-INDEX x6 ],
                 [ PRED pred1, ARG0 e10, ARG1 x10 ] >
```

Generated paraphrases

```
large(a)&large(b)
```

A is large and B is large.
A is large, and B is large.
A and B are large.
Both A and B are large.

A second example

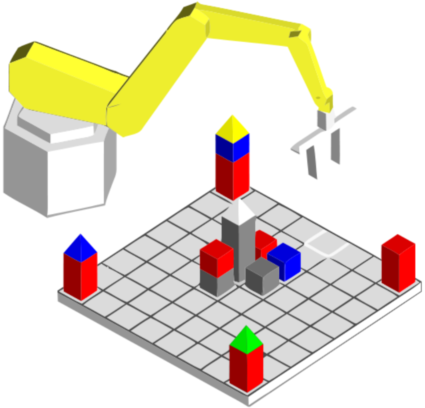
```
(cube(a)&cube(b))->leftof(a,b)
```

If A is a cube and B is a cube, then A is to the left of B.
If A and B are cubes, then A is to the left of B.
If both A and B are cubes, then A is to the left of B.
If A and B are both cubes, then A is to the left of B.
A is to the left of B, if A and B are both cubes.

Task: Interpreting robotic spatial commands

- Semeval-2014 Shared Task 6
- Parse English commands to change states in a ‘blocks’ world
- Generate corresponding Robot Control Language statements
- Evaluate based on correct altered state of the game board
- [Packard, 2014]

Game board illustration



Example of robot command

Pick up the turquoise pyramid standing over a white cube

$$
\langle\, h_0, e_2,\\
\left|\begin{array}{l}
h_4\text{:pronoun\_q}(x_3, h_5, h_6),\\
h_7\text{:pron}(x_3),\\
h_1\text{:\_pick\_v\_up}(e_2, x_3, x_8),\\
h_9\text{:\_the\_q}(x_8, h_{10}, h_{11}),\\
h_{12}\text{:\_turquoise\_a\_1}(e_{13}, x_8),\\
h_{12}\text{:\_pyramid\_n\_1}(x_8),\\
h_{12}\text{:\_stand\_v\_1}(e_{14}, x_8),\\
h_{12}\text{:\_over\_p}(e_{15}, e_{14}, x_{16}),\\
h_{17}\text{:\_a\_q}(x_{16}, h_{18}, h_{19}),\\
h_{20}\text{:\_white\_a\_1}(e_{21}, x_{16}),\\
h_{20}\text{:\_cube\_n\_1}(x_{16})
\end{array}\right|\\
\{\, h_{18} =_q h_{20},\ h_{10} =_q h_{12},\ h_5 =_q h_7,\ h_0 =_q h_1 \,\}\,\rangle
$$

Generated robot command from ERS

Pick up the turquoise pyramid standing over a white cube

Corresponding RCL statement:

```
(event:
  (action: take)
  (entity:
    (id: 1)
    (color: cyan)
    (type: prism)
    (spatial-relation:
      (relation: above)
      (entity:
        (color: white)
        (type: cube))
```

Acknowledgements

We are grateful to Emily Bender and Stephan Oepen for their considerable help in preparing these materials.

References I

Callmeier, U. (2002).
Preprocessing and encoding techniques in PET.
In Oepen, S., Flickinger, D., Tsujii, J., and Uszkoreit, H., editors, *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing*, page 127 – 140. CSLI Publications, Stanford, CA.

Carter, D. (1997).
The TreeBanker. A tool for supervised training of parsed corpora.
In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, page 9 – 15, Madrid, Spain.

Copestake, A. (2002).
Implementing Typed Feature Structure Grammars.
CSLI Lecture Notes. Center for the Study of Language and Information, Stanford,California.

Copestake, A. (2009).
Slacker semantics. Why superficiality, dependency and avoidance of commitment can be the right way to go.
In *Proceedings of the 12th Meeting of the European Chapter of the Association for Computational Linguistics*, page 1 – 9, Athens, Greece.

References II

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. A. (2005).
Minimal Recursion Semantics. An introduction.
Research on Language and Computation, 3(4):281 – 332.

Dridan, R. (2013).
Ubertagging. Joint segmentation and supertagging for English.
In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Seattle, WA, USA.

Flickinger, D. (2000).
On building a more efficient grammar by exploiting types.
Natural Language Engineering, 6 (1):15 – 28.

Flickinger, D. (2011).
Accuracy vs. robustness in grammar engineering.
In Bender, E. M. and Arnold, J. E., editors, *Language from a Cognitive Perspective: Grammar, Usage, and Processing*, page 31 – 50. Stanford: CSLI Publications.

Ivanova, A., Oepen, S., Øvrelid, L., and Flickinger, D. (2012).
Who did what to whom? A contrastive study of syntacto-semantic dependencies.
In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 2–11, Jeju, Republic of Korea.

References III

Kouylekov, M. and Oepen, S. (2014).
Semantic technologies for querying linguistic annotations. An experiment focusing on graph-structured data.
In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, page 4331 – 4336, Reykjavik, Iceland.

Morante, R. and Blanco, E. (2012).
*SEM 2012 shared task: Resolving the scope and focus of negation.
In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics*, page 265 – 274, Montréal, Canada.

Oepen, S., Flickinger, D., Toutanova, K., and Manning, C. D. (2002).
Lingo Redwoods. A rich and dynamic treebank for HPSG.
In *Proceedings of the 1st International Workshop on Treebanks and Linguistic Theories*, page 139 – 149, Sozopol, Bulgaria.

Oepen, S., Flickinger, D., Toutanova, K., and Manning, C. D. (2004).
LinGO Redwoods. A rich and dynamic treebank for HPSG.
Research on Language and Computation, 2(4):575 – 596.

References IV

Oepen, S. and Lønning, J. T. (2006).
Discriminant-based MRS banking.
In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, page 1250 – 1255, Genoa, Italy.

Packard, W. (2014).
UW-MRS: Leveraging a deep grammar for robotic spatial commands.
In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, page 812 – 816, Dublin, Ireland.

Read, J., Velldal, E., Øvrelid, L., and Oepen, S. (2012).
UiO1: constituent-based discriminative ranking for negation resolution.
In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics*, page 310 – 318, Montréal, Canada.

Song, S. and Bender, E. M. (2012).
Individual constraints for information structure.
In *Proceedings of the 19th International Conference on Head-Driven Phrase Structure Grammar (HPSG 2012)*, page 330 – 348, Daejeon, Korea.

Toutanova, K., Manning, C. D., Flickinger, D., and Oepen, S. (2005).
Stochastic HPSG Parse Disambiguation using the Redwoods Corpus.
Research on Language and Computation, 3:83 – 105.