Language and Ontology
As Abstractions from Language

John F. Sowa

VivoMind Intelligence, Inc.

29 September 2007

Main Points in This Talk

1. Logic and ontology are abstractions from language.

2. Language develops, both in the species and in the individual, as an accompaniment to social activity.

3. Pragmatics is the primary purpose of language, and the first aspect to be developed in the infant and in the species.

4. Semantics develops as a mapping of words and sentences to reality, but only after distinct words and sentences appear in language.

5. Complex syntax is a late development that makes communication more precise and efficient.

6. Different aspects of reality are distinguished in using language for different purposes in different activities (Wittgenstein's *Sprachspiele*).

7. Formalization is a method of analyzing linguistic structures, defining them precisely, and using the results in precise computation.

8. Analogical reasoning is a more general version of logical reasoning, which does not require a prior step of formalization.

Three Views of Formal Semantics

Richard Montague (1970):

I reject the contention that an important theoretical difference exists between formal and natural languages.

Hans Kamp (2001):

Natural language semantics increasingly takes on the complexion of a branch of a general theory of information representation and transformation. The role of logical inference in the processes of linguistic interpretation indicates an interleaving of inferential and other representation-manipulating operations. This suggests that the inferential relations and operations that have often been considered the essence of logic are better seen as an integral part of a wider repertoire. Thus logic comes to look much more like a general theory of information, than as a discipline concerned more or less exclusively with deduction.

Barbara Partee (2005):

The present formalizations of model-theoretic semantics are undoubtedly still rather primitive compared to what is needed to capture many important semantic properties of natural languages... There are other approaches to semantics that are concerned with other aspects of natural language, perhaps even cognitively “deeper” in some sense, but which we presently lack the tools to adequately formalize.

Three Views of Formalization

Paul Halmos, mathematician:

Mathematics — this may surprise or shock some — is never deductive in its creation. The mathematician at work makes vague guesses, visualizes broad generalizations, and jumps to unwarranted conclusions. He arranges and rearranges his ideas, and becomes convinced of their truth long before he can write down a logical proof... the deductive stage, writing the results down, and writing its rigorous proof are relatively trivial once the real insight arrives; it is more the draftsman's work not the architect's.

Richard Feynman, physicist:

We have a habit in writing articles published in scientific journals to make the work as finished as possible, to cover up all the tracks, to not worry about the blind alleys or describe how you had the wrong idea first, and so on. So there isn't any place to publish, in a dignified manner, what you actually did in order to get to do the work.

Gian-Carlo Rota, mathematician:

Shocking as it may be to a conservative logician, the day will come when currently vague concepts such as motivation and purpose will be made formal and accepted as constituents of a revamped logic, where they will at last be alloted the equal status they deserve, side-by-side with axioms and theorems.

Origin of Language

Formal systems begin with syntax, define semantics on top of syntax, and define pragmatics on top of semantics.

But infants begin to use language for some purpose from their very first words and their very loosely structured two-word phrases.

Early hominids, perhaps Australopithecus 4 million years ago, must have developed some protolanguage for some purpose -- perhaps in hunting parties, as in the calls and gestures of wolves.

In *The Symbolic Species*, Deacon proposed that the first step to a true language was the transition from indexicals to symbols.

As an example, vervet monkeys have three kinds of warning calls:

- **One for eagles, another for snakes, and a third for leopards.**
- **Some people claim that vervets have a primitive form of symbols.**
- **But Deacon claimed that their calls could be interpreted as the equivalent of a pointing finger: *look up*, *look down*, *look around*.**
- **Unlike an index, a symbol can be used to refer to something that is not present in the context.**

Language as Vocal Gestures

In language development, both in the species and in each individual, the fundamental purpose is to support and enhance social interactions.

People typically gesticulate while speaking, even when talking on the telephone.

Chimpanzees and human infants less than a year old can learn and use gestures to communicate.

In any social activity, the characteristic motions can be used as iconic gestures, and pointing gestures can be used as indexicals.

Chimps and infants can learn to use such gestures to indicate their desires, even when the objects of those desires are absent.

This development follows Peirce's claim that “symbols grow” from icons and indexes – and from symbols with simpler meanings.

It is consistent with Wittgenstein's *language games* (Sprachspiele), which are the social activities that determine the pragmatics directly and the semantics indirectly.

Syntax evolves from the temporal ordering of gestures or calls in the activities.

Origin of Formal Logic and Ontology

The ancient Greeks had as many lawsuits as modern Americans.

But the plaintiffs and defendants had to plead their own cases in court.

The Sophists earned a good living by teaching people how to present their cases in the best possible light.

Plato criticized the Sophists for ignoring truth and trying to make the weaker case seem to be the stronger.

Aristotle analyzed and classified the arguments:

- **Deductive methods preserve truth.**
- **Unsound methods, called fallacies, are plausible, but misleading.**
- **Syllogisms are deductive methods for reasoning about the categories of an ontology.**
- **Original meaning of κατεγορια: an accusation in a court of law.**

Formalization of Mathematics

Euclid adopted Aristotle's approach and extended his techniques to cover the elements of mathematics, especially geometry.

Euclid's definitions follow Aristotle's method of specifying the *genus* or general type and the *differentiae* that distinguish the species from other species of the same genus.

Aristotle also had the first recorded use of letters as variables to label the terms of his syllogisms.

It's not clear whether Euclid adopted the use of letters from Aristotle, or whether Aristotle adopted them from earlier use in geometry.

The methods of Aristotle and Euclid remained the standards of precision for deductive reasoning until the 19th century.

But Aristotle also recognized the prelogical need for analyzing data in order to derive the general principles used in deduction.

Fundamental Problem

* Deduction is the method of reasoning at the foundation of current database and knowledge-based systems.

* Deduction is precise, predictable, and brittle.

* If everything is perfect, deduction is perfect.

* If anything in the system is imperfect, deduction can magnify and propagate the imperfection to the point of a total collapse.

* When multiple systems interoperate, the likelihood and the danger of imperfection escalates.

Prospects for a Universal Ontology

Attempts to create a universal classification of all concepts:

* 4th century BC: Aristotle's categories and syllogisms.

* 17th century: Universal language schemes by Descartes, Mersenne, Pascal, Leibniz, Newton, Wilkins, and others.

* 18th century: More schemes, the Grand Academy of Lagado, Kant's categories.

* 19th century: Roget's Thesaurus, Oxford English Dictionary.

* Early 20th century: Many terminologies in many different fields.

* 1960s: Computerized versions of the terminologies.

* 1970s: ANSI/SPARC Conceptual Schema.

* 1980s: Cyc, WordNet.

* 1990s: SRKB, ISO Conceptual Schema, Semantic Web, many workshops.

* 2000s: Many proposals, no consensus.

Informal terminologies and dictionaries have been extremely successful.

Formal systems are still research projects.

Methods of Reasoning

Three methods of formal logic:

1. **Deduction. Apply a general principle to infer some fact.**

```
 Given:  Every bird flies.  Tweety is a bird.
 Infer:  Tweety flies.
```

2. **Induction. Assume a general principle that subsumes many facts.**

```
 Given:  Tweety, Polly, and Hooty are birds.  Fred is bat.
         Tweety, Polly, and Hooty fly.  Fred flies.
 Assume: Every bird flies.
```

3. **Abduction. Guess a new hypothesis that explains some fact.**

```
 Given:  Every bird flies.  Tweety flies.
 Guess:  Tweety is a bird.
```

According to Peirce (1902), "*Besides these three types of reasoning there is a fourth, analogy, which combines the characters of the three, yet cannot be adequately represented as composite.*"

Four Views of Analogical Reasoning

1. **By logicians:**

   Deduction is reasoning from “first principles.”

   Analogy is an unsound, but interesting heuristic.

2. **By psychologists:**

   Analogy is a fundamental cognitive mechanism.

   Language and reasoning depend heavily on analogy.

3. **Theoretical:**

   Analogy is a very general pattern-matching process.

   Deduction, induction, and abduction depend on disciplined uses of analogy.

4. **Computational:**

   A powerful and flexible technique for reasoning, learning, and language processing.

   But practicality depends on finding analogies efficiently.

Society of Mind

Proposal by Marvin Minsky, based on neural hypotheses.

The idea is good, but a semiotic basis is more fundamental: every neuron is itself a complex semiotic system.

Implemented in the Flexible Modular Framework™.

* Multiple interacting agents.

* Communication by message passing.

* Heterogeneous knowledge representations.

* Heterogeneous ontologies.

* Heterogeneous reasoning methods.

Communication requires two agents to agree at some local level of messages, not at a global level of all possible messages.

Example of Analogy: How is a cat like a car?

Analogy of Cat to Car	
Cat	Car
head	hood
eye	headlight
cornea	glass plate
mouth	fuel cap
stomach	fuel tank
bowel	combustion chamber
anus	exhaust pipe
skeleton	chassis
heart	engine
paw	wheel
fur	paint

Data from WordNet and other sources are translated to conceptual graphs.

The VivoMind Analogy Engine (VAE) starts at the nodes for Cat and Car and tries to find the longest matching paths through the graphs.

All analogies found are ranked by a semantic-distance measure.
The above analogy received the highest score for that pair of words

Operations Performed During Pattern Matching

Following paths from each starting node:

[Cat]→(HasPart)→[Head]→(HasPart)→[Eye]→(HasPart)→[Cornea]

[Car]→(HasPart)→[Hood]→(HasPart)→[Headlight]→(HasPart)→[GlassPlate]

Matching concept nodes with similar types, properties, and relations:

- **Head and hood are in the front.**
- **Eyes and headlights are related to light.**
- **Cornea and glass plate are transparent.**
- **Paws and wheels support the body, and there are four of each.**

Approximate matching (ignoring the nodes in red):

[Cat]→(HasPart)→[Mouth]→(Flow)→[Esophagus]→(Flow)→[Stomach]→
→(Flow)→[Bowel]→(Flow)→[Anus]

[Car]→(HasPart)→[FuelCap]→(Flow)→[FuelTank]→
→(Flow)→[CombustionChamber]→(Flow)→[Muffler]→(Flow)→[ExhaustPipe]

Note: The source data did not contain information about the pipe from the fuel cap to the fuel tank. That missing part would have matched the esophagus of the cat.

Three Methods Used by VAE

1. Matching concept types, in order of increasing semantic distance:
   - Identical types.
   - Subtype - supertype.
   - Siblings of same supertype.
   - More distant cousins.

2. Matching subgraphs:
   - Match isomorphic subgraphs independent of type labels.
   - Merge adjacent nodes to make them isomorphic.

3. Finding metalevel mappings that can relate subgraphs,
   — even though they are not isomorphic.

Methods #1 and #2 are used for the Cat-Car example.

Method #3 (combined with #1 and #2) is used for aligning ontologies.

Need to Align Ontologies

Any topic can be described at many levels of detail with different choices of labels for the concept and relation types.

There is no standard upper-level ontology, but there are many important ontologies that must be accommodated:

- Some widely used ontologies, such as Cyc, OpenCyc, SUMO, Dolce, BFO, etc.

- Many ontologies required by governments and large organizations.

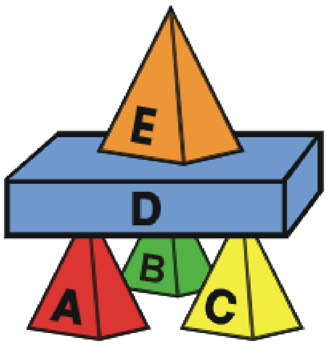
- Example: The Amazon.com ontology.

- An enormous number of legacy systems with no explicit ontologies.

Critical requirement: Enable heterogeneous systems to interoperate despite differences in their underlying ontologies, whether implicit or explicit.

Example of Different Ontologies

- The structure on the right may be described in different ways:



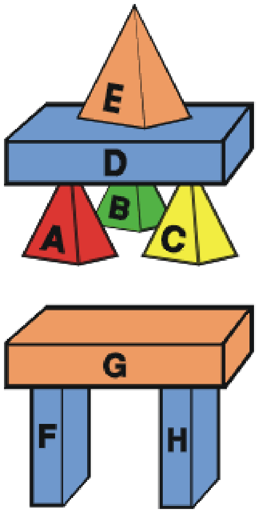
- English sentence: "A red pyramid A, a green pyramid B, and a yellow pyramid C support a blue block D, which supports an orange pyramid E."

- A relational database would use tables.

- But there are many different options for organizing the tables and choosing labels for the tables and columns.

- These choices lead to different ontologies, which may be structurally very different from one another.

Representation in a Relational DB



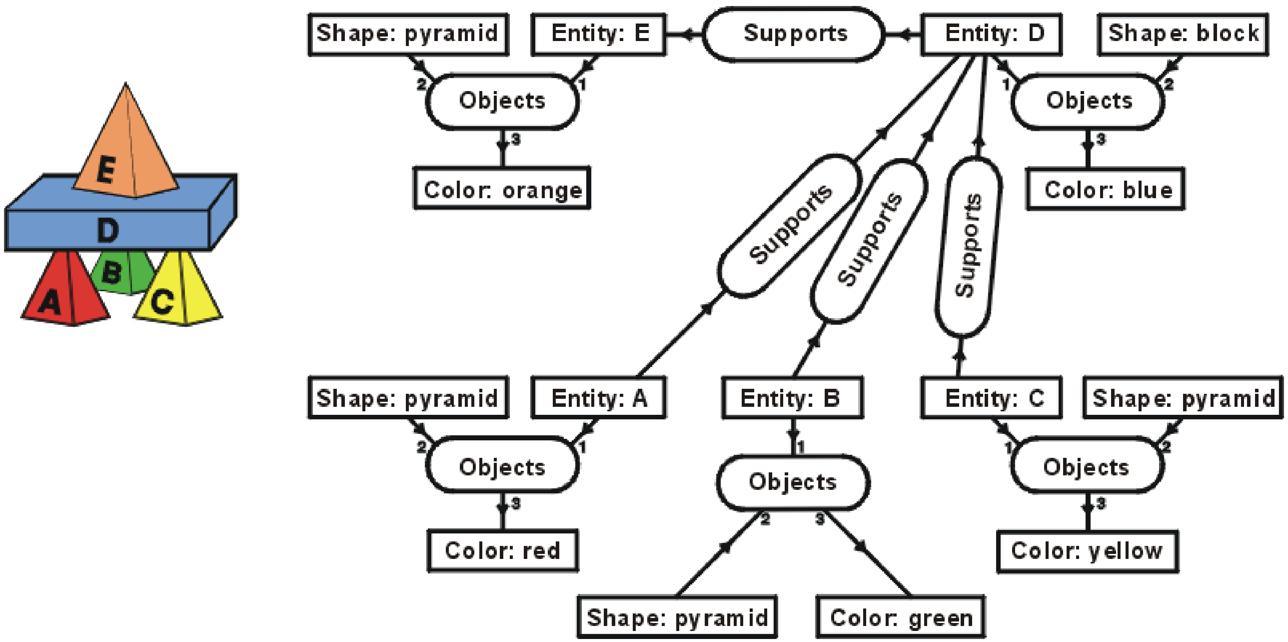
Objects

ID	Shape	Color
A	pyramid	red
B	pyramid	green
C	pyramid	yellow
D	block	blue
E	pyramid	orange
F	block	blue
G	block	orange
H	block	blue

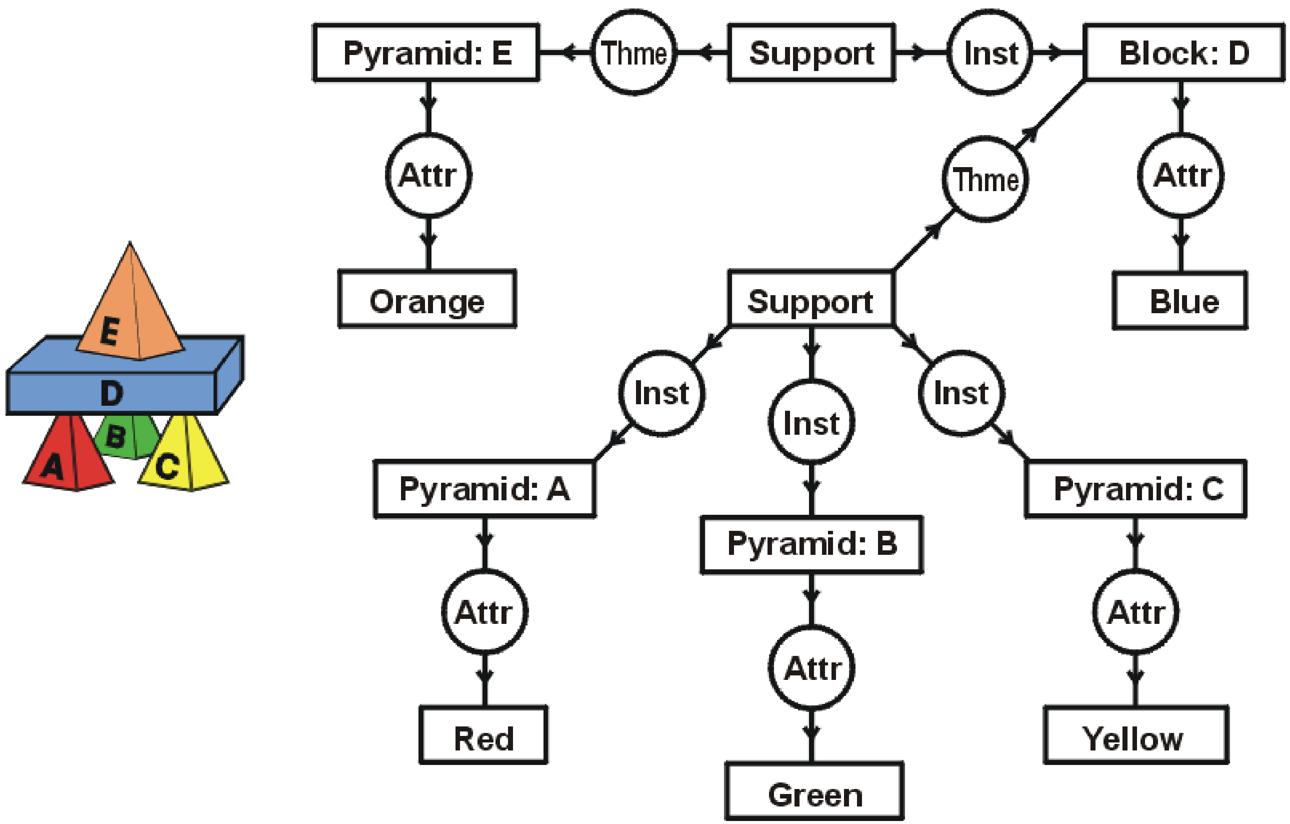
Supports

Supporter	Supportee
A	D
B	D
C	D
D	E
F	G
H	G

Conceptual Graph from Relational DB



Conceptual Graph from English Sentence

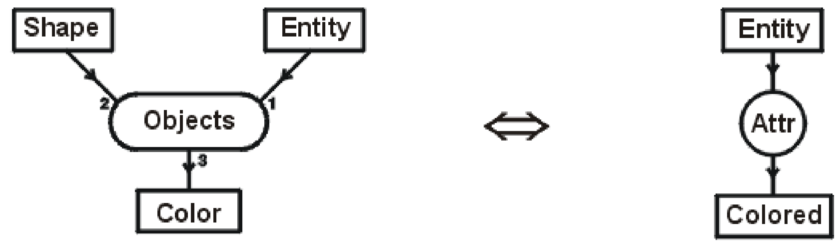


“A red pyramid A, a green pyramid B, and a yellow pyramid C support a blue block D, which supports an orange pyramid E.”

The Two CGs Look Very Different

* CG from RDB has 15 concept nodes and 8 relation nodes.
* CG from English has 12 concept nodes and 11 relation nodes.
* All the type labels of concept and relation nodes are different.
* But there are some structural similarities.
* VAE uses method #3 to find them.

Transformations Found by VAE



Top transformation was applied to 5 subgraphs.

Bottom one was applied to 4 subgraphs.

One application of any given transformation could be due to chance, but 4 or 5 applications are strong evidence for its significance.

The fact that these two transformations completely map each graph to the other is convincing.

Computational Complexity

Research by Falkenhainer, Forbus, & Gentner:

Problem: In a knowledge base of N graphs, the time to find the best analogy to a graph G takes time proportional to N^3.

If N is 10, N^3 is a thousand.
But if N is a billion, N^3 is 1,000,000,000,000,000,000,000,000,000.

MAC/FAC Solution:

Use a search method to narrow down the likely candidates from some large number N to a much, much, much smaller number n.

With a good search method:

* **The execution time for the search should be proportional to log(N).**
* **It should find some small number of graphs, n.**
* **And the most relevant graphs should be among those n.**

Question: What kinds of search algorithms have these properties?

Semantic Distance Measure

Graph encodings that map discrete structures of nodes and arcs to a continuous metric space.

Many different encodings have been developed over the years.

The most advanced work has been done in organic chemistry.

Those encodings take into account the graph structure plus the labels on the nodes (e.g., chemical elements and bonds).

Many variations have been developed and tested over the years, but they all have the ability to find similar graphs in logarithmic time.

Such techniques have been applied to conceptual graphs.

The results can reduce the search space for the best possible match from some large number N to a very small number n.

Criticisms of Logical Deduction

Deduction in mathematics can be precise, but deduction about any empirical subject must depend on prior induction, which is almost always incomplete.

Criticism by the physician, Sextus Empiricus (2nd century AD):

> ***Every human is an animal.***
> ***Socrates is human.***
> ***Therefore, Socrates is an animal.***

If the major premise was derived by checking every human,
then Socrates was considered, and the argument is circular.

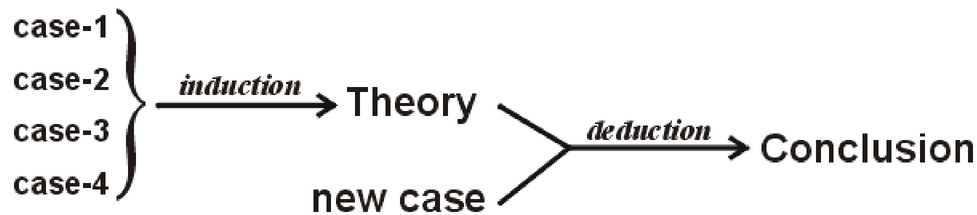
Otherwise, the induction was incomplete, and the conclusion is uncertain.

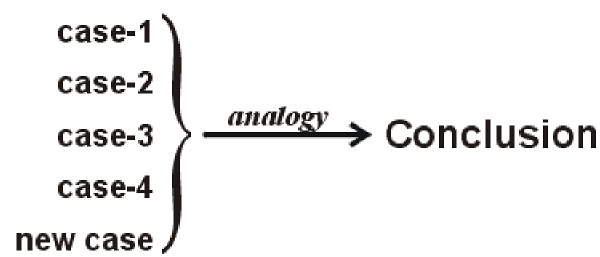
Criticism by the legal scholar, Ibn Taymiyya (14th century AD):

> **Every empirical theory is derived by induction from cases.**
>
> **Any deduction from such a theory can be obtained by applying analogical reasoning to exactly the same cases.**

Case-Based Reasoning

Ibn Taymiyya's comparison of logical and analogical reasoning:





If the same cases are used, analogy can give the same answer in one step that requires two steps by induction followed by deduction.

What Ibn Taymiyya did not recognize: If the same theory could be reused for many different applications, deduction would be more efficient.

Logic as a Disciplined Use of Analogies

The structure-mapping operations of analogy are used in every kind of logical reasoning:

* Deduction. Every step requires a unification, which is special case of the structure mappings used in analogies.

* Induction. Analogies are used to find common generalizations of multiple instances.

* Abduction. The operation of guessing or forming an initial hypothesis, called abduction, requires analogies to find likely causes or explanations.

In both human reasoning and computer implementations, the same underlying operations can support both logical and analogical reasoning.

An Application of Case-Based Reasoning

A textbook publisher needs to evaluate student answers to math questions.

*** Free-form answers in English sentences.**

*** Much harder to evaluate than multiple choice.**

Typical question:

The following numbers are 1 more than a square: 10, 37, 65, 82.

If you are given an integer N that is less than 200,
how would you determine whether N is 1 more than a square?

Explain your method in three or four sentences.

How could a computer system evaluate student answers?

Determine whether they are correct, incorrect, or partially correct?

And make helpful suggestions about the incorrect answers?

Publisher's Current Procedure

To evaluate new exam questions, the publisher normally gives the exam to a large number of students.

For each question, they get about 50 different answers:

* Some are completely correct
  — but stated in different ways.

* Some are partially correct
  — and the teacher says what is missing.

* Others are wrong
  — in many different ways.

Result: 50 pairs of student answer and teacher response.

Each pair of (answer,response) is a case for case-based reasoning.

Intellitex Parser

VivoMind has developed a parser named Intellitex:

* Uses a rather simple grammar.

* Depends on analogies for interpreting sentences.

* Generates conceptual graphs as output.

* Robust: always generates some CG as its best guess.

These properties are important for handling typical student answers, which frequently have poor grammar and incomplete sentences.

Minor errors are not necessarily bad — provided that Intellitex makes the same errors consistently in all cases.

Using VAE to Evaluate Student Answers

Use VAE to compare each new answer to the 50 cases:

1. For all 50 cases, translate student answer to conceptual graphs.
2. Translate each new answer to a new CG.
3. Compare the new CG to the 50 CGs for previous answers.
4. Use the measure of semantic distance to determine the best match.
5. If there is a good match, print out the corresponding response.
6. Otherwise, send the new answer to a teacher to evaluate.

Result:

VAE found a good match for most of the student answers.

For each good match, the previous teacher's response was appropriate.

When VAE failed to find a good match, the new case could be
added to the list of cases in order to improve its coverage.

There was no need for the teachers to write rules or programs.

Legacy Re-engineering

Version of Intellitex applied to three languages — English, COBOL, and JCL:

* 1.5 million lines of COBOL.

* Several hundred JCL scripts.

* 100 megabytes of English documentation — reports, manuals, e-mails, Lotus Notes, HTML, and transcriptions of oral communications.

Goal:

* Analyze the COBOL and JCL to determine:

  Data dictionary, data flow diagrams, process architecture diagrams, system context diagrams.

* Analyze the English documentation to determine:

  Discrepancies between the documentation and the implementation.

  English glossary of all terms and their changes over the years.

  English captions on the diagrams derived from COBOL and JCL.

Sample Documentation

Mixture of English words, computer jargon, and names of programs, files, and COBOL variables:

```
The input file that is used to create this piece of the Billing Interface
for the General Ledger is an extract from the 61 byte file that is
created by the COBOL program BILLCRUA in the Billing History production
run.  This file is used instead of the history file for time efficiency.
This file contains the billing transaction codes (types of records) that
are to be interfaced to General Ledger for the given month.  For this
process the following transaction codes are used:  32 - loss on unbilled,
72 - gain on uncollected, and 85 - loss on uncollected.  Any of these
records that are actually taxes are bypassed.  Only client types
01 - Mar, 05 - Internal Non/Billable, 06 - Internal Billable, and
08 - BAS are selected.  This is determined by a GETBDATA call to the
client file.  The unit that the gain or loss is assigned to is supplied
at the time of its creation in EBT.
```

Note nonstandard syntax: `32 - loss on unbilled, 06 - Internal Billable.`

An Important Simplification

Extremely difficult problem:

* Trying to understand English specifications.

* Mapping the results to an executable COBOL program.

Much, much easier problem:

* Mapping COBOL to conceptual graphs.

* Using CGs derived from COBOL to analyze English documentation.

* Using VAE to find similarities and differences in the CGs.

Results

Job finished in 8 weeks by two programmers, Arun Majumdar and André LeClerc.

*** Four weeks for customization:**

Design and logistics.

Additional programming for I/O formats.

*** Three weeks to run Intellitex + VAE + extensions:**

24 hours a day on a 750 MHz Pentium III.

VAE handled matches with strong evidence (close semantic distance).

Matches with weak evidence were confirmed or corrected by Majumdar and LeClerc.

*** One week to produce a CD-ROM with integrated views of the results:**

Glossary, data dictionary, data flow diagrams, process architecture, system context diagrams.

Contradiction Found by VAE

From analyzing English documentation:

* Every employee is a human being.

* No human being is a computer.

From analyzing COBOL programs:

* Some employees are computers.

What is the reason for this contradiction?

Quick Patch in 1979

A COBOL programmer made a quick patch:

* Two computers were used to assist human consultants.

* But there was no provision to bill for computer time.

* Therefore, the programmer named the computers Bob and Sally, and assigned them employee ids.

For more than 20 years:

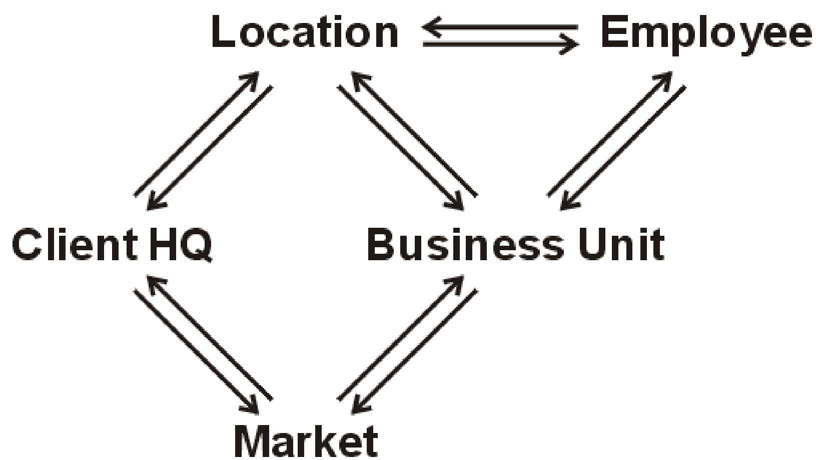
* Bob and Sally were issued payroll checks.

* But they never cashed them.

VAE discovered the two computer “employees”.

Mismatch Found by VAE

A diagram of relationships among data types in the database:



Question: Which location determines the market?

According to documentation: Business unit.

According to COBOL programs: Client HQ.

Management had been making decisions based on incorrect assumptions.

Conclusions

No evidence of formal logic as a prerequisite for learning, understanding, or speaking a natural language.

Common logical operators — *and*, *or*, *not*, *if-then*, *some*, *every* — are present in every NL. But they are used in many different senses, which include classical first-order logic as an important special case.

Analog reasoning is fundamental. Induction, deduction, and abduction are important, highly disciplined special cases.

But analogy is a more general reasoning method, which can be used even with images, prior to any version of language.

No evidence of a highly axiomatized ontology for any natural language.

But many important commonalities result from common human nature, experience, and activities (see Talmy, Wierzbicka, etc.).

Formal, logic-based systems with deeply axiomatized ontologies have been fragile and limited in their coverage of natural language texts.

Analogy-based systems with loosely defined terminologies can be far more robust and efficient for many applications.

Related Publications

Analogical Reasoning:
http://www.jfsowa.com/pubs/analog.htm

Architectures for Intelligent Systems:
http://www.jfsowa.com/pubs/arch.htm

Graphics and Languages for the Flexible Modular Framework:
http://www.jfsowa.com/pubs/gal4fmf