

INTRODUCTION

REPRESENTATIONS OF MEANING

Theme

Language is not simply the words that you use, syntactically structured. Language is a method of communicating information so as to achieve goals to arrive at desired states and interpretations of the world. It is a highly complex procedure involving various types of knowledge and processes. Also, a very brief overview of issues and methods of representing meaning, tailored to shallow semantic and language generation.

Summary of Contents

1. Introduction: What is language?

Compare language of humans to Natural Language Processing (NLP). Do this in a sentence-by-sentence readthrough and discussion of a text, with stepwise illustration and gradual building up of pertinent processes, memory structures, goals, affects, inferences, etc., in order to illustrate diversity of knowledge and processes required to understand a question, formulate an answer, and generate it. (Have three volunteers to blackboard; one records words, one records the concepts 'behind' words, and one records interrelationships among concepts to get overall meanings.)

Note all relevant information on blackboard as it is encountered, into one of several columns:

World knowledge of various kinds; (info)

Episodic knowledge and reminders

Situational effects

Interpersonal effects

and then the process:

Expectations: activated, and realized or flushed.

2. Stages and levels of processing

Draw out the NLP pipeline: from words through parsing etc. to internal representation, and then out again to words. Draw some applications in the middle.

Show the substages: POS tagging, discourse analysis, macro- and micro-planning.

Describe lexis, syntax, discourse, semantics, pragmatics.

nonsense — *green deeply ideas the the sleep colorless*

syntax: — *colorless green ideas sleep furiously* (Chomsky)

semantics — *happy tired children sleep deeply*

pragmatics — *terrorist =? freedom fighter =? guerrilla* or *get off my foot! =? you're heavy*

3. Types/levels of representation of language

Many types, for different purposes, including:

Lemmatizing	demorphed, individuated word tokens
Tagging	Part of Speech (POS) tags
Syntactic parsing	Parse trees (many theories of grammar)
Semantic analysis	Semantic readings (many theories of semantics)
Discourse analysis	Discourse structures (trees?) Cohesiveness graphs (lexical chains) Information structure (theme/rheme, given/new, topic/focus)
Pragmatic analysis	Speech Acts; Locution/Illocution/Perlocution Interpersonal and Textual metafunctions (Systemic grammar)

4. (Shallow) Semantics: The Core, the Tar Pit

Three aspects:

1. **Form** (formalisms and notation systems)

- Predicate notation: (red ball) = the/a ball has the color red
- Frame notation: [X1 (:type Ball) (:color Red)] = [(type X1 Ball) (:color X1 Red)]

2. **Formality**

- **Formal**: first order logic and its extensions (higher orders, modality, probabilities, etc.)
- **Informal**: content-only representation that supports no formal inference

3. **Content**

- **Parsimonious** (few types, much composition, lots of inference)
- **Profligate** (many types, problems constraining composition, little easy inference)

Basic semantic types

- Relations (:color, :number :smaller-than :spouse...)
- Events (Build Eat Explode Fly...)
- Objects (Car Human Idea Democracy...)
- States (Happiness Hunger Health...)
- State changes (changes in state values)

Each semantic type has its own particular structure and relations. There is no standard system for representing them, however, and it takes quite a lot of experience to build up your own.

How to get the best? Ontologies record all the representation items used in a system: an inventory of terms, usually taxonomized using the IS-A (a-kind-of) relation:

Boy ISA Human ISA Mammal ISA Animal ISA LivingThing ISA PhysicalEntity ISA Object

This allows more specific (lower) entities to ‘inherit’ properties from less specific (higher) ones: everything that is true about Mammals is also true about Humans and about Boys, and everything that is true about Humans is also true about Boys. (The method of defining high-level basic classes and then specializing them by adding features or *differentiae* dates back to Aristotle, who did this all the time and laid the foundation of many of the ways we still see the world.)

Often, for language processing, Ontologies have an Upper Model that records the basic classes needed to support NLP engines. Usually, Upper Models contain between 200 and 2000 entities. The Penman Upper Model was built at ISI in the 1980s to support English language generation, mostly. The Penman Upper Model’s basic types are:

OBJECT

DECOMPOSABLE-OBJ and NONDECOMPOSABLE-OBJ

SOCIAL-OBJ and PHYSICAL-OBJ

NON-CONSCIOUS-THING and CONSCIOUS-BEING

NAMED-OBJ

MISC-STUFF

PROCESS

MATERIAL-PROCESS

NONDIRECTION-ACTION

MOTION-PROC (he rode)

AMBIENT-PROC (it was warm)

DIRECTED-ACTION

CREATIVE-MAT-ACTION (he made the cake)

DISPOSITIVE-MAT-ACTION (he ate the cake)

MENTAL-PROCESS (he thought...)

VERBAL-PROCESS (he said...)

RELATIONAL-PROCESS (he is old; carrots grow faster than beans)

ONE-PLACE-RELATION (he exists)

TWO-PLACE-RELATIONS (he is uglier than she is)

THREE-PLACE-RELATION (he bought apples from her)

QUALITY (redness, happiness...)

Other Ontologies and term taxonomies currently work ed are CYC, WordNet, and ISI’s Omega (omega.isi.edu).

5. Semantics: Two Aspects of Representing Knowledge

The content and organization of knowledge plays a central role in language processing. Two classes of issues studied: representation *content* (symbol systems, ontologies) and representation *systems and reasoning* (inference frameworks and axioms). The AI subfield of Knowledge Representation (KR) has worked with semantics, focusing much attention on systems and reasoning.

In both aspects there are widely different approaches. One can represent them as follows:

Content					Systems			
parsimonious		profligate			absolute		rigorous	
primitives	↔	plenty			anarchy	↔	rules	
CD	Dahlgren	Sowa	Hobbs		FRL	KEE	KL-ONE, Loom	

5.1 Content

There are two principal styles: *Parsimonious* (choosing just a few primitive concepts and composing them into many meanings) and *Profligate* (choosing many different concepts, each represented independently). Both approaches have strengths and weaknesses.

The Parsimonious Approach

Represent a set of example sentences, first in Conceptual Dependency (CD) (Schank and Abelson, 77) and then using open-ended standard representation predicates:

Mary gives John a book

Mary gave John the book

Mary sells John a book

John buys a book from Mary

Mary read John a story

Mary gave John a kiss

Mary gave John a black eye

Mary saw a plane flying to New York

Mary saw the Grand Canyon flying to New York

John got a book

John got better

John was healthy

to illustrate PTRANS, MTRANS, STATEs, STATE-CHANGEs, ENABLE, RESULT, etc. The idea is to try to represent the *meanings* of these sentences, by capturing the *meanings* of their words and composing them appropriately. Notice that you need for this composition both the meaning symbols and the relations that link them.

Some typical **case relations for processes**:

agent, patient (= object), instrument, beneficiary,

location, source, destination,

manner,

cause/result, temporal-sequence, enable...

and **for objects**:

name, size, age, location, nationality, color, weight...

and **for states**:

experiencer, state, degree

The Parsimonious approach is nice, because your system can know that “buy” and “sell”, or “come” and “go”, are more or less the same thing. But you pay a price: you have to decompose complex concepts into these primitives! How do “Democracy” or “Love” decompose? You generally end up unable to build a real-world domain model.

The Profligate Approach

On the other hand, the Promiscuous (profligate) approach is nice too, because your work in representing is much easier. But you pay a price here too: your system cannot know the relatedness of closely related concepts: “come” and “go”, or “price” and “buy” and “buyer”. Also you have no guidance on what primitives to create, or why. If you build a domain model of more than 1500 concepts you generally end up with a bit of a mess, and have trouble enforcing consistency.

To help with consistency and concept organization, people often build **Ontologies**—taxonomies of the symbols, arranged in increasing specificity. This is the subject of a whole course, and is a fascinating mixture of philosophy, linguistics, and psychology. Near the top level you need to differentiate (at least): Actions/Events, States, State-Changes, Objects, Relations, Qualities. And it goes on from there.

A problem in the Profligate approach is to determine how many concepts there are — that is, how to ‘chop up’ the continuum of meaning fields into discrete concepts. The recently developed OntoNotes procedure that uses word senses as a ‘bridge’ is instructive. Exercise: how many word senses do the following sentences contain? How are the senses taxonomized, from least to most similar?

1. *Drive the demons out of her and teach her to stay away from my husband!!*

2. *Shortly before nine I drove my jalopy to the street facing the Lake and parked the car in shadows.*
3. *He drove carefully in the direction of the brief tour they had taken earlier.*
4. *Her scream split up the silence of the car, accompanied by the rattling of the freight, and then Cappy came off the floor, his legs driving him hard.*
5. *With an untrained local labor pool, many experts believe, that policy could drive businesses from the city.*
6. *Treasury Undersecretary David Mulford defended the Treasury's efforts this fall to drive down the value of the dollar.*
7. *Even today range riders will come upon mummified bodies of men who attempted nothing more difficult than a twenty-mile hike and slowly lost direction, were tortured by the heat, driven mad by the constant and unfulfilled promise of the landscape, and who finally died.*
8. *Cows were kept in backyard barns, and boys were hired to drive them to and from the pasture on the edge of town.*
9. *He had to drive the hammer really hard to get the nail into that plank!*
10. *She learned to drive a bulldozer from her uncle, who was a road maker.*
11. *I used to drive a taxi (for work) before I went to night school.*
12. *Beware—Ralph drives a hard bargain; you will probably lose all your money.*

5.2 KR Formalisms and Representation Mechanisms

What functionalities do you want your KR system to have?

- Representation of information
Associate bits of info together; build frames/assertion packets/etc.
- Consistency maintenance
Do you want this? Automatic checking that all is well. Humans aren't consistent; why should a system be? The problems of inconsistency
- Inheritance
Why do you want this?—for saving space against redundancy. Tweety the Bird and Oliver Ostrich. Strong vs. weak property inheritance. Classification only possible in strong inheritance
- Defaults
The Nixon diamond. What to do?— disallow/allow anything/ask user
- Inference
Rules defined in T-Box hold on A-Box entities; automatic inference (classification an example). Forward and backward-chaining

As discussed above, there's no "best" solution.

The range here goes from simple free property-inheritance schemes like KRL (Bobrow & Winograd 77) or FRL (Roberts & Goldstein 77) to highly structured ones like KL-ONE (Brachman ~67) and Loom (MacGregor 89). Examples in FRL:

```
E1 := (make-new-frame
      :isa EAT
      :actor JOHN1
      :object (make-new-frame :isa EGG :number 2 :prep SCAMBLED)
      :time Wednesday15))
```

Here inheritance fills in what is not given (:loc = ?) and what is given simply overrides the ISA's value. Freedom but no structure or enforcement of coherence. In contrast, KL-ONE family enforces agreement to role-filler constraints, which makes the system much harder to deal represent things in but guarantees consistency and enables classification. Instantial and conceptual knowledge. A Loom example:

```
(defconcept Phys-Obj
  :constraints (and (:exactly 1 weight) (:exactly 1 location)))

(defconcept Fragile-Thing :is (:and Phys-Obj :primitive))

(tellm Fragile-Thing FT1)
```

Redundancy in doing all this. Property-inheritance. Frameworks in which to represent information. Terminological and Instantial knowledge bases. KL-ONE, NIKL, Loom, and Brachman vs FRL and KEE: strong and weak inheritance. Defaults and overruling, and strong property inheritance requirements and the classifier.

Tying the two KR parts together: an example (the Upper Model).

Predicate calculus. formula := (pred -args-); arg := const | var | fn applic (which is evaled); connective := and | or | not | if | iff | forall | exists; preds; funcs. Denotations: terms → individuals or classes; formulas → ? Not T/F because depends on interp of denotation. Hard question. Soudness (cannot prove non-T thing) and consistency (all statements eventually T).

Tweety is a bird.	(bird Tw)
Tweety eats all good foods.	($\forall X$) ((food X) and (good X) \Rightarrow (eat Tw X))
Sunflower seeds are good food.	(food SS) and (good SS)
\Rightarrow Some bird eats sunflower seeds.	($\exists y$) (eat Y SS) and (bird Y) via (eat Tw SS)

Problems: Compositionality. Set-theoretic Model. Skolemization. Hypothetical, actual, and assumed status. Time. Quantification. Logical operators. Logics: first-order, higher-order, modal, etc.

Now for some fun.

Represent these sentences:

- John goes to Boston
- John went to Boston
- John owns a book
- John has a book
- Mary gave John a book
- Mary bought a book for John.
- Mary traded John money for a book.
- Mary gave John a kiss.
- John had a headache.
- Mary gave John a headache.
- Mary gave John a black eye.

- The book is red.

- Joe had a hamburger. He paid for it.
- Mark and Karen will come and have lunch with us on their way home.

- The children couldn't wait to open their presents!
- She is getting panicky because she may not finish the project in time.
- Mary slept deeply.
- Pete kicked the bucket.
- Why did the chicken cross the road?

An example of a case frame is:

```
(X15
 :type EAT
 :actor (X16
  :type MAN
  :size LARGE
  :identifiable T)
 :object (X17
  :type PEACH
  :size SMALL
  :color RED
```

:identifiable T)
:time PAST
:speechact ASSERTION
:polarity POSITIVE)