DISCOURSE REPRESENTATION THEORY
What is DRT?

- Discourse Representation Theory
  - 1980’s Hans Kamp
- A particular way of dealing with semantics and logic in natural language
- Emphasis is on dynamic vs. static view on natural language semantics
  - Discourse is a sequence of sentences
  - We want to represent everything that emerges in discourse (common ground, inferences, presuppositions, accommodation, anaphor, etc.)
- Fully (interpretive) semantic representation
What is DRT? (2)

- **Goals:**
  - To represent the content: meaning of the text so far
  - To represent the context: anaphor across sentences, …
  - To represent utterances in a way that emphasizes their logical structure
  - To allow language processing of phenomena that depend on logical structure.

- **Data types:** Discourse representation structures (DRS’s)
Issue: scope of operators

- **A woman** walks.  **She** smokes.

\[
\exists x (\text{woman}(x) \land \text{walk}(x)) \quad \text{smoke}(x)
\]

\[
\exists x (\text{woman}(x) \land \text{walk}(x) \land \text{smoke}(x))
\]

Need to expand the scope of the existential quantifier.
DRS components

• Discourse referents:
  • Variables, representing objects; anything which can serve as the antecedent for an anaphor

• Conditions:
  • Represent properties and relationships

• Examples:
  Thomas(u)
  v married w
DRS’s

DRT Basics

1. new discourse: empty box

2. Fred owns "Linguistic Semantics"

universe (disc. referents)

set of conditions
Relative clauses

Fred owns a book which Mary adores.

Diagram:

\[ x \in Z 
\text{Fred}(x) 
\text{book}(y) 
\text{Mary}(z) 
\text{adores}(y) 
\text{owns}(z, y) \]
Connectives

If Fred owns LS, he uses it.
Quantification (universal)

A dog barked.

<table>
<thead>
<tr>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog(x)</td>
</tr>
<tr>
<td>bark(x)</td>
</tr>
</tbody>
</table>

Every dog barked.

<table>
<thead>
<tr>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog(x)</td>
</tr>
</tbody>
</table>

\[ \Rightarrow \]

| bark(x) |
More universal quantification

\( d \)
\(`\text{"The-Dream-of Gerontius"}(d)`
\( \text{boring}(d) \)

\( e \)
\( \text{Elgar}(e) \)

\( x \)
\(`\text{"choral-work-of"}(x,e)`
\( \forall x \)
\( \text{boring}(x) \)
Negation

Jones does not own a Porsche.
Disjunction
Either Fred doesn't own a car, or (else) he hides it.
Genericity/typicality
Intersentential anaphor

- A woman walks. She smokes.

\[
\begin{array}{|c|}
\hline
x, y \\
\hline
\text{woman}(x) \\
\text{walk}(x) \\
y = x \\
\text{smoke}(y) \\
\hline
\end{array}
\]

- discourse referent \( x, y \), in the top part of the box.
- conditions upon these discourse referents in the lower part of the box.

Pronouns/Anaphor: A woman sneezes. She is sick.
Intersentential anaphor

• Mary ordered a milk shake,
  John tasted it.

<table>
<thead>
<tr>
<th>$x$, $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = \text{Mary}$</td>
</tr>
<tr>
<td>$\text{order}(x, y)$</td>
</tr>
<tr>
<td>$\text{shake}(y)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x$, $y$, $u$, $v$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = \text{Mary}$</td>
</tr>
<tr>
<td>$\text{order}(x, y)$</td>
</tr>
<tr>
<td>$\text{shake}(y)$</td>
</tr>
<tr>
<td>$u = \text{John}$</td>
</tr>
<tr>
<td>$\text{taste}(u, v)$</td>
</tr>
<tr>
<td>$v = y$</td>
</tr>
</tbody>
</table>

The discourse referent $y$ is accessible for discourse referent $v$.

An anaphoric link between it and milk shake is allowed.

0. Fred owns LS. It fascinates him.

<table>
<thead>
<tr>
<th>$x$, $y$, $u$, $v$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = \text{Fred}$</td>
</tr>
<tr>
<td>$\text{LS}(x, y)$</td>
</tr>
<tr>
<td>$x \text{ owns } y$</td>
</tr>
<tr>
<td>$u = y$</td>
</tr>
<tr>
<td>$u \text{ fascinates } v$</td>
</tr>
</tbody>
</table>

(Cf. similarly for ... a Porsche... )
Referents and accessibility

A dog barked. It was happy.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog(x)</td>
<td>bark(x)</td>
</tr>
<tr>
<td>happy(y)</td>
<td>y = x</td>
</tr>
</tbody>
</table>

Discourse referent x is accessible

Every dog barked. ?It was happy.

<table>
<thead>
<tr>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
</tr>
<tr>
<td>dog(x)</td>
</tr>
<tr>
<td>happy(y)</td>
</tr>
<tr>
<td>y = ??</td>
</tr>
</tbody>
</table>

Discourse referent x is not accessible
Inaccessibility

Inaccessibility: Jones does not own a Porsche. He likes it.

\[
\begin{align*}
\exists x & \exists y \\
\text{Jones}(x) & \quad \text{Porsche}(y) \\
\exists z & \quad x \text{ owns } y \\
& \quad z = x \\
& \quad u = y \\
& \quad \exists z \text{ liked } u
\end{align*}
\]

is not a valid ORS
Sned doesn't own L.S. He likes it (however).

\[ x \quad y \quad z \quad n \]

\[
\text{Jones}(x) \\
\text{Lives}(y) \\
\text{Owns}(x, y) \\
\text{Likes}(z, y) \\
z = x \\
v = y \\
z \text{ likes } y
\]
Equivalence

\[
x \quad \text{delegate}(x) \quad \text{arrive}(x) \\
\]

\[
x \quad z \\
\text{delegate}(x) \\
\text{arrive}(x) \\
\]

\[
y \\
\text{delegate}(y) \\
\text{arrive}(y) \\
\]

\[
\Rightarrow \\
x = y \\
\]

\[
\text{register}(z) \\
\]

\[
z = x \\
\]

\[
x \quad y \\
\text{delegate}(x) \\
\text{arrive}(x) \\
\text{register}(y) \\
y = x \\
\]

\[
x \\
\text{predro}(x) \\
\]

\[
y \\
\text{donkey}(y) \\
\text{own}(x,y) \\
\]

\[
\Rightarrow \\
z \quad w \\
\text{beat}(z,w) \\
z = x \\
w = y \\
\]

\[
\text{delegate}(x) \\
\]

\[
\Rightarrow \\
\text{arrive}(x) \\
\]
Mapping from syntax to semantics

\[ S \rightarrow \begin{array}{c} a \text{ delegate} \end{array} \quad \rightarrow \quad \begin{array}{c} x \\
\text{delegate}(x) \\
\text{arrived} \end{array} \]

\[ S \rightarrow \begin{array}{c} y \end{array} \quad \rightarrow \quad \begin{array}{c} y \\
\text{arrived} \end{array} \]

\[ S \rightarrow \begin{array}{c} x \end{array} \quad \rightarrow \quad \begin{array}{c} x \\
\text{delegate}(x) \\
\text{arrive}(x) \end{array} \]

\[ S \rightarrow \begin{array}{c} x \end{array} \quad \rightarrow \quad \begin{array}{c} x \\
\text{register}(x) \end{array} \]

\[ S \rightarrow \begin{array}{c} y \end{array} \quad \rightarrow \quad \begin{array}{c} y \\
\text{registered} \end{array} \]

\[ S \rightarrow \begin{array}{c} y \end{array} \quad \rightarrow \quad \begin{array}{c} y \\
\text{equal}(x) \end{array} \]

\[ S \rightarrow \begin{array}{c} x \end{array} \quad \rightarrow \quad \begin{array}{c} x \\
\text{delegate}(x) \\
\text{register}(y) \end{array} \]

\[ S \rightarrow \begin{array}{c} y \end{array} \quad \rightarrow \quad \begin{array}{c} y \\
\text{equal}(x) \end{array} \]
Merging

\[
\begin{array}{c}
x \\
delegate(x) \\
arrive(x) \\
fem(x)
\end{array}
\quad \quad \quad
\begin{array}{c}
y \\
register(y) \\
y = x
\end{array}
\quad \quad \quad
\begin{array}{c}
x \\
delegate(x) \\
arrive(x) \\
fem(x) \\
register(y) \\
y = x
\end{array}
\]

\[
\begin{array}{c}
x \\
y \\
farmer(x) \\
tractor(y) \\
owns(x,y)
\end{array}
\quad \quad \quad
\begin{array}{c}
\Rightarrow

z \\
u \\
u = y \\
neighbour(z,x) \\
share(x,u,z)
\end{array}
\]
More types of quantification

- **MOST $x$**
  - $x$ $y$
  - farmer($x$)
  - tractor($y$)
  - own($x,y$)

- **neighbour($z,x$)**
  - $v = x$
  - $u = y$

- **MORE $x$**
  - $x_{pl}^x$ $y$ $u$
  - boy($x$)
  - present($y$)
  - $x$ got $y$
  - $u = x$
  - "make-happy"($y,u$)

- **MORE $x$**
  - $x_{pl}^x$ $y$ $u$
  - girl($x$)
  - present($y$)
  - $x$ got $y$
  - $u = x$
  - "make-happy"($y,u$)
\[
\begin{align*}
\forall x & \quad \text{student}(x) \\
\forall x, y & \quad \text{student}(x) \land \text{student}(y) \land x \neq y \\
\forall x & \quad \text{topic}(u) \land \text{chose}(x,u) \\
\forall x & \quad \text{topic}(u) \land \text{chose}(x,u) \land \text{chose}(y,v) \land u \neq v
\end{align*}
\]
John took Mary to Acapulco. They had a lousy time.

\[ U \cup V \]

- John \((u)\)
- Mary \((v)\)
- Acapulco \((y)\)
- \(U\) took \(v\) to \(y\)
- \(Z = u \cup v\)
- \(V = Z\)
- \(U\) had a lousy time.
Sets, classes, and plurals

\[
\begin{align*}
X & \quad y \\
\text{lawyer}^*(X) & \\
|X| & = 5 \\
\text{secretary}(y) & \\
\text{hired}(X, y)
\end{align*}
\]

\[
\begin{align*}
t & \quad s & \quad X & \quad Y \\
\text{Tom}(t) & \\
\text{Sue}(s) & \\
X & = t \, \oplus \, s \\
\text{meet}(t, s) & \\
Y & = X \\
\text{talk}(Y)
\end{align*}
\]

\[
\begin{align*}
x & \\
\text{boy}(x) & \\
\text{at}(x)
\end{align*}
\]

\[
\begin{align*}
z & \quad y \\
\text{book}^*(z) & \\
\text{buy}(x, z) & \\
y & = x \\
\text{person}^*(y) & \\
\text{want}(y, z)
\end{align*}
\]
Summation

Susan found every book which Bill needs. They are on his desk.
Times and events

Events

\[ n = \text{now} \quad (\text{utterance time}) \]
\[ e = \text{event} \]
\[ t = \text{time} \]

Fred drank a 7-up at midnight.

\[ \exists e \exists x \exists y \exists t \left[ \text{drinking}(e, \text{Fred}, x) \& \ 7\text{-up}(x) \& \text{midnight}(t) \& \text{at}(e, t) \right] \]

(for now)
Temporal inclusion and abutment

Mary wrote the letter on Sunday.

\[ \text{Mary}(x) \]
\[ \text{the letter}(y) \]
\[ \text{Sunday}(t) \]
\[ e \leq n \]
\[ e \leq t \]
\[ e : \text{write}(x, y) \]

\[ \leq : \text{temporal inclusion} \]
\[ e \times s : \text{abuts} \]
More times and events

\[
\begin{align*}
n & j \ t_1 \ e_1 \ x \ t_2 \ e_2 \\
\text{Josef}(j) \\
t_1 & \prec n \\
e_1 & \subseteq t_1 \\
e_1: & \text{“turn-around”}(j) \\
t_1 & \prec t_2 \\
\text{“the man”}(x) \\
t_2 & \prec n \\
e_2 & \subseteq t_2 \\
e_2: & \text{“pull-gun”}(x)
\end{align*}
\]

\[
\begin{align*}
n & j \ t_1 \ e_1 \ x \ t_2 \ s_2 \\
\text{Josef}(j) \\
t_1 & \prec n \\
e_1 & \subseteq t_1 \\
e_1: & \text{“turn-around”}(j) \\
t_1 & \subseteq t_2 \\
\text{“the man”}(x) \\
t_2 & \prec n \\
t_2 & \subseteq s_2 \\
s_2: & \text{PROG}(\wedge e_2: \text{“pull-gun”}(x))
\end{align*}
\]
Temporal modification

a. After the exam Fred had a headache.

\[
\begin{align*}
\text{nfs} t_{loc} t_{ex} t_{reg} \\
\text{Fred(f)} \\
\text{“the exam”}(t_{ex}) \\
\text{t}_{loc} \prec n \\
\text{t}_{ex} \supseteq t_{reg} \\
\text{t}_{loc} \subseteq t_{reg} \\
\text{t}_{loc} \subseteq s \\
s: \text{“have-a-headache”}(f)
\end{align*}
\]

b.  

\[
\begin{align*}
\text{nfs} t_{loc} t_{y} \\
\text{Fred(f)} \\
\text{lawn mower(z)} \\
\text{t}_{loc} \prec n \\
\text{DAY}(t_{y}) \\
\text{t}_{y} \supseteq \text{DAY-OF}(n) \\
\text{t}_{loc} \subseteq t_{y} \\
\text{e} \subseteq t_{loc} \\
e: \text{buy(f,z)}
\end{align*}
\]
Time points

- Mary left on January 1st.
  - n=now (utterance time)
  - t=reference time
  - t’=event time
  - x=discourse entity

\[
\begin{array}{cccc}
n & t & t' & e \\
\end{array}
\]

\[
\begin{align*}
e & \subseteq t \\
t & < n \\
t' & < t \\
\text{the 1st of January}(t') \\
\text{Mary}(x) \\
e: & x \text{ leave}
\end{align*}
\]
States

\[ \forall s \forall x \exists y \]

on Sunday \( t \)
\[ s \leq t \]
\[ t \leq n \]
\[ \text{Mary}(x) \]
\[ s = \text{ill}(x) \]

Mary has met the president.

\[ \forall s \forall x \exists y \]
\[ t = n \]
\[ t \leq s \]
\[ \text{Mary}(x) \]
\[ e \in x \subseteq s \]
\[ \text{The president}(y) \]
\[ e : x \text{ meet } y \]
More states and event endpoints

- Since she arrived, Mary has been busy.

\[
\begin{array}{llllllllllll}
 n & t & s & s' & e & t' & t'' & x & z & e' & t'' \\
 t = n & s \circ t & e = \text{beg}(s') & e \supset s & t' \subseteq s' & t'' = \text{beg}(t') & t = \text{end}(t') & & & & \\
 & & & z = x & & & & \text{x be busy} & & & \\
 s': & e \subseteq t'' & t'' < n & e' \subseteq t'' & t'' = \text{loc}(e') & e': & z \text{ arrive} & \\
\end{array}
\]
Connecting utterances

Luigi was writing to the Department Chairman. He had applied for the job without much hope. But the Committee had invited him for a talk, he had given a perfect presentation, they had offered him the job and he had accepted. Now he was worried about what he was going to teach.
A local story
Mary’s visits

- Mary (m)
- Aunt (a, m)
- "Sunday" (t_s)
- on (t_{adv}, t_s)
- t_{loc} ⊆ n
- e ⊆ t_{loc}
- e: “go-to-see” (m, a)

- Mary (m)
- Aunt (a, m)
- t < n
- t = dur (s)
- "last summer" (t_s)
- t_{ls} = dur (s)

- s: "Sunday" (t_s)
  - t_s ⊆ t

- ∀ t_s
  - t_{loc} e t_{adv}
  - on (t_{adv}, t_s)
  - t_{loc} ⊆ t_{adv}
  - e ⊆ t_{loc}
  - e: “go-to-see” (m, a)

- t, t'_{adv}, m
  - Mary (m)
  - Tuesday (t'_{adv})
  - t < n: t = t'_{adv}

- e
  - e ⊆ t
  - e: call (m)
More re Mary

Since last summer Mary has moved to Paris.

\[ n \times s' = t' \times y \]

\[ t = n \]
\[ t < n \]
\[ e \times s \]
\[ e \subseteq t' \]
\[ \text{last summer}(t') \]
\[ t'' = \text{beg}(t') \]
\[ t = \text{end}(t') \]
\[ \text{Mary}(x) \]
\[ \text{Paris}(y) \]
\[ e : x \text{ move to } y \]
Presuppositions and triggers

John made a mistake again.
Link Grammar

• What is a link?
  • Shows a relationship between pairs of words
    • Subject + verb
    • Verb + object
    • Preposition + object
    • Adjective + adverbial modifier
    • Auxiliary + main verb
  • Labels each relationship accordingly
• Potential link types are specified by technical rules
• Possible to score linkages, penalize links
The LG parser

• Freely available for research purposes
• Robust (e.g. information retrieval, MT)
• Calculates simple, explicit relations
• Fast
• Written in C
• Not based on any linguistic theory
• More appropriate for some tasks than traditional phrase-structure parsers
LG example parses

**Linkage 1, cost vector =** (UNUSED=0 DIS=2 AND=0 LEN=23)

```
+-----------------------------------------Xp----------------------------------------+
|                         +-----------------------MVp-----------------------+       |
|                         +---------------MVp--------------+                |       |
|                         |      +-------Jp-------+        +----Js---+      |       |
+--Wd--+Sp*+-PPf-+--Pg*b--+--MVp-+     +----AN----+        |  +---D--+      +-Js+   |
|      |   |     |        |      |     |          |        |  |      |      |   |   |
```

LEFT-WALL I.p 've been.v majoring.v in Material engineering.n at my University in Korea.

**Linkage 1, cost vector =** (UNUSED=0 DIS=2 AND=0 LEN=27)

```
+----------------------------------------------Xp----------------------------------------------+
|      +-----------Wdc-----------+               +------------------Opt-----------------+      |
|      |      +--------CO--------+               |        +--------------AN-------------+      |
|      |      |     +-----D*u----+-------Ss------+        |            +-------AN-------+      |
+--Wc--+      |     +--La-+      +--Mp--+--J-+   |        |            |      +----AN---+      |
|      |      |     |     |      |      |    |   |        |            |      |         |      |
```

LEFT-WALL but probably the best.a class.n for.p me was.v medicine.n and first.n aid.n principles.n.
LG example parses

Thomas Smith, Haverhill, married v at Andover 6 January 1659, Unice Singletary of Salisbury.

LEFT-WALL he was v freeman n 1666.

he was v killed v by the Indians n 15 March 1698.
LG parser’s robustness

Mary married I think, 23 November 1661, Samuel Gay.
No complete linkages found.

Mary married.v [I] [think] [,] 23 November 1661 , Samuel Gay .
Sample link parse

He was killed by the Indians 15 March 1698.

he was.v killed.v by the Indians.n 15 March 1698.
LG-Soar: LG + Soar + DRT

• Parse input via LG parser
• Input words, links into Soar
• Productions to identify and infer:
  • Entities: discourse referents
  • Attributes: properties of entities
  • Actions, states
  • Other relationships: spatial, temporal
  • Anaphor, deixis, other pragmatic content
The system
Associating information

• Individuals: i1, i2

i1
Name: Thomas Smith
Lived: Haverhill
Married: i2
-where? Andover
-when? 6 January 1659
Died: 15 March 1698

i2
Name: Unice Singletary
Lived: Salisbury
Corresponding DRS

\[ u, v, w, x, y, z, a, b, c, d, e \]

\[ \text{Thomas}(u), \text{Smith}(v), \text{Haverhill}(w), \text{v~w}, \]

\[ \text{Andover}(x), \text{Unice}(y), \text{Singleton}(z), \]

\[ \text{Salisbury}(a), \text{v married z, } b=v, \text{freeman}(c), \]

\[ \text{b was c, } d=v, \text{Indians}(e), \text{d was killed} \]

\[ \text{at } x, \text{of } a, \text{propername } yz, \]

\[ \text{month } = \text{January}, \text{day } = 6, \text{year } = 1659, \]

\[ \text{by } e, \text{day } = 15, \text{month } = \text{March}, \text{year } = 1698, \]
Predicate logic equivalent

- Thomas Smith, Haverhill, married at Andover 6 January 1659, Unice Singletary of Salisbury.

\text{Unice}(y), \text{Singletary}(z), \text{prep}(\text{"at"}, x), \text{verbal}(\text{"married"}, v, x) \\
\text{propername}=uv \\
\text{modifier}=\text{"Haverhill"} \\
\text{Thomas}(u), \text{Smith}(v), \text{Haverhill}(w), \text{Andover}(x), 6(m), \text{January}(n), \\
\text{propername}=yz \\
\text{time}(\text{day } m, \text{month } n), 1659(o), \text{time}(\text{month } n, \text{year } o), \text{Salisbury}(a), \\
\text{modifier}=\text{"January"} \\
\text{modifier}=\text{"Andover"}
The output

• Predicate-argument relationships
• Discourse representation structures
  • CLIG grapher output
• GEDCOM files
LOGICAL FORM IDENTIFICATION FOR MEDICAL CLINICAL TRIALS
Approach

- Identify and extract predicate logic forms from medical clinical trials (in)eligibility criteria
  - Clint Tustison, Soar 22
- Match up the information with other data, i.e., patients’ medical records
  - Craig Parker, MD & medical informatics at IHC
- Tool for helping match subjects with trials
- Use of UMLS, the NIH’s vast unified medical terminological resource
Process

Clinical Trials
(www input)

Text processing

LG syntactic parser

Soar engine

Predicate Calculus

Post-Processing
(output)
Input

- ClinicalTrials.gov
- Sponsored by NIH and other federal agencies, private industry
- 8,800 current trials online
- 3,000,000 page views per month
- Purpose, eligibility, location, more info.
Process: Input

Clinical Trials (www input)

A criterion equals adenocarcinoma of the pancreas.

Syntactic Parser

Soar engine

Predicate Calculus

Post-Processing (output)
A criterion equals adenocarcinoma of the pancreas.

LEFT-WALL a criterion.n equals.v adenocarcinoma[?].n of the pancreas.n .
Shallow semantic processing

- Soar (not NL-Soar) backend
- Translates syntactic parse to logic output by reading links \(\rightarrow\) shallow semantics
- Identify concepts, create predicates, determine predicate arity, instantiate variables, perform anaphoric and coreference processing
- Predicate logic expressions
Process: Logic Output

Clinical Trials (www input)

A criterion equals adenocarcinoma of the pancreas.

Syntactic Parser

Soar engine

Predicate Calculus

criterion(N2) & adenocarcinoma(N4) & pancreas(N5) & equals(N2,N4) & of(N4,N5).

Post-Processing (output)
Post-processing

- Prolog axioms
- Remove subject/verb
- Eliminate redundancies
- Filter irrelevant information

\[ c\text{riterion}(N2) \land adenocarcinoma(N4) \land pancreas(N5) \land equals(N2,N4) \land of(N4,N5). \]

\[ adenocarcinoma(N4) \land pancreas(N5) \land of(N4,N5). \]
XML output for downstream

<criteria trial="http://www.clinicaltrials.gov/ct/show/NCT00055250">
  <criterion>
    <text>Eligibility</text>
    <text>Criteria</text>
    <text>Inclusion Criteria:</text>
    <text val="1">Adenocarcinoma of the pancreas</text>
    <pred val="1">pancreas(N5) & adenocarcinoma(N4) & of(N4,N5).</pred>
  </criterion>
</criteria>
Sample LG applications

- Grading EFL essays
- Extracting biographic facts from genealogical documents
- Analyzing newspaper headlines
- Clinical trial records and patient data matching
- Other languages (Farsi, Arabic, Lushootseed)
- BOLT-E cognitive robotics
- FamilySearch OntoSoar
News headlines extraction

grenade attack(x) & U.S. soldier(y) & Iraq(z) & in(y,z) & kills(x,y).

wall street analysts(x) & stock prices(y) & inflate(x,y) & routinely(inflate).
Machine reading

• Aka “deep reading” and “reading the Web”
• YAGO-NAGA
• KnowItAll
• NELL
Example 1

243314. Charles Christopher Lathrop, N. Y. City, b. 1817, d. 1865, son of Mary Ely and Gerard Lathrop; m. 1856 Mary Augusta Andruss, 992 Broad St., Newark, N. J., who was b. 1825, dau. of Judge Caleb Halstead Andruss and Emma Sutherland Goble. Mrs. Lathrop died at her home, 992 Broad St., Newark, N. J., Friday morning, Nov. 4, 1898. The funeral services were held at her residence on Monday, Nov. 7, 1898, at half-past two o’clock P. M. Their children:

2. William Gerard, b. 1858, d. 1861.
3. Theodore Andruss, b. 1860.
4. Emma Goble, b. 1862.

Miss Emma Goble Lathrop, official historian of the New York Chapter of the Daughters of the American Revolution, is one of the youngest members to hold office, but one whose intelligence and capability qualify her for such distinction.
Children of James Harwood, No. 103.

229. Myra, born July 20, 1839 in Eden, Vt. She married Elijah Spencer, Dec. 25, 1851. They had five children: Arvilla, born in 1852, is not living; Mariette, born Dec. 25, 1854, married Jonathan Snyder, have a family; Leverett, born Feb. 6, 1857, married Cora Smith, Nov. 2, 1879, had two children, Perry F. and Ida I. Leverett died May 21, 1910; Rosa E., born Jan. 13, 1860, married Emmett Byers, and have children; and Harrison, born about 1862, is not living. Elijah Spencer died in the Union army in 1863, and his widow married Jonathan Squires, who was born in Ohio, July 25, 1823, by whom she had one son, J. Wilbur, born June 16, 1865, in DeKalb county, Ind., married Cora M. Thomas, Aug. 24, 1887, they reside in St. Joseph, Mich., five children. Mrs. Myra Squires died in Allen county, Ind., Feb. 13, 1874.
Results on *The Ely Ancestry*

a book of 830 pages, including our Example 1

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Instance Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persons</td>
<td>16,848</td>
</tr>
<tr>
<td>Births</td>
<td>8,609</td>
</tr>
<tr>
<td>Deaths</td>
<td>2,406</td>
</tr>
<tr>
<td>Genders</td>
<td>1,674</td>
</tr>
<tr>
<td>Couples</td>
<td>3,343</td>
</tr>
<tr>
<td>Children</td>
<td>3,049</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>35,929</strong></td>
</tr>
</tbody>
</table>
Coal

- Not as useful when linguistics required
- No cognitive modeling: Soar is just a programming environment
- Linguistic grammar development very opaque

Nuggets

- Robust parsing
- Flexible: various applications
- Wide range of applications
- Various semantic outputs
- Interest growing
MS spam filtering patent application
Machine reading: FRED
Dialogue Move Engines

- Computer systems (usually agents) that explicitly manage dialogues
  - Participants
  - Common ground
  - Plans
  - Beliefs
  - Agenda
State-of-the-art in Dialogue Management

• Finite State Machine
  • Hard-coded solution; domain specific

• Belief-desire-intention architectures
  • Better representation of dialogue flow
  • Conversational record and Dialogue Move Engine

• Cognitive Modeling system (NL-Soar)
  • Discourse Recipes
  • Learning
  • Syntactic and semantic parses
Godis

Dialogue grammar (resource interface)
Database (resource interface)
Plan library (resource interface)

Latest speaker input
Program state
Next moves

Dialogue Move Engine (DME)
update
select

Control

Input
Interpretation
Generation
Output

TIS
Information State (IS)

Dialogue grammar (resource interface)
Database (resource interface)
Plan library (resource interface)
Cognitive robotics