Learning by Reading: Toward 'Shallow' Semantics-Based NLP

Eduard Hovy

USC Information Sciences Institute

USC/ISI: Jerry Hobbs, Chin-Yew Lin, Hans Chalupsky, Patrick Pantel, Andrew Philpot, Rutu Mulkar, Marco Pennacchiotti

University of Texas at Austin: Bruce Porter, Ken Barker, students

BBN: Ralph Weischedel, Paul Martin

SRI: David Israel, Jenn Baker

Consultant and overall project director: Noah Friedland

USC INFORMATION SCIENCES INSTITUTE

The old dream of AI…

…can we build a machine that learns by reading?

NLP and KR&R separated in the 1960s

Current relevant work in NLP:

- Information extraction of events (IE):
  - FRUMP (1977), MUC, ACE, etc.…since 1980s
- Term/concept/relation induction and ontology construction:
  - Pantel, Snow & Jurafsky, Kozareva et al., …
- Instance harvesting:
  - Hearst, Ravichandran & Hovy, KnowItAll (Etzioni), WebFountain (IBM)…

The Big Problems!

- *"Salt (Na^+Cl^-) is a white powder with a salty taste. As you can see, it is an ionic compound. You will see the powder dissolve when you put it into water.*"
  - Does the formula Na^+Cl^- have a salty taste?
  - Is the powder the formula? Can you write a powder?
  - Does the taste dissolve? Or the whiteness?

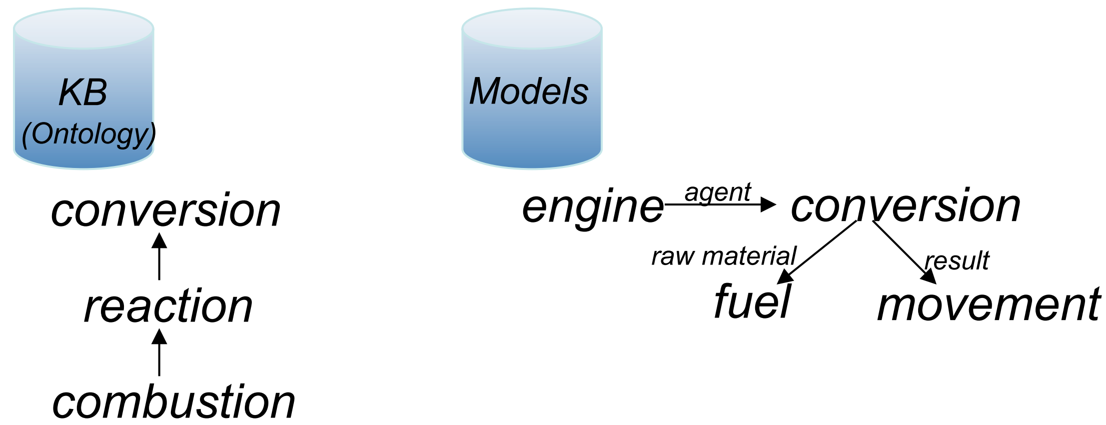
- A lot of information is hidden, and a lot assumed:
  - Knowledge gaps : explicit links between one term and another
  - Omissions : missing (assumed known?) information

Language is full of what Peter Clark calls '*loosespeak*'

USC INFORMATION SCIENCES INSTITUTE

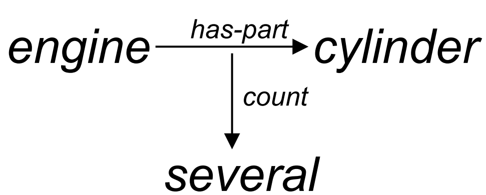
Example: Incrementally adding knowledge into a growing worldview

Given knowledge



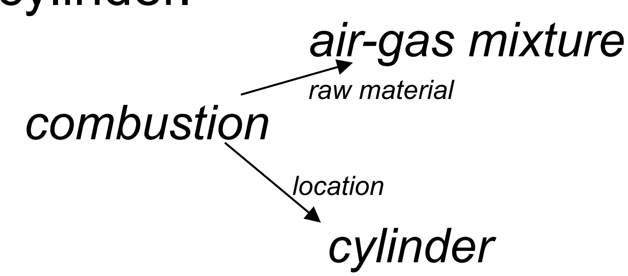
Text1

"An engine has several cylinders."



Text2

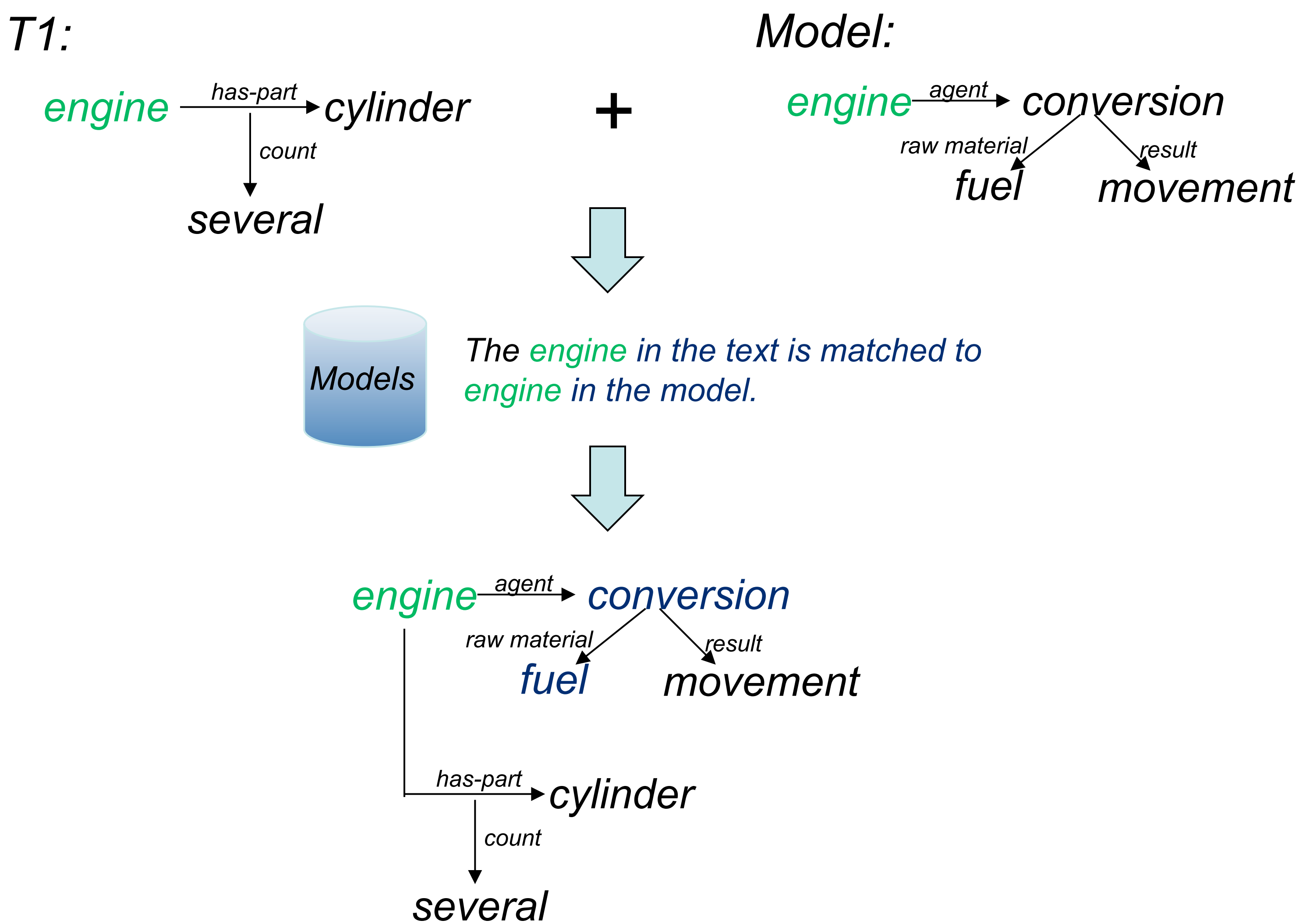
"The air-gas mixture combusts in the cylinder."



USC INFORMATION SCIENCES INSTITUTE

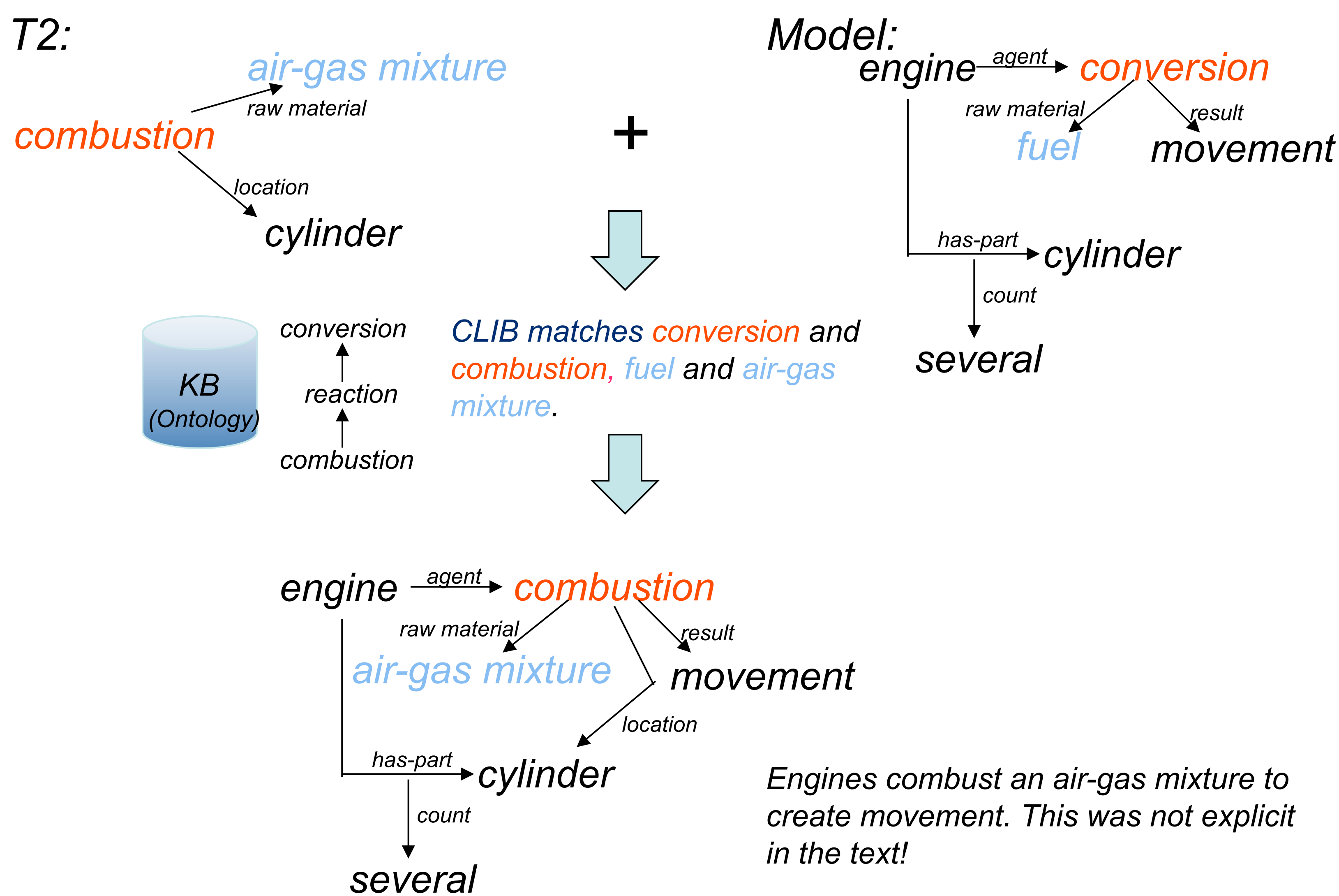
Slide by Noah Friedland

Example: Knowledge integration, text 1



Slide by Noah Friedland

Example: Knowledge integration, text 2



Engines combust an air-gas mixture to create movement. This was not explicit in the text!

Slide by Noah Friedland

USC INFORMATION SCIENCES INSTITUTE

But what about gaps?

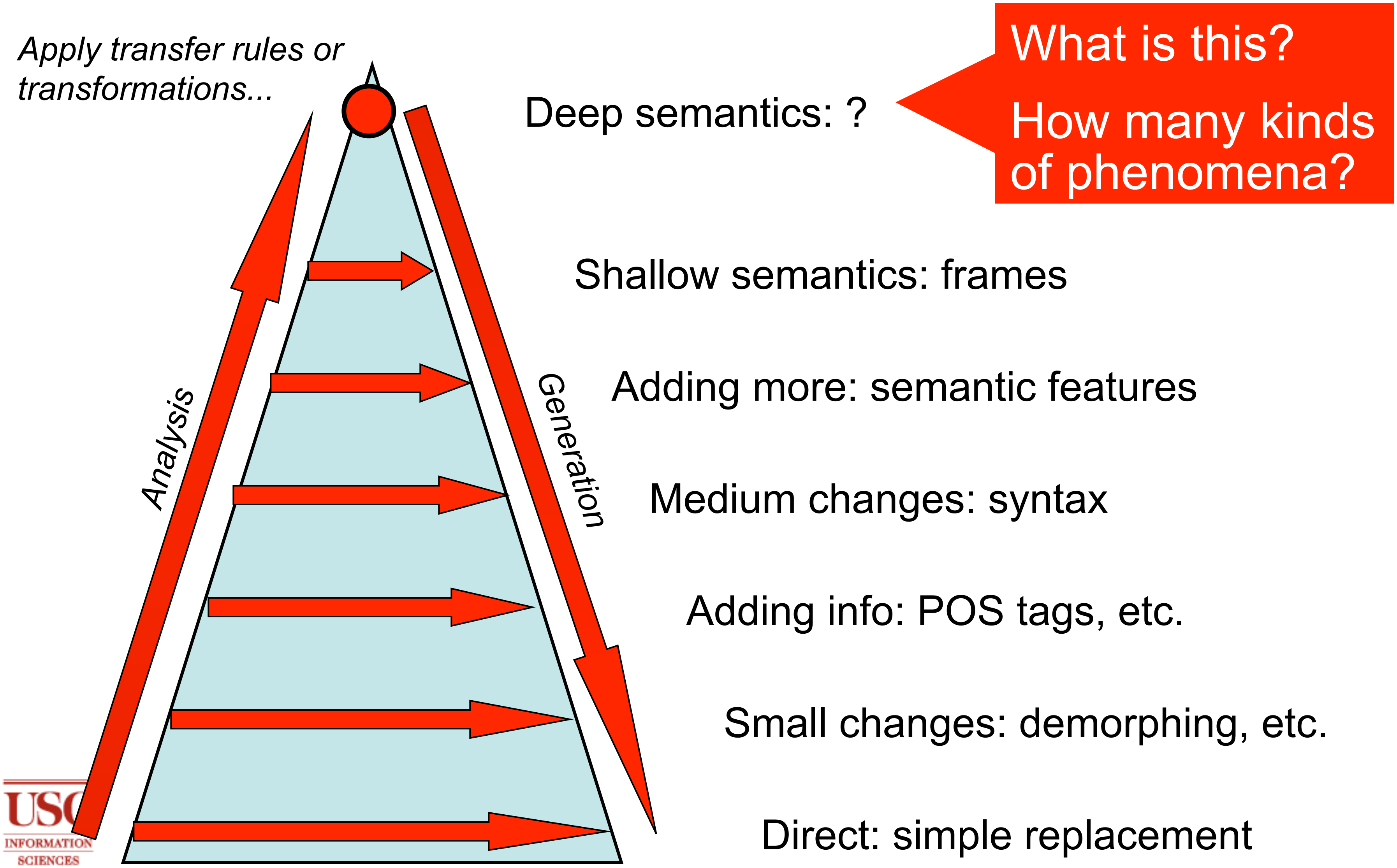
- How do you find the knowledge to connect together independent pieces?
- What kinds of background knowledge and assumptions are ‘allowed’?
- How do you take new information from text and turn it into the appropriate ‘gap-filling’ rules?

USC
INFORMATION
SCIENCES
INSTITUTE

Our approach

- Reading:
  - Need common and standardized representation (formalism, terms, and relation names)
  - Need some basic background starting knowledge / models
  - Must convert NL sentences into formalism expressions (handle wordsense, coref, tense, modality, etc.)
  - Use abductive inference to close gaps (connect representations)
  - Must integrate new information:
    - Either ensure global consistency,
    - Or handle alternative possible interpretations
- Testing: QA
  - Read question, convert to rep, match rep to knowledge, provide answer

NLP at increasing depths



USC INFORMATION SCIENCES INSTITUTE

Toward a Global Language: Some aspects of semantics

Somewhat easier

Bracketing (scope) of predications
Word sense selection (incl. copula)
NP structure: genitives, modifiers…
Concepts: ontology definition
Concept structure (incl. frames and roles)
Coreference (entities and events)
Pronoun classification (ref, bound, event, generic, other)
Identification of events
Temporal relations (incl. discourse, aspect)
Manner relations
Spatial relations
Direct quotation and reported speech

Perhaps more difficult

Quantifiers and numerical expressions
Comparatives
Coordination
Information structure (theme/rheme)
Focus
Discourse structure
Other adverbials (modals, evidentials, etc.)
Identification of propositions (modality)
Opinions and subjectivity
Pragmatics/speech acts
Polarity/negation
Presuppositions
Metaphors

First, the BIG PROBLEM: Language is incomplete at the surface level…so how can you create *enough*, *rich*, and *deep* semantic 'background' knowledge?

USC INFORMATION SCIENCES INSTITUTE

The knowledge bottleneck problem

Creating rich and deep enough semantic knowledge…

- By human knowledge definition/entry?
  - AI: Conceptual Dependency (Schank & Abelson 1970s), etc.
  - Ontologies: CYC (Lenat 1990s–), etc.
  - Instance mining from the web: (IBM's WebFountain 2005–), etc.

  …but there's too much knowledge, and human knowledge entry is not consistent!

- By machine?…Learning by Reading:
  - Provide small amount of startup knowledge: 'seed'
  - Then let computer read and bootstrap its own knowledge
  - Original goal of AI — 1950s

LbR: Can we reconnect NLP and KR?

- Why are we silly enough to think we can do it now?
  - Robust parsers like MINIPAR, Charniak, Collins…
  - Large shallow semantic resources like WordNet
  - Progress on knowledge rep and reasoning (KR&R) systems
  - Success of project HALO in 2004 (Friedland et al.)
  - DARPA's LbR projects (2005) and Project Möbius (2006–): this talk

- Problems:
  - Seeds: what to start with? Why?
  - Data: what to read? In what order?
  - Reps: what to represent? Why?
  - Inference: how to learn axioms?
  - Evaluation: how to measure LbR?
  - Applications: how to use the results (MT?)

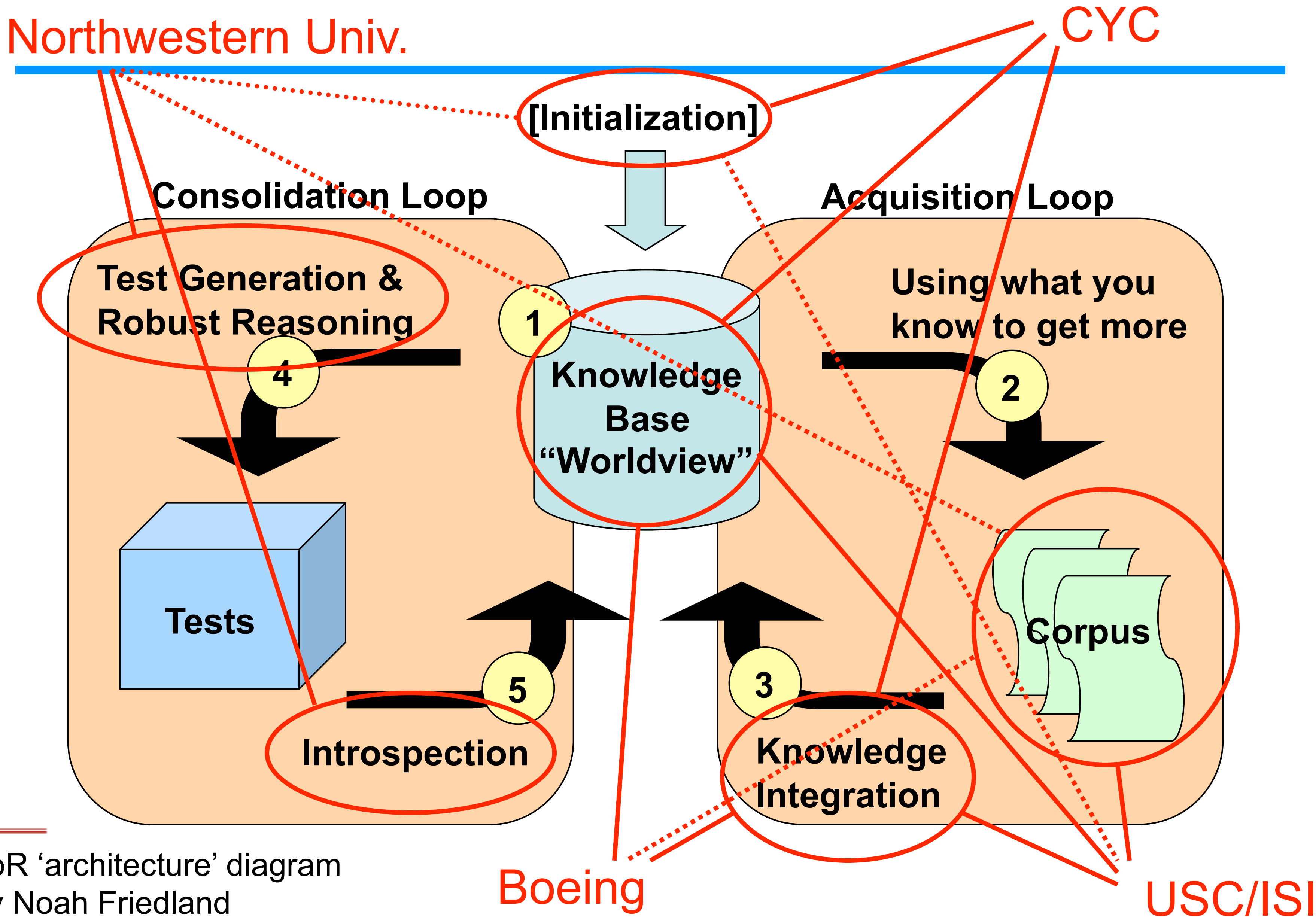
Clearly, we are sure to fail — the question is, *how* will we fail? What will we learn?

USC INFORMATION SCIENCES INSTITUTE

Talk overview

1. Introduction: The dream
2. Background: DARPA LbR seedlings in 2005
3. The Möbius experiment 2006–07
   - Partners, architecture, and domain
   - NL interpretation: Parsing, Logical Form, Abduction
   - Deep inference and shallow broad coverage
4. Tests and evaluations
5. What did we learn? The Future

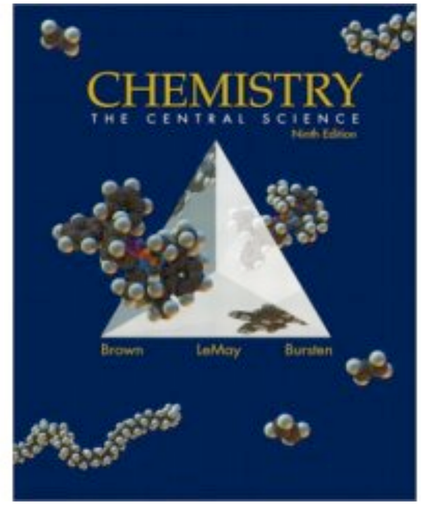
2005 DARPA's LbR seedling: Framework

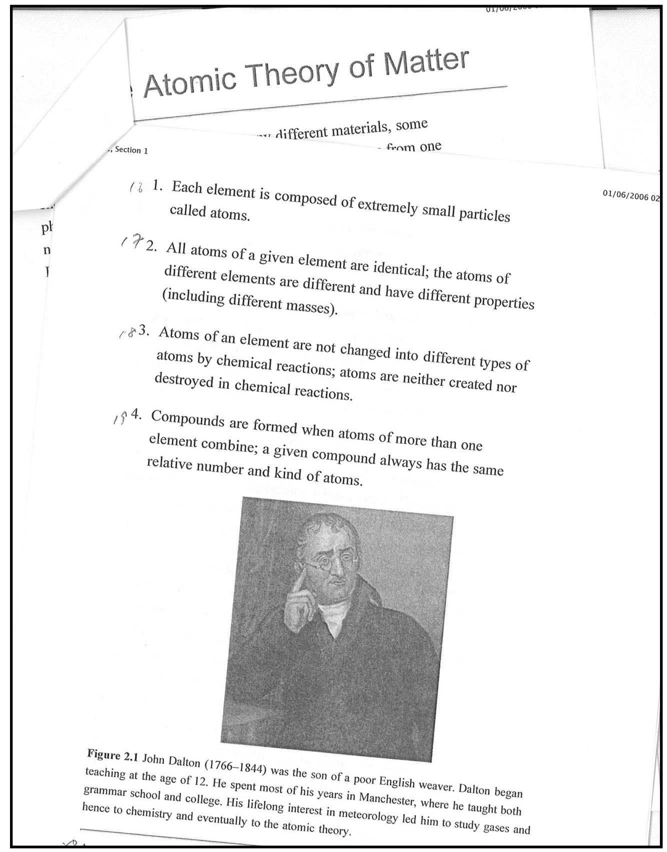


LbR 'architecture' diagram
by Noah Friedland

Domain: Source text

- High School Chemistry textbook
  - *Chemistry: The Central Science* (9th ed). Brown, LeMay, Bursten, Burdge
  - 313590 word tokens; 12722 diff words
- Sample text, processed:
  - <S SNTNO="13">As chemists learned to measure the amounts of materials that reacted with one another to make new substances , the ground was laid for a chemical atomic theory .</S>
  - <S SNTNO="14">That theory came into being during the period 1803 – 1807 in the work of an English schoolteacher , John Dalton (Figure 2.1) .</S>
  - <S SNTNO="15">Reasoning from a large number of observations , Dalton made the following postulates :</S>
  - <S SNTNO="16">Each element is composed of extremely small particles called atoms .</S>
  - <S SNTNO="17">All atoms of a given element are identical ; the atoms of different elements are different and have different properties (including different masses) .</S>
  - <S SNTNO="18">Atoms of an element are not changed into different types of atoms by chemical reactions ; atoms are neither created nor destroyed in chemical reactions .</S>
  - a given compound always has the same relative number and kind of atoms .</S>



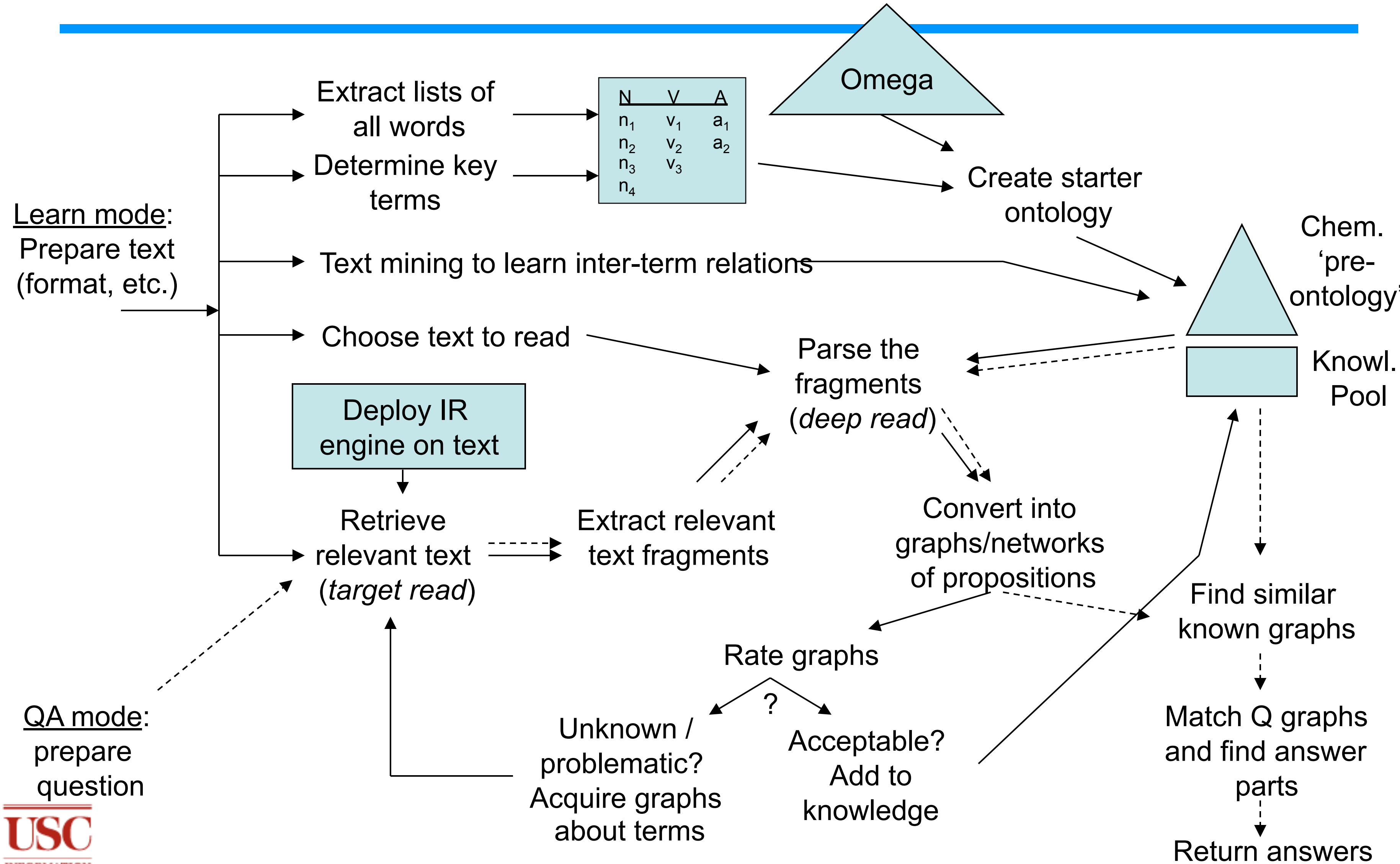

Atomic Theory of Matter

1. Each element is composed of extremely small particles called atoms.
2. All atoms of a given element are identical; the atoms of different elements are different and have different properties (including different masses).
3. Atoms of an element are not changed into different types of atoms by chemical reactions; atoms are neither created nor destroyed in chemical reactions.
4. Compounds are formed when atoms of more than one element combine; a given compound always has the same relative number and kind of atoms.

Figure 2.1 John Dalton (1766–1844) was the son of a poor English weaver. Dalton began teaching at the age of 12. He spent most of his years in Manchester, where he taught both grammar school and college. His lifelong interest in meteorology led him to study gases and hence to chemistry and eventually to the atomic theory.

USC INFORMATION SCIENCES INSTITUTE

ISI system development stages



ISI: Using the learned knowledge in the PowerLoom reasoning system

First load core term definitions

Then assert the semantic reading created for a new sentence

```
STELLA(38): (demo "~/Projects/learning-by-reading/queries.plm")
Now reading from `~/Projects/learning-by-reading/queries.plm'.
Type `?' at the pause prompt for a list of available commands.
;;; -*- Mode: Lisp; Package: STELLA; Syntax: COMMON-LISP; Base: 10 -*-

;; Each element is composed of extremely small particles called atoms.

(ASSERT
 (FORALL (?E34 ?E35 ?x)
  (=> (AND
       (subject ?E34 ?E35)
       (element' ?E35 ?x))
      (EXISTS (?E62 ?E63 ?s1 ?e10 ?y ?e4 ?e9 ?e3 ?e5 ?a ?z ?e11 ?s2 ?
      e6)
       (AND
        (asserted ?E62 ?E63)
        (compose' ?E63 ?x ?s1)
        (plural ?e10 ?y ?s1)
        (small' ?e4 ?y)
        (extremely ?e9 ?e4)
        (particle' ?e3 ?y)
        (call' ?e5 ?a ?y ?z)
        (plural ?e11 ?z ?s2)
        (atom' ?e6 ?z))))))
```

USC INFORMATION SCIENCES INSTITUTE

A simple Y/N question

```
;; Is each element composed of extremely small particles called atoms?
;;     (literal copy of sentence 1):
|= (ASK
  (FORALL (?E34 ?E35 ?x)
    (=> (AND
          (subject ?E34 ?E35)
          (element' ?E35 ?x))
         (EXISTS (?E62 ?E63 ?s1 ?e10 ?y ?e4 ?e9 ?e3 ?e5 ?a ?z ?e11 ?s2 ?e6)
           (AND
            (asserted ?E62 ?E63)
            (compose' ?E63 ?x ?s1)
            (plural ?e10 ?y ?s1)
            (small' ?e4 ?y)
            (extremely ?e9 ?e4)
            (particle' ?e3 ?y)
            (call' ?e5 ?a ?y ?z)
            (plural ?e11 ?z ?s2)
            (atom' ?e6 ?z))))))
------ pause ------

TRUE
```

Q

A

USC INFORMATION SCIENCES INSTITUTE

Variants handled by reasoner

```
;; Is each element composed of extremely small
       particles?
;;    (sentence 1 but with fewer restrictions):

|= (ASK
  (FORALL (?E34 ?E35 ?x)
    (=> (AND
         (subject ?E34 ?E35)
         (element' ?E35 ?x))
        (EXISTS (?E62 ?E63 ?s1 ?e10 ?y ?e4 ?e9 ?e3)
          (AND
           (asserted ?E62 ?E63)
           (compose' ?E63 ?x ?s1)
           (plural ?e10 ?y ?s1)
           (small' ?e4 ?y)
           (extremely ?e9 ?e4)
           (particle' ?e3 ?y))))))

------ pause ------

TRUE
```

```
;; Is each element composed of particles?
;;    (and even fewer restrictions):

|= (ASK
  (FORALL (?E34 ?E35 ?x)
    (=> (AND
         (subject ?E34 ?E35)
         (element' ?E35 ?x))
        (EXISTS (?E62 ?E63 ?s1 ?e10 ?y ?e4 ?e9 ?e3)
          (AND
           (asserted ?E62 ?E63)
           (compose' ?E63 ?x ?s1)
           (plural ?e10 ?y ?s1)


           (particle' ?e3 ?y))))))

------ pause ------

TRUE
```

BUT: Need knowledge of English

```
;; Is each element composed of atoms?

|= (ASK
  (FORALL (?E34 ?E35 ?x)
    (=> (AND
         (subject ?E34 ?E35)
         (element' ?E35 ?x))
        (EXISTS (?E62 ?E63 ?s1 ?e10 ?y ?e4 ?e9 ?e3)
          (AND
           (asserted ?E62 ?E63)
           (compose' ?E63 ?x ?s1)
           (plural ?e10 ?y ?s1)
           (atom' ?e3 ?y))))))
------ pause ------

UNKNOWN
```

Need an axiom about the meaning of "call" (transfer of properties, as in "*X's called Y's*"):

```
(DEFRULE R1
  (=> (AND (HOLDS ?r1 ?e3 ?y)      ;; (=> (AND (particle' ?e3 ?y)
           (call' ?e5 ?a ?y ?z)    ;;          (call' ?e5 ?a ?y ?z)
           (HOLDS ?r2 ?e6 ?z))     ;;          (atom' ?e6 ?z))
      (HOLDS ?r2 ?e3 ?y))          ;;      (atom' ?e3 ?y)))
  :forward-only? TRUE)
------ pause ------
```

```
;; |P|(FORALL (?r1 ?e3 ?y ?e5 ?a ?z ?r2 ?e6)
   (=> (AND (HOLDS ?r1 ?e3 ?y)
            (call' ?e5 ?a ?y ?z)
            (HOLDS ?r2 ?e6 ?z))
       (HOLDS ?r2 ?e3 ?y)))
```

Now the query works:

```
;; Is each element composed of atoms?

|= (ASK
  (FORALL (?E34 ?E35 ?x)
    (=> (AND
         (subject ?E34 ?E35)
         (element' ?E35 ?x))
        (EXISTS (?E62 ?E63 ?s1 ?e10 ?y ?e4 ?e9 ?
        e3)
          (AND
           (asserted ?E62 ?E63)
           (compose' ?E63 ?x ?s1)
           (plural ?e10 ?y ?s1)
           (atom' ?e3 ?y))))))
------ pause ------

TRUE
```

The lesson

- To ‘stitch together’ the incoming logical propositions obtained from a sentence, you need a lot of background knowledge about the basic meanings of English words
- This knowledge must be defined using a core set of terms that fit the system’s starting models

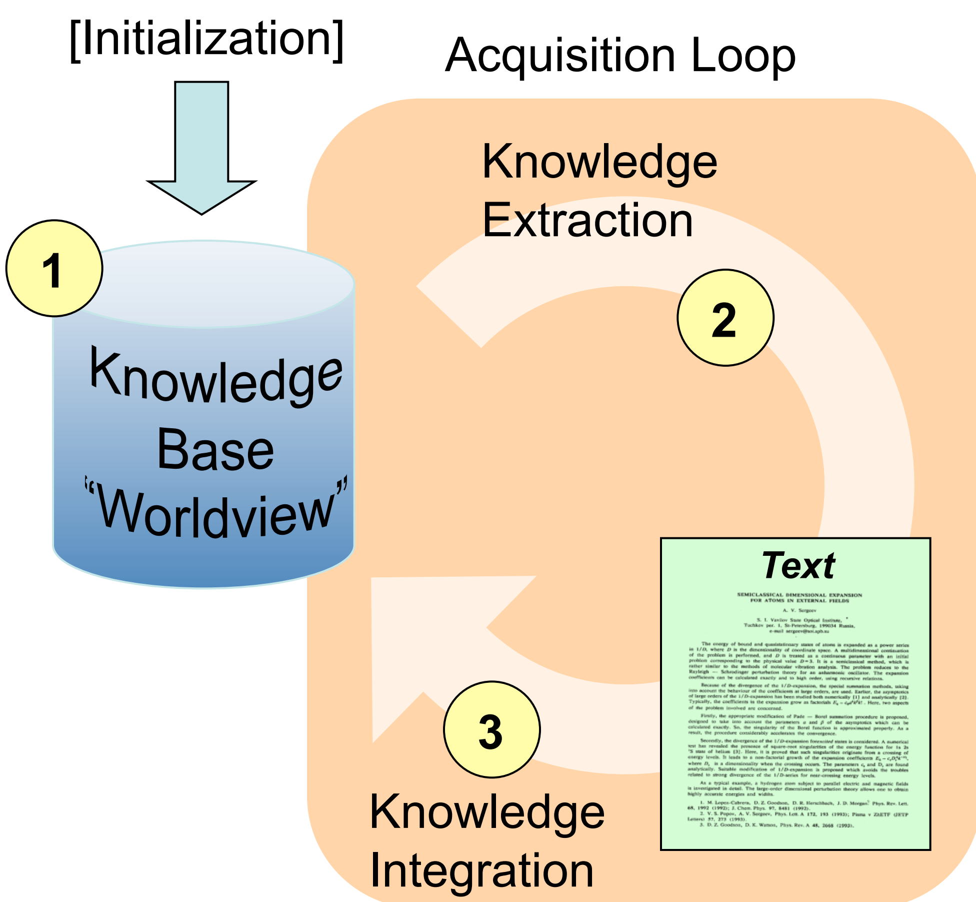
So let’s investigate how a machine can learn a model of something concrete, and build up is knowledge of its parts and functioning…

Talk overview

1. Introduction: The dream
2. Background: DARPA LbR seedlings in 2005
3. The Möbius experiment 2006–07
   – Partners, architecture, and domain
   – NL interpretation: Parsing, Logical Form, Abduction
   – Deep inference and shallow broad coverage
4. Tests and evaluations
5. What did we learn? The Future

USC INFORMATION SCIENCES INSTITUTE

Möbius architecture 2006–07



1 **KM Knowl Base with 1000 concepts (Texas)**

2 **Text parsing/interp and knowledge extraction (ISI and BBN)**

3 **Knowledge refinement, integration and hypothesis generation (Texas and ISI)**

Framework, data, and interfaces (SRI)

USC INFORMATION SCIENCES INSTITUTE

Slide by Noah Friedland

Möbius domains 1 and 2

Domain selection criteria:

- College student level text, not too mathematical (math or physics), not purely descriptive (some parts of biology or anatomy), and not argument-based or rhetorical
- Containing descriptions of both form and function
- Easy to find texts online
- (Potential) military relevance (for DARPA)

- 2006 domain: **the (human) heart**
  - Typical text:

    The heart is a muscular pump. It is responsible for distributing blood throughout the body. The heart is a little larger than a fist. It is located behind and protected by the ribs. The heart is divided into four chambers. The top two chambers are called atria, while the bottom two chambers are called ventricles. The septum is a wall of muscle that divides the left and right sides of the heart. The heart is nourished by oxygenated blood. Large arteries connect the heart to the body and the lungs, delivering de-oxygenated blood from the body into the heart.

- 2007 domain: **engines** (steam, gas, turbine)

USC INFORMATION SCIENCES INSTITUTE

Historical summary

Human Heart – focused on providing **qualitative** evidence for the feasibility of LbR

Engines – focused on **quantitatively** establishing this feasibility in a much broader domain

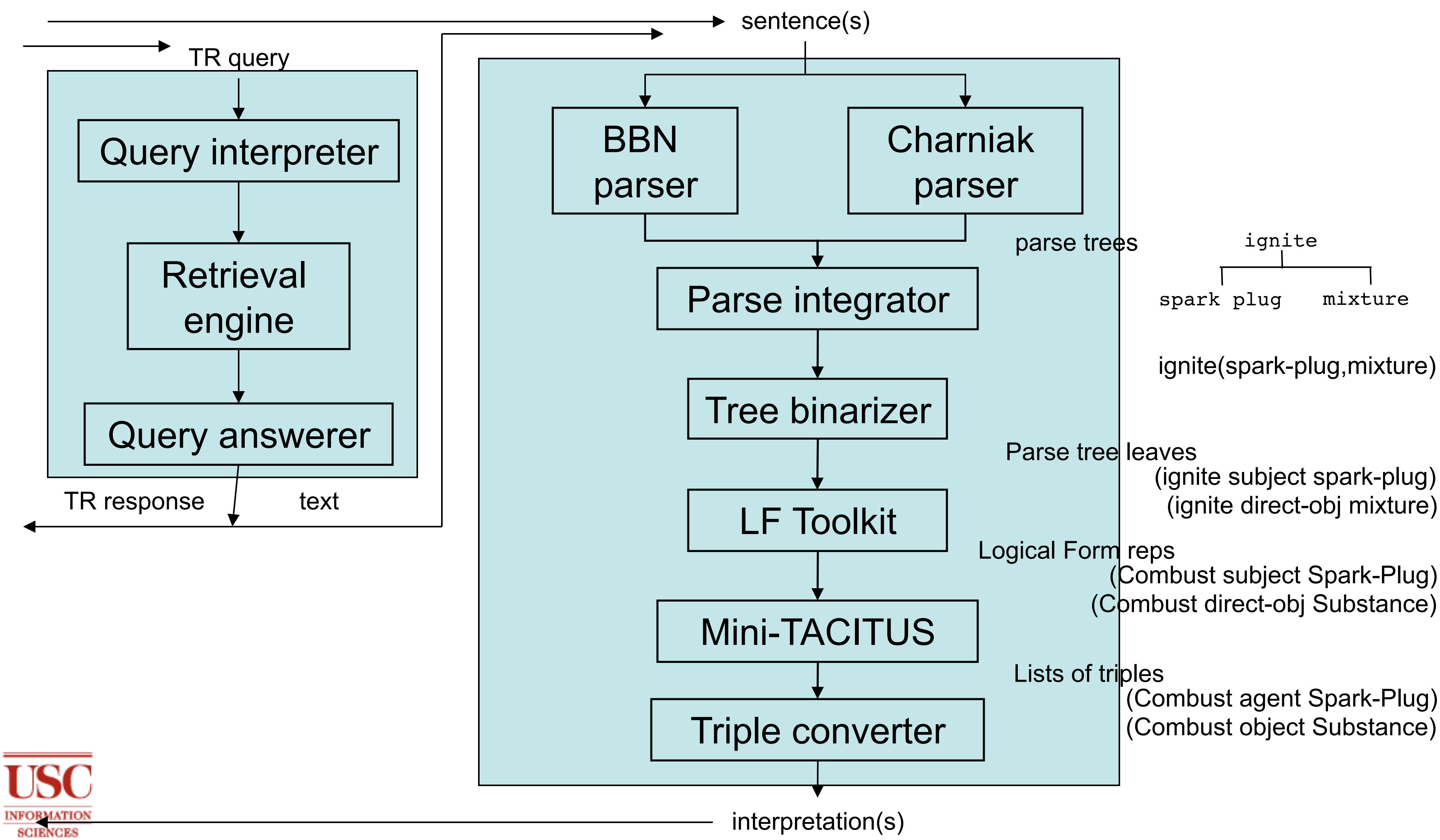
Three major Learning-by-Reading (LbR) research challenges identified:

1. Bridging the NL-KR gap – harvesting logical forms from naturally occurring text
2. Synthesizing the Knowledge Model– incrementally and automatically forming robust models from text
3. Doing problem solving – developing problem solving techniques to allow the utilization and maintenance of text-derived models

Talk overview

1. Introduction: The dream
2. Background: DARPA LbR seedlings in 2005
3. The Möbius experiment 2006–07
   - Partners, architecture, and domain
   - NL interpretation: Parsing, Logical Form, Abduction
   - Deep inference and shallow broad coverage
4. Tests and evaluations
5. What did we learn? The Future

NL pipeline


sentence(s)
TR query
Query interpreter
Retrieval engine
Query answerer
TR response
text
BBN parser
Charniak parser
parse trees
ignite
spark plug
mixture
Parse integrator
ignite(spark-plug,mixture)
Tree binarizer
Parse tree leaves
(ignite subject spark-plug)
(ignite direct-obj mixture)
LF Toolkit
Logical Form reps
(Combust subject Spark-Plug)
(Combust direct-obj Substance)
Mini-TACITUS
Lists of triples
(Combust agent Spark-Plug)
(Combust object Substance)
Triple converter
interpretation(s)

NL triple formation

"The heart is a pump that works together with the lungs"

Parser output

Example sentence (simplified):

[1] The heart is a pump that works together with the lungs [S-SNT]
    (SUBJ) [2] <The heart>1 [S-NP]
        (DET) [3] The [S-DEF-ART]
        (PRED) [4] heart [S-COUNT-NOUN]
    (PRED) [5] is [S-AUX]
    (COMPL) [6] a pump that works together with the lungs [S-REL-CLAUSE]
        (MOD) [7] a pump [S-NP]
            (DET) [8] a [S-INDEF-ART]
            (PRED) [9] pump [S-NOUN]
        (SUBJ) [10] that [S-INTERR-NP]
            (PRED) [11] that [S-INTERR-PRON]
        (PRED) [12] works [S-INTR-VERB]
        (DIR) [13] together with the lungs [S-PP]
            (P) [14] together with [S-PREP]
                (LEXICAL-1) [15] together [S-ADV]
                (LEXICAL-2) [16] with [S-PREP]
            (PRED) [17] the lungs [S-NP]
                (DET) [18] the [S-DEF-ART]
                (PRED) [19] lungs [S-COUNT-NOUN]

Simplified Logical Form

is(e0,x0,x1)
 heart-nn(x0)
 pump-nn(x1)
work-vb(e1)
 lung-nn(x3)
 together_with(e2,e1,x3)
 agent_of2(x3,e1)
 agent_of1(x1,e1)

NL Triples

[Interpt Number: 20
Cost: 56
e0-is
  eventuality-of is
x0-heart
  is x1-pump
  instance-of heart
x1-pump
  agent-of e1-work
  instance-of pump
e1-work
  instance-of work
  together-with x3-lung
…]

General approach, theory, and engines designed by Jerry Hobbs

Step 1: Parsing

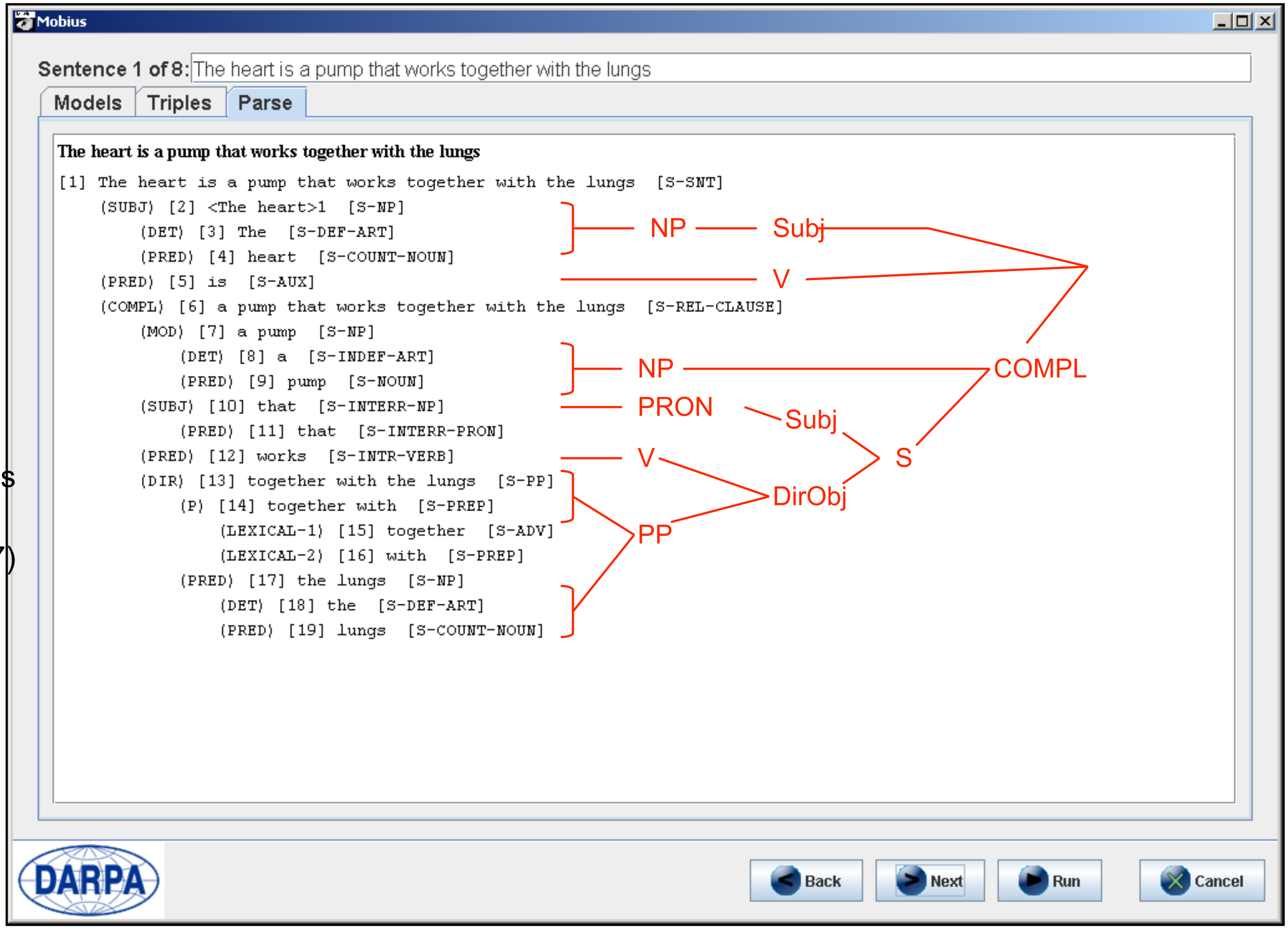
LbR 05: tested Charniak and Hermjakob parsers

Möbius06: Hermjakob

- Deterministic shift-reduce
- Trainable
- Syntax and semantic labels
- Tested on MT (Hermjakob 97) and QA (TextMap 03)

Möbius07: Charniak

BBN Serif



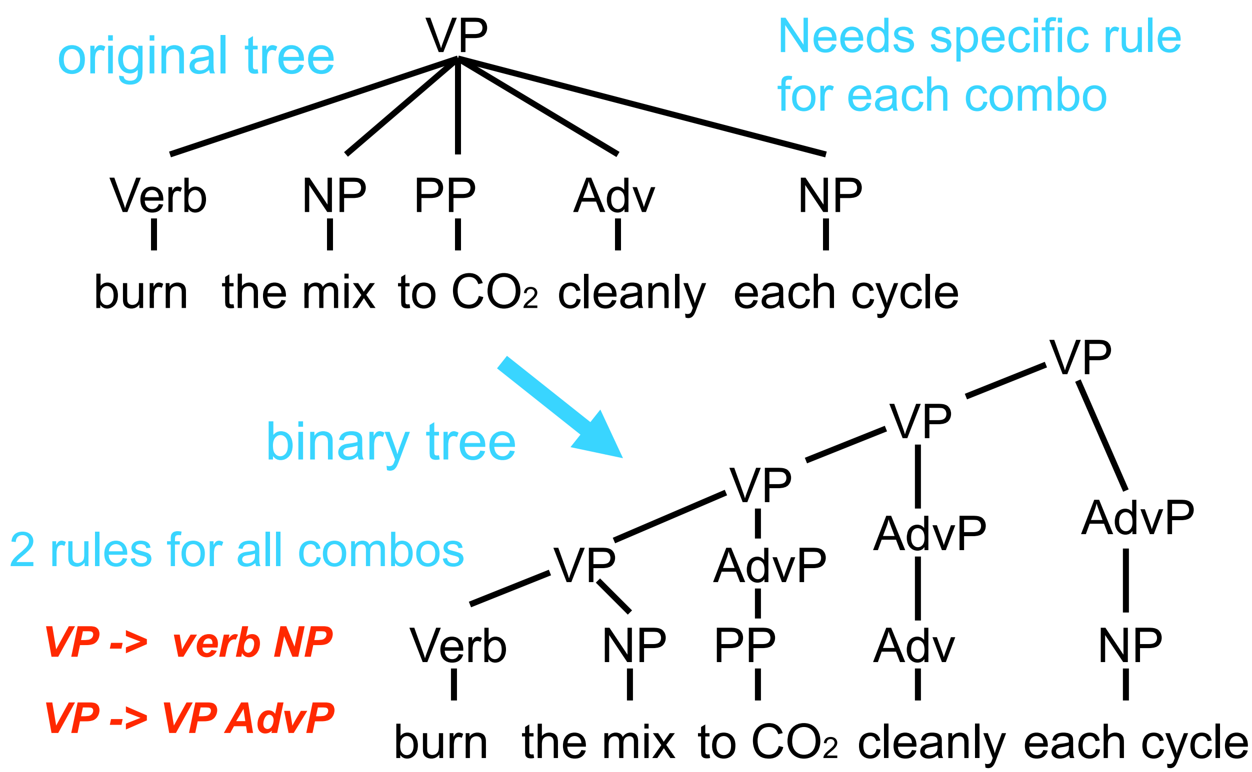
Step 2: Post-parse tree binarization

Statistical parsers provide too much variation in parse trees for easy conversion into Logical Forms (over 10,000 possible forms at each node)

Parse tree binarizer converts parse tree into simpler binary format, percolating lexical and context info

Small set of binarization (< 500) and extraction (< 300) rules maps parse tree nodes into Logical Form fragments

Many more trees match the rules that establish logical connections among output triples



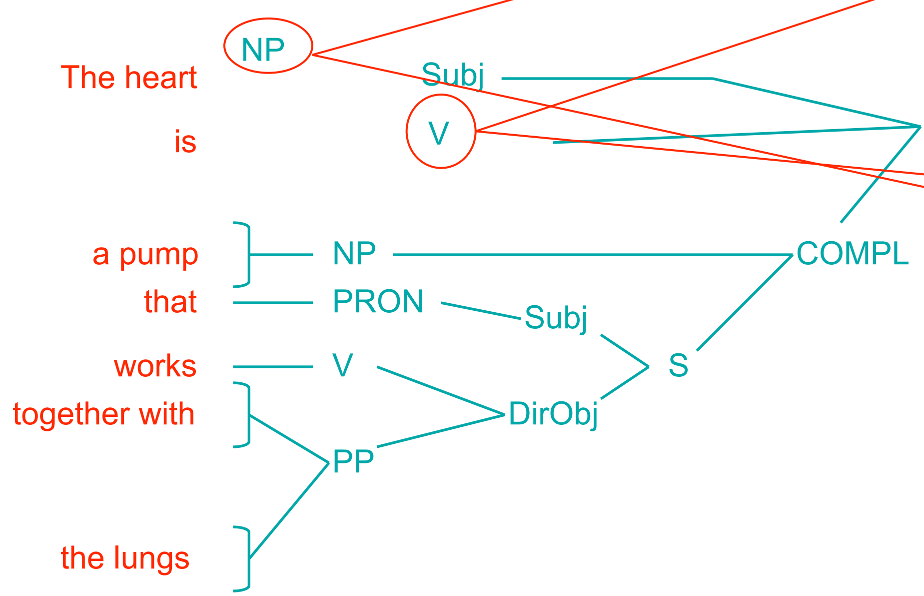
Step 3: Creating Logical Form

Small set of rules: all nouns treated same way; all verbs use case frame structure

- Almost each word (each parse tree part) produces one or more LF expressions
- Variables *x0* and *e0* rep instances of entities and event occurrences
- This is an intermediate form: not the final output delivered to KR

x0: the heart
x1: the pump
x3: the lung

e0: x0 is x1
e1: x1 works
e2: e1 together_with x3



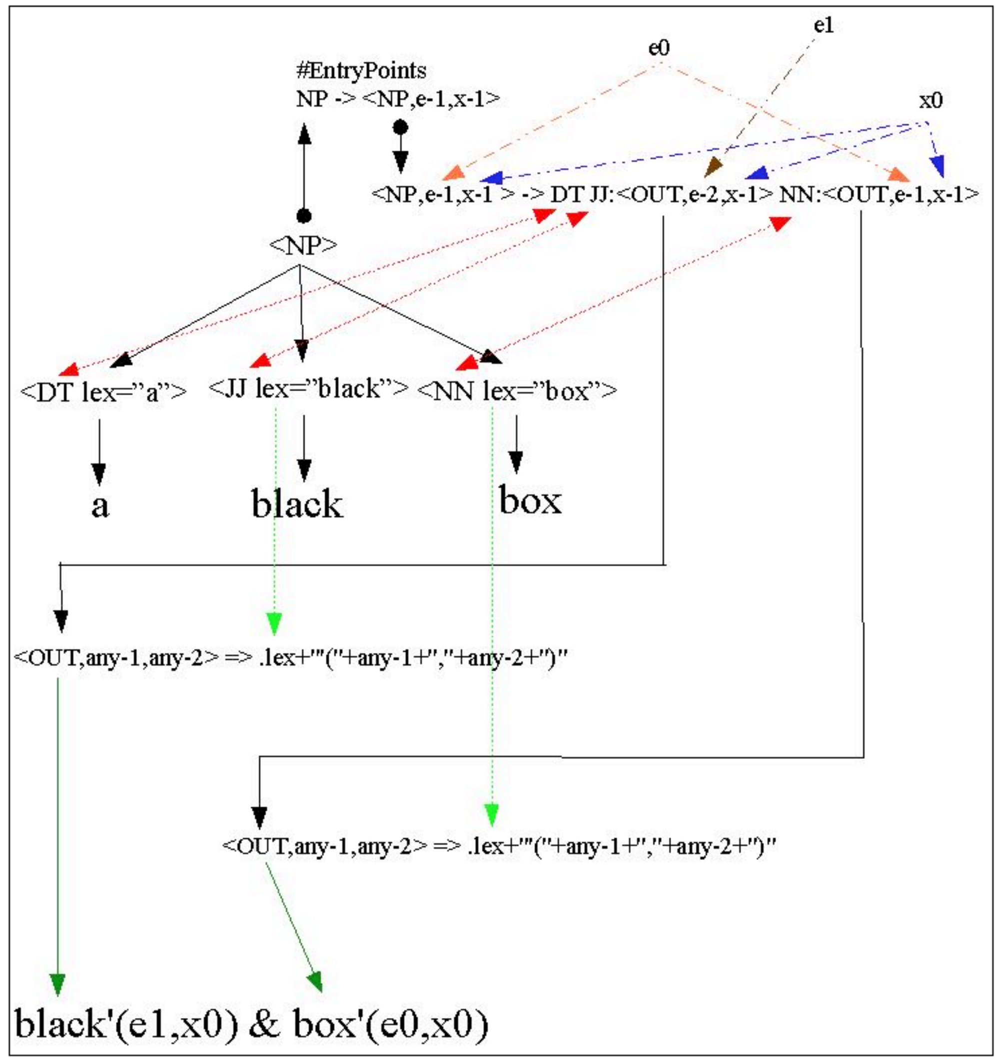
LF output:

is(e0,x0,x1)
heart-nn(x0)
pump-nn(x1)
work-vb(e1)
lung-nn(x3)
together_with(e2,e1,x3)
agent_of2(x3,e1)
agent_of1(x1,e1)

USC INFORMATION SCIENCES INSTITUTE

Some aspects of LF

- LF ('Hobbs Normal Form)' is a simplified semantic notation, using dependency tree structure
- Represents surface-level 'semantics' of specific phenomena:
  - determiners
  - plurals (give rise to sets)
  - explicit eventuality of presence or occurrence of something
- Does not:
  - represent semantics of open-class words (but WSD being added in Möbius07)
  - canonicalize words
  - handle complex NPs

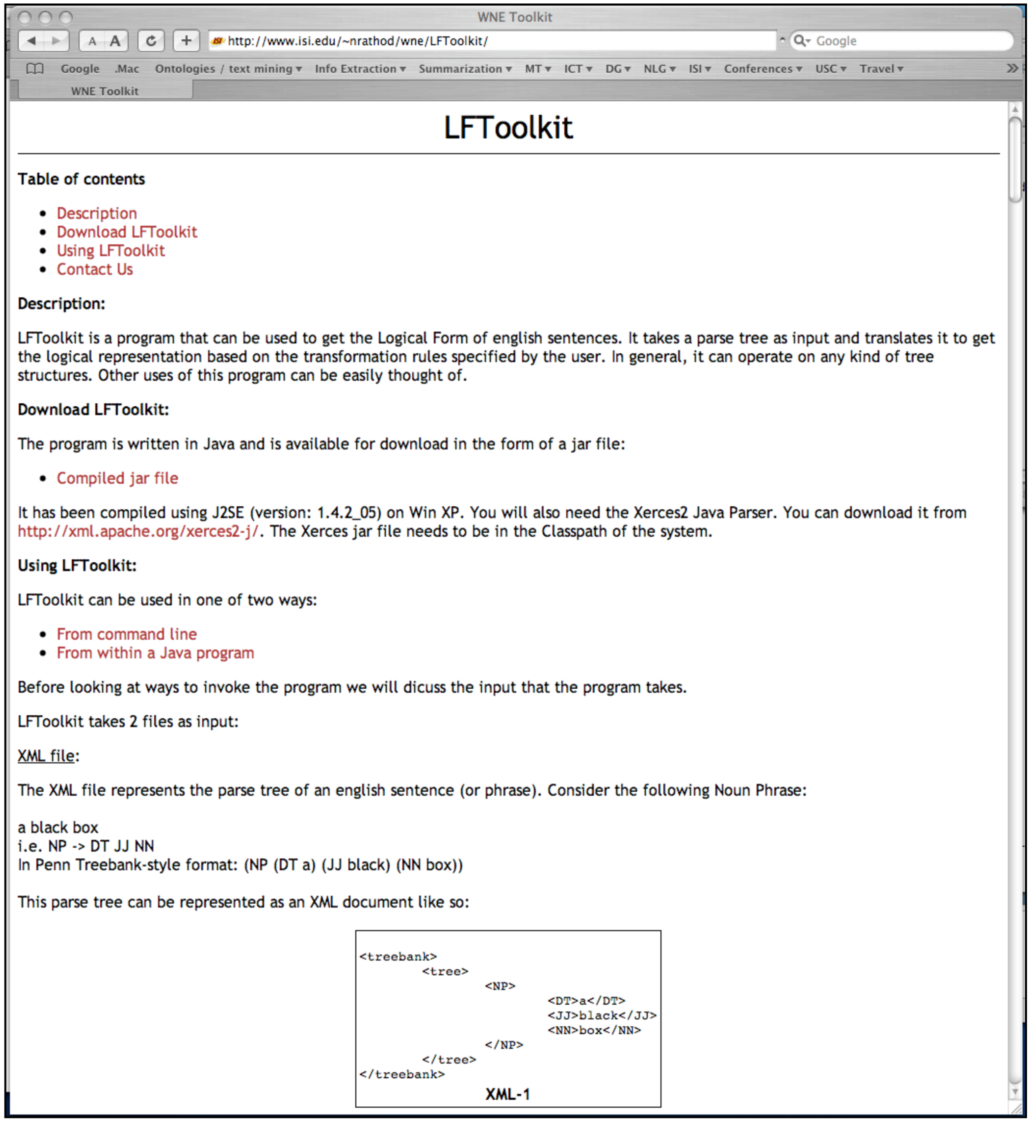


USC INFORMATION SCIENCES INSTITUTE

LF Toolkit

- Built at ISI by Jerry Hobbs and student
- 2007: Contains rules for converting Charniak parser output to LF
- Almost all WordNet glosses already converted to LF

- Download and build your own rules to convert Penn Treebank-style syntax trees into LF
- See http://www.isi.edu/~nrathod/wne/LFToolkit/


WNE Toolkit

http://www.isi.edu/~nrathod/wne/LFToolkit/

Google .Mac Ontologies / text mining Info Extraction Summarization MT ICT DG NLG ISI Conferences USC Travel

WNE Toolkit

LFToolkit

Table of contents

- Description
- Download LFToolkit
- Using LFToolkit
- Contact Us

Description:

LFToolkit is a program that can be used to get the Logical Form of english sentences. It takes a parse tree as input and translates it to get the logical representation based on the transformation rules specified by the user. In general, it can operate on any kind of tree structures. Other uses of this program can be easily thought of.

Download LFToolkit:

The program is written in Java and is available for download in the form of a jar file:

- Compiled jar file

It has been compiled using J2SE (version: 1.4.2_05) on Win XP. You will also need the Xerces2 Java Parser. You can download it from http://xml.apache.org/xerces2-j/. The Xerces jar file needs to be in the Classpath of the system.

Using LFToolkit:

LFToolkit can be used in one of two ways:

- From command line
- From within a Java program

Before looking at ways to invoke the program we will dicuss the input that the program takes.

LFToolkit takes 2 files as input:

XML file:

The XML file represents the parse tree of an english sentence (or phrase). Consider the following Noun Phrase:

a black box
i.e. NP -> DT JJ NN
In Penn Treebank-style format: (NP (DT a) (JJ black) (NN box))

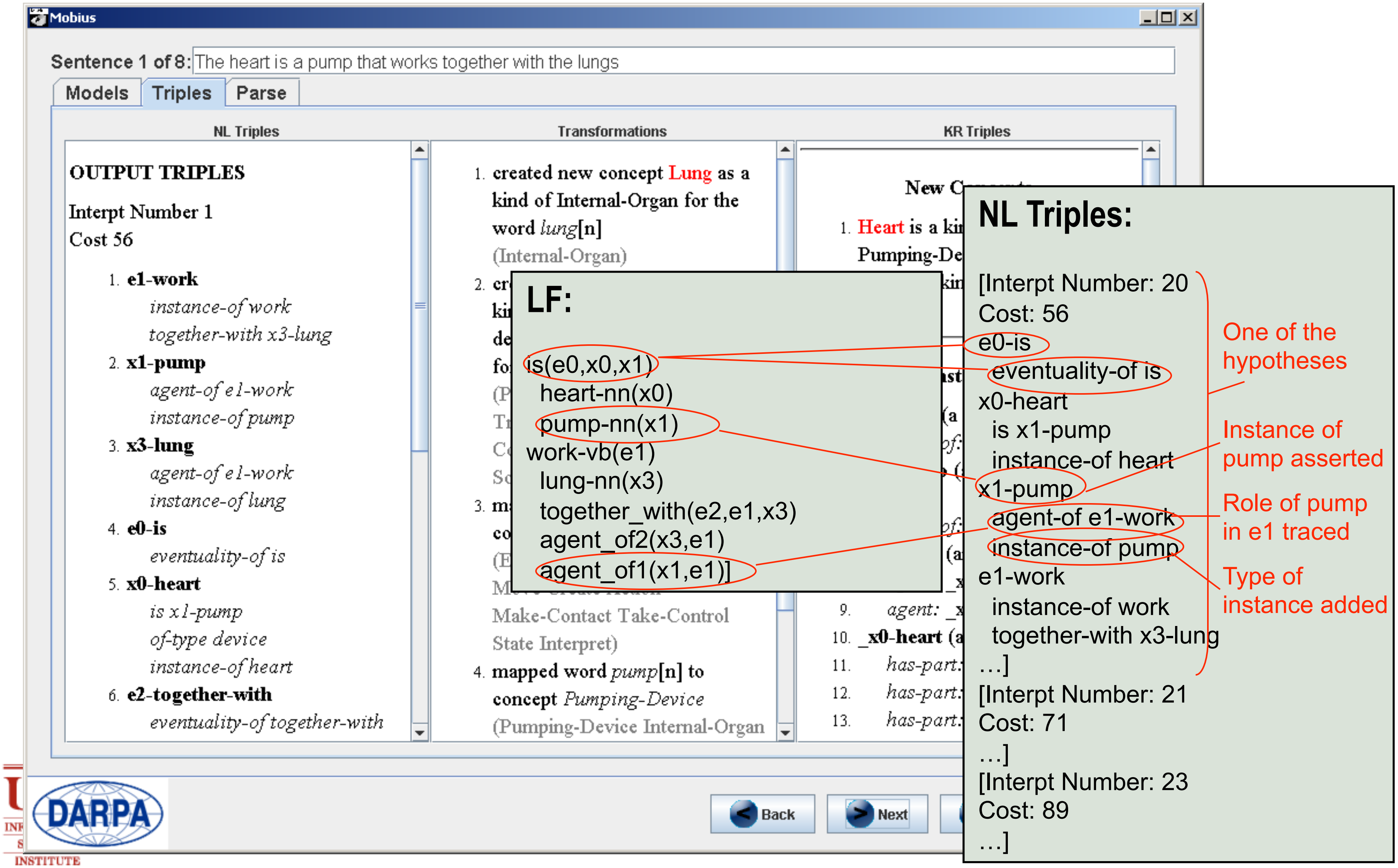
This parse tree can be represented as an XML document like so:

```
<treebank>
        <tree>
                <NP>
                        <DT>a</DT>
                        <JJ>black</JJ>
                        <NN>box</NN>
                </NP>
        </tree>
</treebank>
```

XML-1

USC INFORMATION SCIENCES INSTITUTE

Step 4: Forming NL triples



LF → NL triple rewrite rules

Noun	pump-nn(x1) → x1-pump instance-of pump	1 rule
Verb	work-vb(e0,x1) → e0-work instance-of work agent-of x1,e0	1 rule
Adj	right-adj(x1) → x1-<noun> mod right	1 rule
Cardinality	15(x1) → x1-<noun> cardinality 15	1 rule
Conjunction	and(x2,x3,x4) → x2-<role-of verb> x2 and x3 x2 and x4	1 rule for AND, 1 for OR; none yet for other conjunctions
Other	is(e0,x0,x1) → e0-is eventuality-of is	1 rule for BE, a few more for others

Step 5: Triples integrated by KR module

- KR module contains:
  - Ontology
  - Starting domain models
  - Growing expertise model from text(s) contents
- Activities:
  - Accept NL triples
  - Reformat as needed
  - Match triples against existing model(s):
    - If match, just add in
    - If no match, and no inconsistency, assume tentatively
    - If mismatch, (potentially) spawn new hypothesis (set of triples)
  - If needed, generate diagnostic triples and feed back to main system for Targeted Reading

How this all works is a whole separate talk!

USC INFORMATION SCIENCES INSTITUTE

Talk overview

1. Introduction: The dream
2. Background: DARPA LbR seedlings in 2005
3. The Möbius experiment 2006–07
   – Partners, architecture, and domain
   – NL interpretation: Parsing, Logical Form, Abduction
   – Deep inference and shallow broad coverage
4. Tests and evaluations
5. What did we learn? The Future

USC INFORMATION SCIENCES INSTITUTE

Why are there gaps and omissions?

- The big cause of failure is narrow coverage :
  - Not enough words
  - Not enough relations
- Example:
  - NL triples still at surface semantics level:
    - *The heart squirts blood…* (*squirt heart blood*)
    - *The heart pumps blood…* (*pump heart blood*)
    - *The heart makes blood move…* (*make-move heart blood*)
  - KR expects standardized input at deeper level:
    - all must be (*pump heart blood*)

too many different symbols!

- How to coerce term types, and to 'invent' linking knowledge?

USC INFORMATION SCIENCES INSTITUTE

Abduction

If you know

A

and you have the abductive rule (axiom)

B & A → A

then you can assume also

B

(for a certain penalty or cost).

So: build lots of abductive rules to hypothesize gap-filling knowledge.

Mini-TACITUS

- LF → NL triple rewriting can include some **gap filling**
- How? Using Mini-TACITUS (Hobbs et al. 99):
  - Paradigm: **abductive reasoning**
  - General abductive axiom scheme: [*a OR (b AND c) OR …* → *d*]

    Abductive reasoning builds all hypotheses that *might* lead to input:
    - Given *d*, assume *a* (= hypothesis 1), or assume *b and c* (= hypothesis 2), etc.
  - Associate *cost* (number) with each hypothesis, depending on number of assumptions, etc. — fewer assumptions is better
  - Output: ranked list of hypotheses

- Using axioms:
  - Initially start with minimal set of 'rep. rewriting' axioms, to handle shallow semantic phenomena (e.g., *plurals*)
  - Can also include domain-specific axioms that represent domain model, built by hand (e.g., *blood is a fluid*)
  - Later, try to learn content-based axioms automatically from input

Abduction does the work

- Axiom for getting from anything to PUMP:

  SQUIRT(*x1,x2*) $\rightarrow$ PUMP(*x1,x2*)

- Axiom for linking up the arguments:

  DEVICE(*x1*) & FLUID(*x2*) & ?(*x1,x2*) $\rightarrow$ PUMP(*x1,x2*)

- Example input:

  "…the heart expands, fills with blood, and squirts the blood…"

  gives

  - HEART(*x4*) & EXPAND(*?,x4*)
  - BLOOD(*x6*) & FILL(*?,x6*)
  - BLOOD(*x6*) & SQUIRT(*?,x6*)

  Also have, from knowledge base:

  - DEVICE(*heart*)
  - FLUID(*blood*)

SQUIRT(*x1,x2*)

PUMP(*x1,x2*)

heart=x4=x1

blood=x6=x2

DEVICE(*x1*)

FLUID(*x2*)

HEART(*x1*)

BLOOD(*x2*)

PUMP(*HEART-x1,BLOOD-x2*)

Adding abductive axioms

Axiom1: device(x1) & fluid(x2) & fill(x1,x2) → fill(x1,x2)
Axiom2: device(x1) & heart(x1) → heart(x1)
Axiom3: fluid(x1) & blood(x1) → blood(x1)

Ex incoming LF: heart(x1) …fill(x2,x3)… blood(x4)

Axiom1 ***Axiom2*** ***Axiom3***

device(x1) & heart(x1) …

device(x2) & fluid(x3) & fill(x2,x3) ...

fluid(x4) & blood(x4)

Factor on *device* and *fluid* (so x1 = x2, x3 = x4):

device(x1) & heart(x1) ... fill(x1,x3) ... fluid(x3) & blood(x3)

USC INFORMATION SCIENCES INSTITUTE

Slide by Noah Friedland

Axioms introduce other terms and connect their variables

Axiom1: device(x1) & fluid(x2) → pump(x1,x2)
Axiom2: device(x1) & heart(x1) → heart(x1)
Axiom3: fluid(x1) & blood(x1) → blood(x1)

LF: "…heart(x1)….squirt(x2,y2)…. blood(y3)…"

device(x1)&heart(x1) ~~X~~ fluid(y3)&blood(y3)

x1=x2 y2=y3

"…heart(x2)….pump(x2,y2)…. blood(y2)…"

"pump" replaces "squirt"

USC INFORMATION SCIENCES INSTITUTE

Slide by Noah Friedland

Need only a few general axiom schemas

ISA-of-entity(x1)&entity(x1) → entity(x1)

e.g. fluid(x1)&blood(x1) → blood(x1)

"blood is a fluid"

ISA-of-entity(x1)&ISA-of-entity(x2)&action(x1,x2) → action(x1,x2)

e.g. device(x1)&fluid(x2)&pump(x1,x2) → pump(x1,x2)

"pumping happens by devices on fluids"

KR-preferred-term(x1,x2) → action(x1,x2)

e.g. pump(x1,x2) → squirt(x1,x2)

"squirting is pumping"

USC INFORMATION SCIENCES INSTITUTE

Slide by Noah Friedland

The challenge of scaling up

- Standardize semantic symbols and relations and convert (more of) the free-form NL expressions into the kinds of triples that KR can absorb
- Challenges:
  1. (Semi-)automatically create **specific axioms**
  2. Distill output to deeper level: **replace symbols** with KR-preferred standard ones

- How to create (hundreds of new) abductive axioms, instantiated from the basic schemas?
- How to find *all* (thousands?) of relevant words/ phrases for the instantiation process?

USC INFORMATION SCIENCES INSTITUTE

Extending coverage 1: More words

1. **Corpus-based strategy**: Extracted all sentences that contain anchor terms
   - Corpus: 10GB text, extracted from web and cleaned (done in 10 blocks of files)
   - Anchor terms: "heart" and "blood" (most central for Pump and Organ models)
   - Result: approx. 15,000 sentences
   - Most not useful
2. **Extract all fragments**:
   - Filter useful fragments: “heart” precedes “blood”; anchor words are close in text
   - Result:

Corpus\Word separati	1 word	3 words	4 words	5 words	6 words	7 words
c0	17	25	24	21	16	16
c1	0	25	0	0	0	0
c2	0	34	0	0	0	2
c3	0	34	0	1	1	1
c4	1	14	0	0	0	0
c5	28	33	20	23	4	9
c6	0	35	0	3	0	4
c7	0	42	1	0	0	1
c8	0	23	0	0	0	0
c9	0	33	0	0	0	0
total	**46**	**298**	**45**	**48**	**21**	**33**

3. **Extract all useful words and phrases:**
   - Inspection shows most fragments relate to heart disease
   - Manually extracted 118 relevant words/phrases:

Total	**46**	**298**	**45**	**48**	**21**	**33**	**Total 118**
Manual Elimination	**4**	**44**	**21**	**21**	**8**	**20**	
	(Unique)	(not-unique)	(not-unique`	(not-unique`	(not-unique)	(not-unique)	

USC INFORMATION SCIENCES INSTITUTE

Extending coverage 2: More axioms

- Next: must **create axioms for these words/phrases** as domain model concept descriptions:
  - Problem: Not trivial to form useful triples (that KR can absorb) from relevant NL expressions without ‘cheating’
  - **Option 1: We can cheat** (e.g., by manually mapping each word like “squirt”, “pump”, “move” to PUMP using many fixed rules)
  - **Option 2: We can do it right** (e.g., by *inferring* PUMP, since “squirting / flowing / moving” *of a fluid* always requires pumping)
- To do this, we **harvest text for the logical parts of the argument**:
  - Need a general scheme for derivation, plus axioms that know under which conditions PUMP is logically derivable from the words
  - Must ensure that the axioms connect arguments (left and right sides of triple)
    - Must work even if one of the two sides “heart” or “blood” is missing:
      1. Get all the KB predicates, like *PUMP(x,y)*
      2. Use phrase finding algorithm/text mining/WordNet/Omega/ISI paraphraser/etc. to expand these words to their (quasi-)synonyms (e.g., *squirt*)
      3. Get a list of all the relevant entities, e.g., *heart, blood*
      4. Get a list of all the relevant relations that relate them
      5. Write axioms for constraints on arguments of Knowledge Base predicates
      6. Write corresponding axioms saying what kinds the entities are

Axiom development results

- Total axioms before automated creation: approx. 120
- For 118 new sentence fragments harvested, built 35 new axioms — 22% increase
- Now can handle new phrasing:

  e.g., “The human heart is responsible for circulating blood…”

- Tested effect on coverage with and without axioms
  - Test set of 24 new intro-type sentences that (in most cases) should indicate PUMP or ORGAN
  - Results: 12.7% increase in NL triple coverage, 15.1% increase in KR triples, and doubled coverage on matching to KB models, to 100% (model changed or new)

This actually worked

USC INFORMATION SCIENCES INSTITUTE

Coverage test

	BEFORE	MIDWAY	BEFORE	MIDWAY	BEFORE	MIDWAY	
	NL Triple generation		KR Triple recognition		model generation		
S1	22	24	15	17	0	0	The human heart is a fist-sized organ responsible for circulating blood through the vascular system
S2	41	43	32	31	1	1	The heart is a hollow, muscular body part in creatures like us which is responsible for pumping blood through the body by repeated, rhythmic contractions
S3	25	26	12	13	1	1	The human body uses a liquid medium, blood, that must be circulated continuously throughout the entire human body.
S4	25	hangs	15	hangs	1	1	The heart is an internal organ in animals which function is to pump blood throughout the body
S5	26	28	17	17	0	0	The heart is an involuntary muscle that pumps blood throughout the body by contracting (and relaxing) rhythmically
S6	17	19	11	13	0	0	A heart is an organ of the human body that is used to circulate blood
S7	22	23	15	16	1	1	A human heart is one compartment of the human body located inside the human body
S8	26	28	15	18	1	1	The heart is structurally dynamic and part of a pressurized system of tubes filled with liquid
S9	18	19	8	8	1	1	The heart is one specific enlarged area of this tube system
S10	14	15	10	11	1	1	It contracts and then stops contracting, repeatedly
S11	12	13	7	8	0	0	The heart is a muscular organ with two sides
S12	42	44	28	30	1	1	One side receives blood from the body and pumps it through the lungs to eliminate the waste product carbon dioxide and replenish needed oxygen
S13	kr hangs	46	kr hangs	29	0	0	Every human relies on exactly one vital organ, centrally located inside their body, called the heart, to provide the motive force for blood circulation
S14	10	11	5	6	0	0	The heart is an important organ without which no mammalian organism can survive
S15	18	20	16	15	1	1	The heart is a pump which drives blood circulation through the mammalian body
S16	22	25	9	12	0	0	The heart takes in impure blood and pumps out pure blood
S17	12	13	6	7	0	0	The heart is an organ (part, component) of the human body
S18	19	21	12	14	0	0	Its role is to pump blood to the rest of the body
S19	6	7	3	4	1	1	The heart is a muscle
S20	19	21	12	15	1	1	Its role is to pump blood to the rest of the body
S21	26	28	17	17	0	0	The heart is an involuntary muscle that pumps blood throughout the body by contracting (and relaxing) rhythmically
S22	21	25	15	18	1	1	The human heart contracts to send blood to the lungs and the rest of the body
S23	25	28	15	19	0	0	Deoxygenated blood enters the heart through the right atrium and right ventricle to the lungs where the blood becomes oxygenated
S24	22	25	9	12	0	0	The heart takes in impure blood and pumps out pure blood
	490	552	304	350	12	12	
					0.5	0.5	
% incr	12.7%		15.1%				

USC INFORMATION SCIENCES INSTITUTE

Talk overview

1. Introduction: The dream
2. Background: DARPA LbR seedlings in 2005
3. The Möbius experiment 2006–07
   - Partners, architecture, and domain
   - NL interpretation: Parsing, Logical Form, Abduction
   - Deep inference and shallow broad coverage
4. Tests and evaluations
5. What did we learn? The Future

USC
INFORMATION SCIENCES INSTITUTE

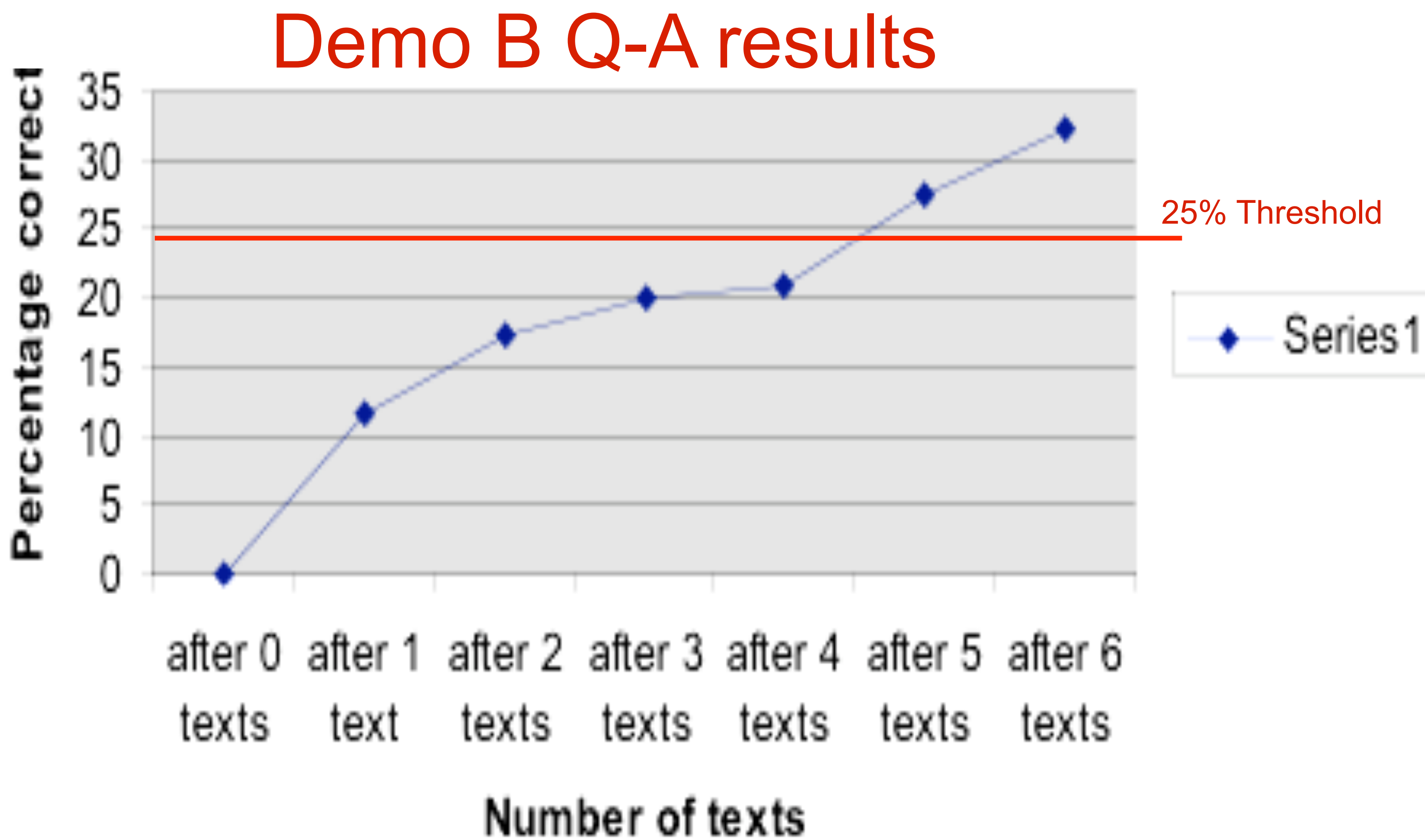
Möbius Y2 accomplishments

Item	Demo A	Demo B	Comments
End-to-end Q-A	PAD Target 5% 21.5/138 = 15.6%	PAD Target 25% 104/321 = 32.4%	Both exceeded PAD requirements
Points lost from Q-A	unknown	55; upper bound on score: 159/321 = 49.5%	Lower bound, Q-A system recall: 104/159 = 65.4%
Data	3 topics, 122 sentences, 57 questions, total score 138	1 topic, 166 sentences, 127 questions, total score 321	

USC INFORMATION SCIENCES INSTITUTE

Slide by Noah Friedland

End-to-end system performance

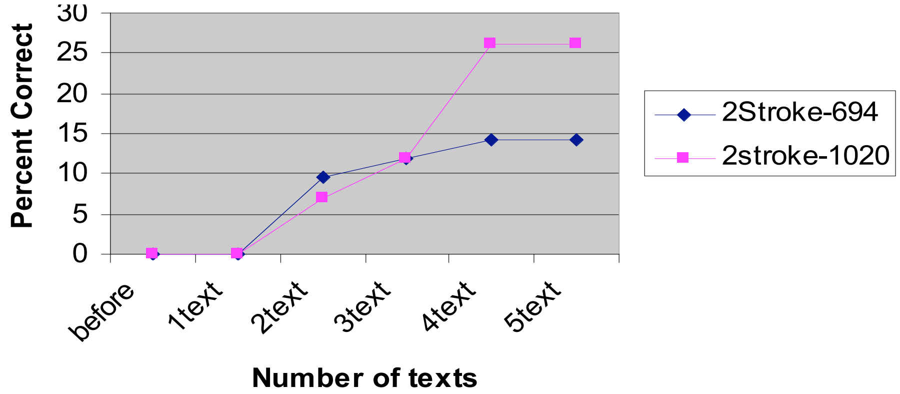


USC INFORMATION SCIENCES INSTITUTE

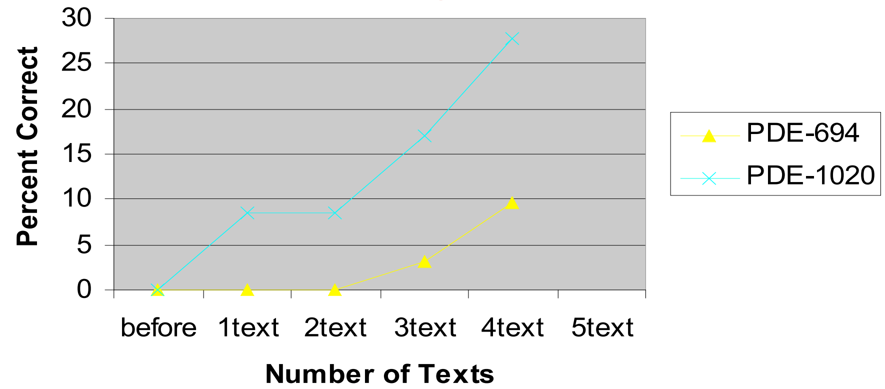
Slide by Noah Friedland

Comparative Demo A-B performance on Demo A data

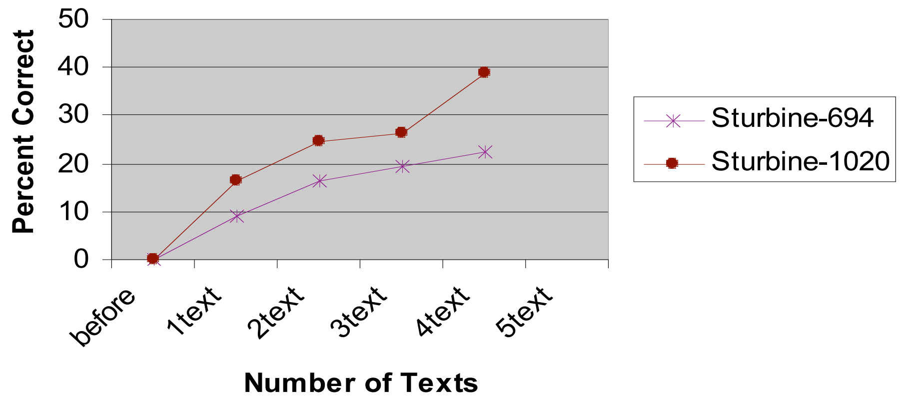
Comparing 2Stroke



Comparing PDE



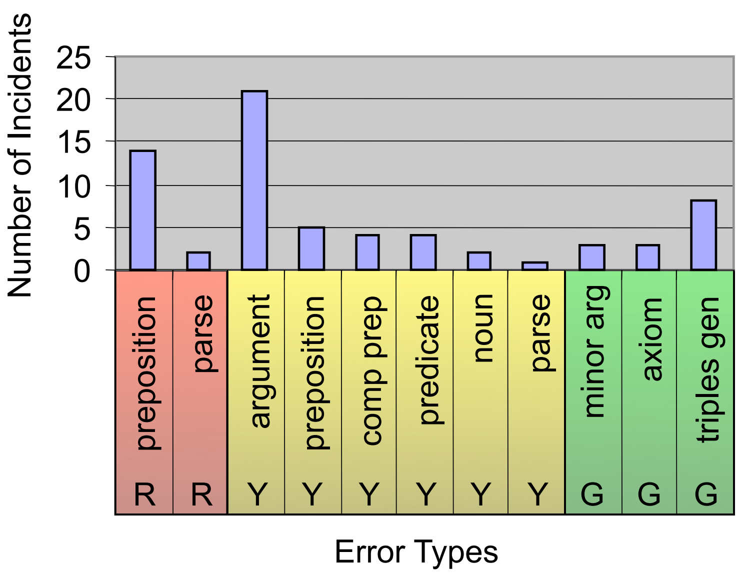
Comparing Steam Turbine



Slide by Noah Friedland

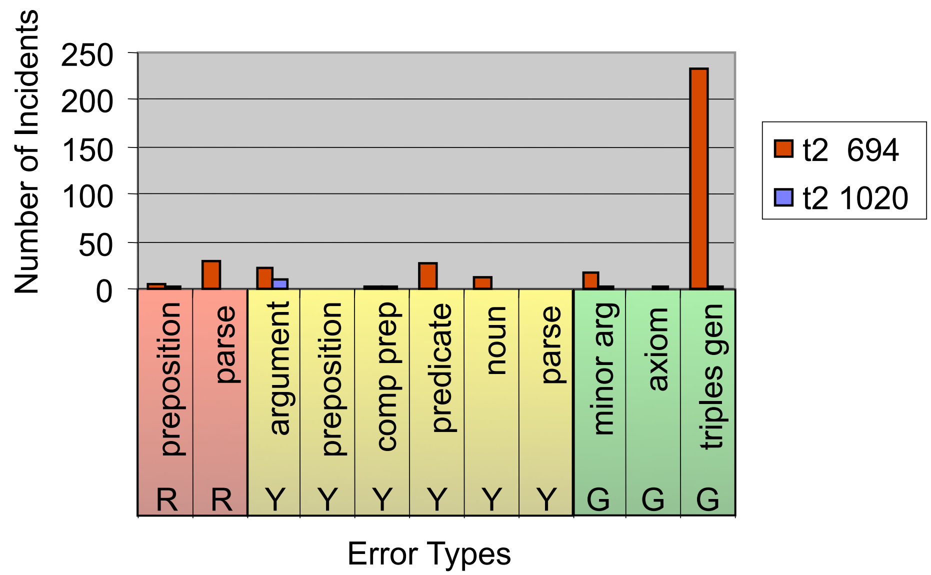
NL error analysis

Build 1020, all demo B texts



Total errors

Comparing 694 vs 1020 on demo B, Text2



Text2 errors

USC INFORMATION SCIENCES INSTITUTE

Slide by Noah Friedland

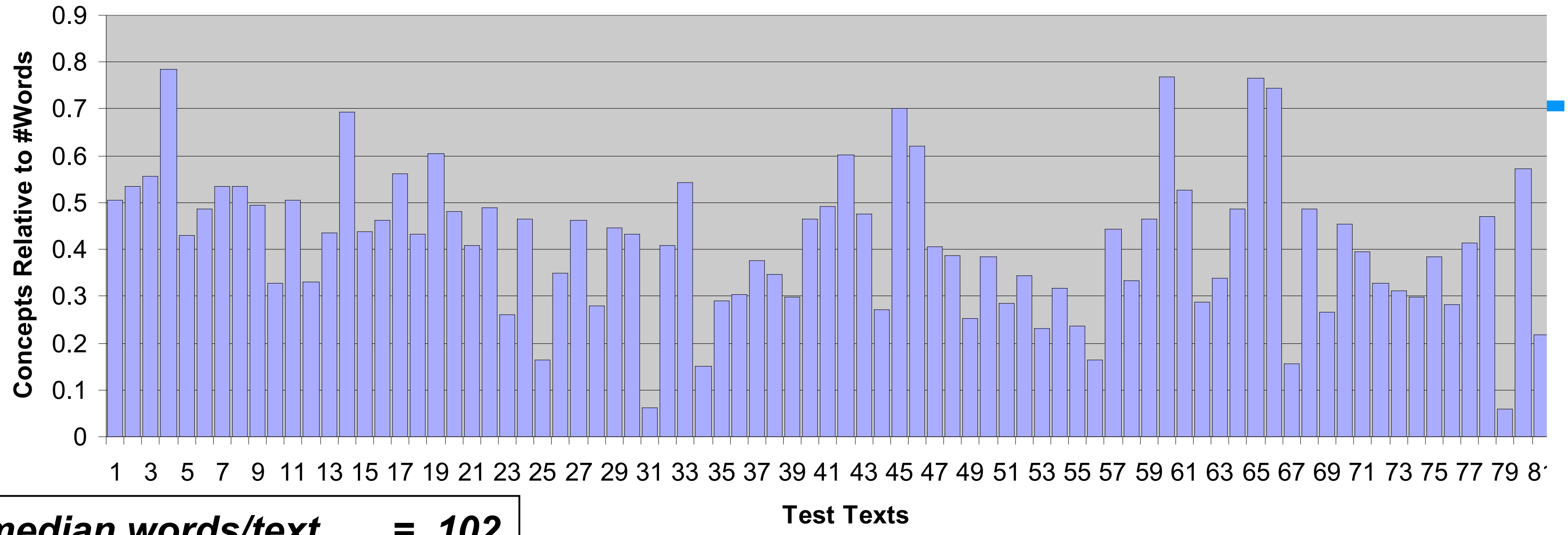
3 Questions

1. Can Möbius learn (many) new concepts and axioms? How many?
2. Can Möbius learn really *new* knowledge, or only variants of what was in the seed KB to start with?
3. Given a "finite" domain and task set: Does a Möbius system's learning rate decrease as it reads more texts about the domain? (I.e., does it learn fewer new facts from a text in a domain if it has already read other similar texts in the domain?)

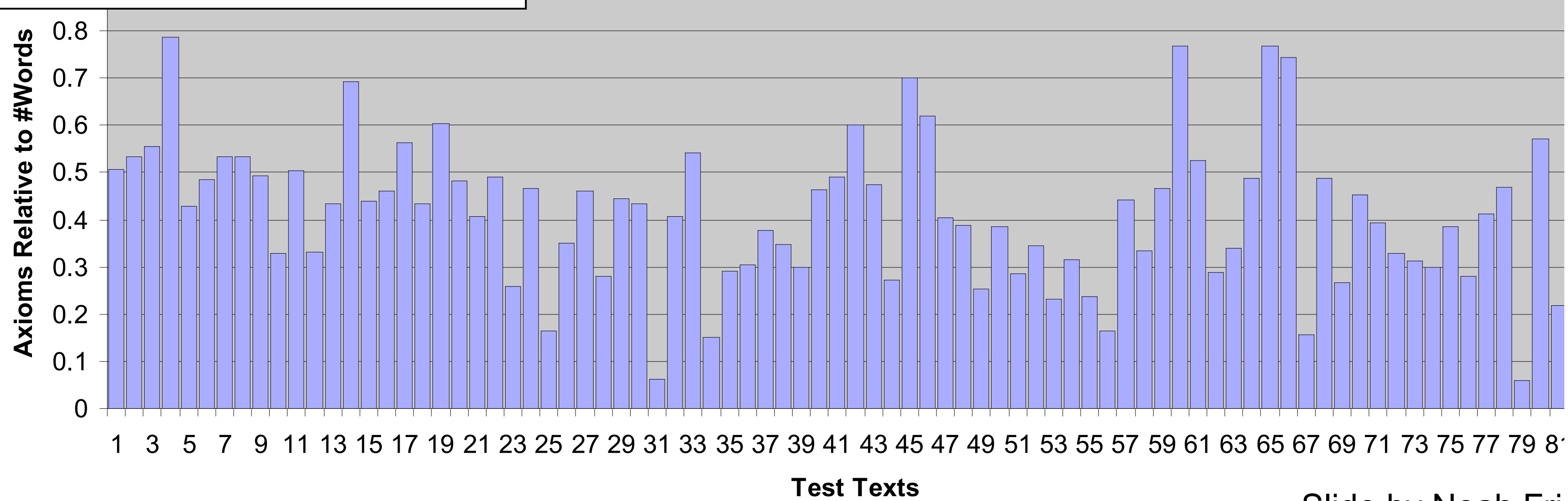
1. The ‘growing from seed’ question

- Question 1: Can Möbius learn (many) new concepts and axioms? How many?
- Experiment: Obtained over 80 texts from web, encyclopedias, etc. Developed system on some, tested on others. Counted concepts and axioms learned
  - Explored Knowledge Base before and after reading

New words and axioms learned, for each text



median words/text = 102
median concepts/text = 11
median axioms/text = 40



Slide by Noah Friedland

2. The ‘learning new info’ question

- Question 2: Can Möbius learn really *new* knowledge, or only variants of what was in the seed KB to start with?
  - If it’s really learning new stuff, Möbius should perform equally on texts that aren’t about *hearts* as pumps (e.g., texts about other kinds of pumps)
  - Results should include some reasonable knowledge about the non-heart subject
  - Mobius should not “hallucinate” heart-like knowledge
- Experiment:
  - Gave Möbius 6 texts unrelated to hearts but talking about “pump” or “pumping”
    - 4 real texts; 2 invented (intended to confuse)
  - Results are similar to experiments with random heart texts
    - Möbius does fairly well on some texts, poorly on others
    - <u>Mobius</u> learns some nice interpretations and some bad ones

Results

text	*#sentences (avg. words/sent)*	*learned concepts*	*unique axioms*	*unique axioms / sentence*
83 heart test texts (medians)	6 (16.8)	11	40	6.3
Airlift pump	4 (12.8)	5	26	6.5
Bicycle pump	7 (16.0)	13	48	6.9
Breast pump	6 (15.5)	13	24	4.0
Peristaltic pump	6 (18.5)	14	58	9.7
Harmonium (organ)	2 (10.5)	4	18	9.0
Shoe (“pump”)	2 (7.0)	2	2	1.0

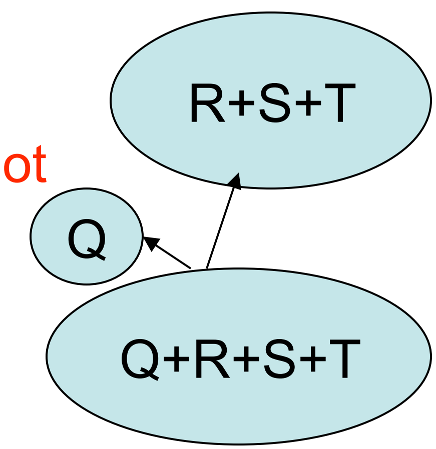
Möbius did use some of its human-authored knowledge to extract knowledge pertaining to pumps (and pump confusers) in other domains

Slide by Noah Friedland

3. The convergence hypothesis

- Question 3: Given a “finite” domain and task set: Does a Möbius system’s learning rate decrease as it reads more texts about the domain? (I.e., does it learn fewer new facts from a text in a domain if it has already read other similar texts in the domain?)
- Experiment:
  - Read four texts Q+R+S+T together; count new concepts C1 and axioms A1
  - Read texts R+S+T leaving out text Q; count new concepts C2 and axioms A2
  - Read text Q alone; count new concepts and axioms C3 and A3
  - Repeat steps 2 and 3, leaving out texts R, S and T in turn

- If (C1 - C2) < C3, some of the concepts in text Q are redundant with those in R+S+T, and were not learned twice
- If (A1 - A2) < A3, same for some of the axioms in Q



USC INFORMATION SCIENCES INSTITUTE

Results

text alone	C3	A3	leave one out	C2	A2	C1-C2	A1-A2
Q	4	16	R+S+T	14	72	**0**	**5**
R	9	30	Q+S+T	10	56	**4**	**21**
S	6	33	Q+R+T	12	48	**2**	**29**
T	8	15	Q+R+S	11	66	**3**	**11**

all texts	C1	A1
Q+R+S+T	14	77

green = (x1 - x2) < x3

Möbius could 'recognize' redundancy across texts and did not simply build (near-)duplicate concepts and axioms

USC INFORMATION SCIENCES INSTITUTE

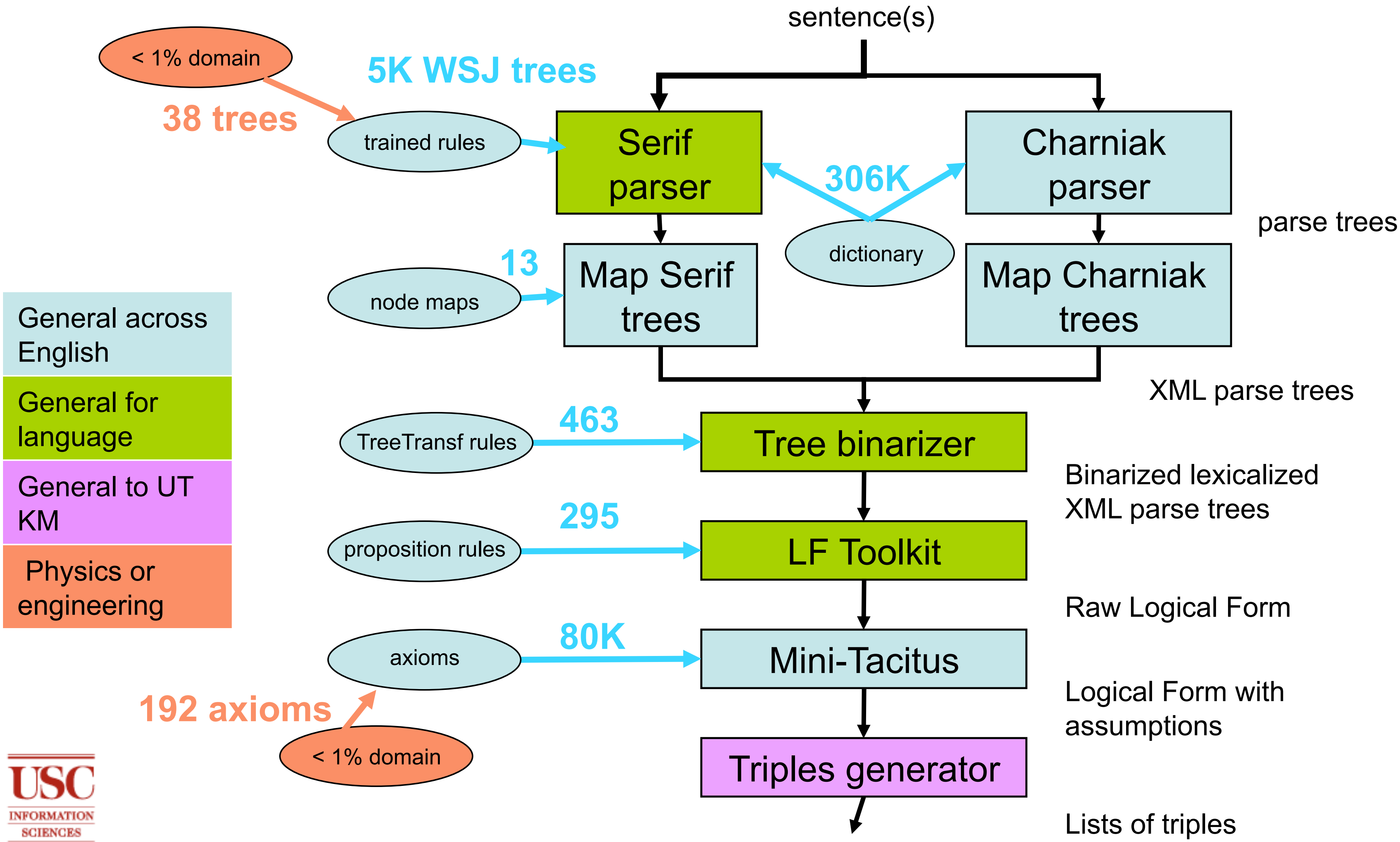
Slide by Noah Friedland

Talk overview

1. Introduction: The dream
2. Background: DARPA LbR seedlings in 2005
3. The Möbius experiment 2006–07
   – Partners, architecture, and domain
   – NL interpretation: Parsing, Logical Form, Abduction
   – Deep inference and shallow broad coverage
4. Tests and evaluations
5. What did we learn? The Future

USC INFORMATION SCIENCES INSTITUTE

How many & which kinds of knowledge?



Slide by Noah Friedland

What did we learn 1?

- What's different from the 1970s?
  - Large-scale parsing—possible ? ✓ ✗
  - Large-scale LF creation—bottleneck ? ✓ ✗
  - Large-scale deeper (triples) creation—ok, for simple semantic phenomena ? ✓ ✗
  - Semantic phenomena—manageable (?), as needed for the text and the questions, but far from fully understood ? ✓ ✗
  - Inference—more than in the 70s, but is still a bottleneck ? ✓ ✗
  - Evaluation—unknown how to do this ? ✓ ✗

- Text is full of 'gaps' and 'loosespeak' — it provides only the framework into which the understander fills the rich background and details through world knowledge and inference

USC
INFORMATION
SCIENCES
INSTITUTE

What did we learn 2?

- It is possible to have a system learn new knowledge and bootstrap itself automatically, but it requires a lot of careful thought about the seed models and the representation
  - Hobbs-style shallow semantic notation is workable because it contains almost no semantics — no ontology, no complex phenomena, just 'ontological promiscuity' approach and arg structure

- Large-scale general-domain LbR is not feasible yet because of the difficulty of:
  - Obtaining **enough axioms** for inference
  - …(can one do this on demand, during reading?)
  - Building rich enough **seed models**
  - …(can one build up a library of standard seed models?)
  - Representing some of the common required **complex semantic phenomena** (negation, modality, discourse-level implication/ entailment, etc.)
  - …(can one implement the work of linguists, logicians, etc.?)

- BUT for circumscribed domains, there is hope…

USC INFORMATION SCIENCES INSTITUTE

What is a shallow representation?

- HNF is similar to Davidsonian semantics:
  - Just simple terms, no canonicalization
  - No disambiguation
  - Simple verb arg structure
  - No explicit relations
    - (*apple X) & (red X)* vs. *(isa X apple) & (color X red)*
  - Semantic phenomena added one by one:
    - Determiners, plurals, negation…
- Ontological promiscuity — there's almost nothing to work with

USC INFORMATION SCIENCES INSTITUTE

Implications for NLP

- Short-term:
  - LbR is not yet ready to support much of NLP or KR&R
  - Both NLP and KR&R can provide useful information about semantic representation (‘triples’) design for the Global Language
- Longer-term:
  - It is unclear how to reconcile statistically built transformations / transfer rules with LbR-like knowledge and reasoning in general
  - LbR capabilities can however help with certain specific phenomena for NLP:
    - Coreference
    - Wordsense disambiguation
    - Argument attachment
- So should we try?
  - Yes: without trying we will never join together KR&R and NLP, and neither is adequate alone
  - No: we are still too far off:
    - KR is too brittle: requires correctness and works only at small scale
    - NLP is too crude: works statistically but has too many errors, and is too shallow

Should we explore LbR, or not?
Your vote, please?

Thank you!