Knowledge Graphs
Past, Present, and Future

John F. Sowa

11 September 2019
Ontology Summit

Early Days of Artificial Intelligence

1960: **Hao Wang's theorem prover took 7 minutes to prove all 378 FOL theorems of *Principia Mathematica* on an IBM 704 – much faster than two brilliant logicians, Whitehead and Russell.**

1960: **Emile Delavenay, in a book on machine translation:**

"While a great deal remains to be done, it can be stated without hesitation that the essential has already been accomplished."

1965: **Irving John Good, in speculations on the future of AI:**

"It is more probable than not that, within the twentieth century, an ultraintelligent machine will be built and that it will be the last invention that man need make."

1968: **Marvin Minsky, technical adviser for the movie *2001*:**

"The HAL 9000 is a *conservative estimate* of the level of artificial intelligence in 2001."

Bird Nest Problem

Robots can perform many tasks with great precision.

But they don't have the flexibility to handle unexpected shapes,

They can't wash dishes the way people do — with an open-ended variety of shapes and sizes.

And they can't build a nest in an irregular tree with irregular twigs, straw, and moss.

If a human guides a robot through a complex task with complex material, the robot can repeat the same task in the same way.

But it doesn't have the flexibility of a bird, a beaver, or a human.

The Ultimate Understanding Engine

Sentences uttered by a child named Laura before the age of 3. *

Here's a seat. It must be mine if it's a little one.

I went to the aquarium and saw the fish.

I want this doll because she's big.

When I was a little girl, I could go "geek geek" like that, but now I can go "This is a chair."

Laura used a larger subset of logic than Montague formalized.

No computer system today can learn and use language as fast, as accurately, and as flexibly as a three-year-old child.

*** John Limber, The genesis of complex sentences.**
http://pubpages.unh.edu/~jel/JLimber/Genesis_complex_sentences.pdf

Knowledge Graphs

Scaling up semantic nets to the size of the WWW.

- Over a century ago, tree and graph notations were designed to represent the semantics of natural languages (NLs).
- In the 1960s, they were implemented in NL processing systems.
- In the 1970s and '80s, they could represent full first-order logic.
- RDF and other notations for the Semantic Web can represent subsets.

But NL understanding (NLU) is still a distant goal.

- Syntactic processing by NLP is acceptable for many purposes.
- Voice recognition is also acceptable for many purposes.
- An approximate semantics is acceptable for some purposes, but it requires a huge amount of development by highly trained people.
- NLP systems cannot explain their answers.
- And users constantly call “AGENT” when communication breaks down.

Problems and Challenges

Early hopes for artificial intelligence have not been realized.

Language understanding is more difficult than anyone thought.

A three-year-old child is better able to learn, understand, and generate language than any current computer system.

Tasks that are easy for many animals are impossible for the latest and greatest robots.

Questions:

- Have we been using the right theories, tools, and techniques?
- Why haven't these tools worked as well as we had hoped?
- What other methods might be more promising?
- What can research in neuroscience and psycholinguistics tell us?
- Can it suggest better ways of designing intelligent systems?

Cyc Project

The most ambitious attempt to build the HAL 9000:

- Cyc project founded by Doug Lenat in 1984.
- Starting goal: Implement the background knowledge of a typical high-school graduate.
- Ultimate goal: Learn new knowledge by reading textbooks.

After the first 25 years,

- 100 million dollars and 1000 person-years of work,
- 600,000 concepts,
- Defined by 5,000,000 axioms,
- Organized in 6,000 microtheories.

Some good applications, but more needs to be done:

- Cyc cannot yet learn by reading a textbook.
- Cyc cannot understand language as well as a child.

What Makes People Intelligent?

Short answer: Flexibility, generality, and adaptability.

- The languages of our stone-age ancestors can be adapted to any subject: science, technology, business, law, finance, and the arts.
- When people invent anything, they find ways to describe it.
- When people in any culture adopt anything from another culture, they borrow or adapt words to describe it in their native language.

Minsky's answer: A society of heterogeneous agents:

"What magical trick makes us intelligent? The trick is that there is no trick. The power of intelligence stems from our vast diversity, not from any single, perfect principle. Our species has evolved many effective although imperfect methods, and each of us individually develops more on our own. Eventually, very few of our actions and decisions come to depend on any single mechanism. Instead, they emerge from conflicts and negotiations among societies of processes that constantly challenge one another." *

* Marvin Minsky (1986) *The Society of Mind*, New York: Simon & Schuster, §30.8.
See also Push Singh & Marvin Minsky (2004) An architecture for cognitive diversity.

Case Study: Cyc and IBM Watson

Why did IBM, not Cyc, beat the Jeopardy! champion?

Short answer: Cyc was not designed for game shows.

- Cyc was designed to represent the general knowledge of a typical high-school student.
- A high-school education isn't sufficient to win at Jeopardy!
- But IBM devoted a large research team to a single problem.

Longer answer: Single paradigm vs. multiple paradigms.

- Cyc is based on a single paradigm: formal logic, deductive reasoning, a very large ontology, and large volumes of data.
- IBM used a team of researchers with a wide range of expertise.
- They started with many independently developed tools and made them interoperate on different aspects of the problem.

Next question: Is it possible to develop Watson-like systems without requiring three dozen PhD researchers?

IBM Watson for Applications

A hybrid with multiple paradigms. *

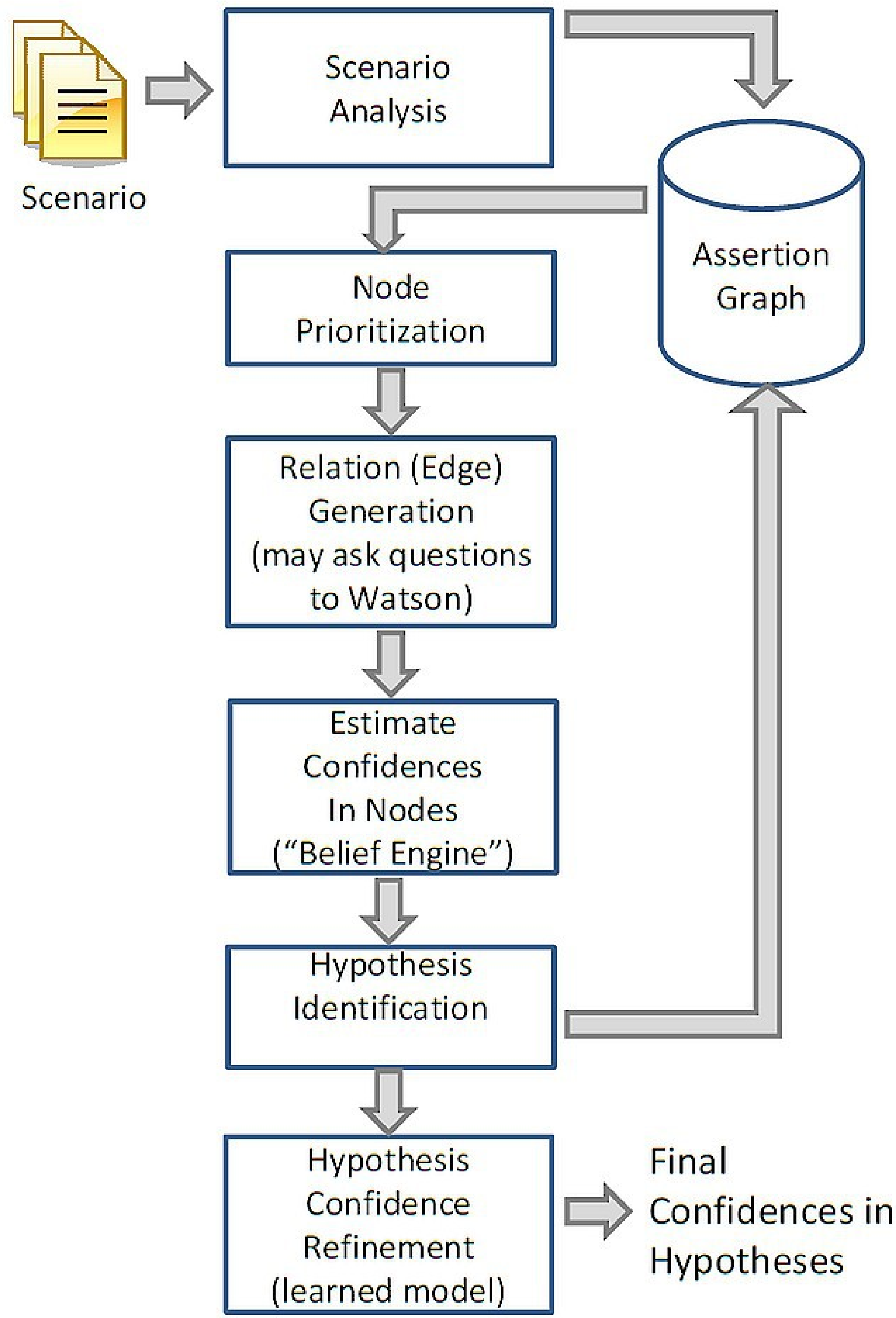
Scenario-based system for reasoning and Q/A about various applications.

The input scenario describes some situation, e.g. a patient's symptoms.

The first step translates the input to an assertion graph.

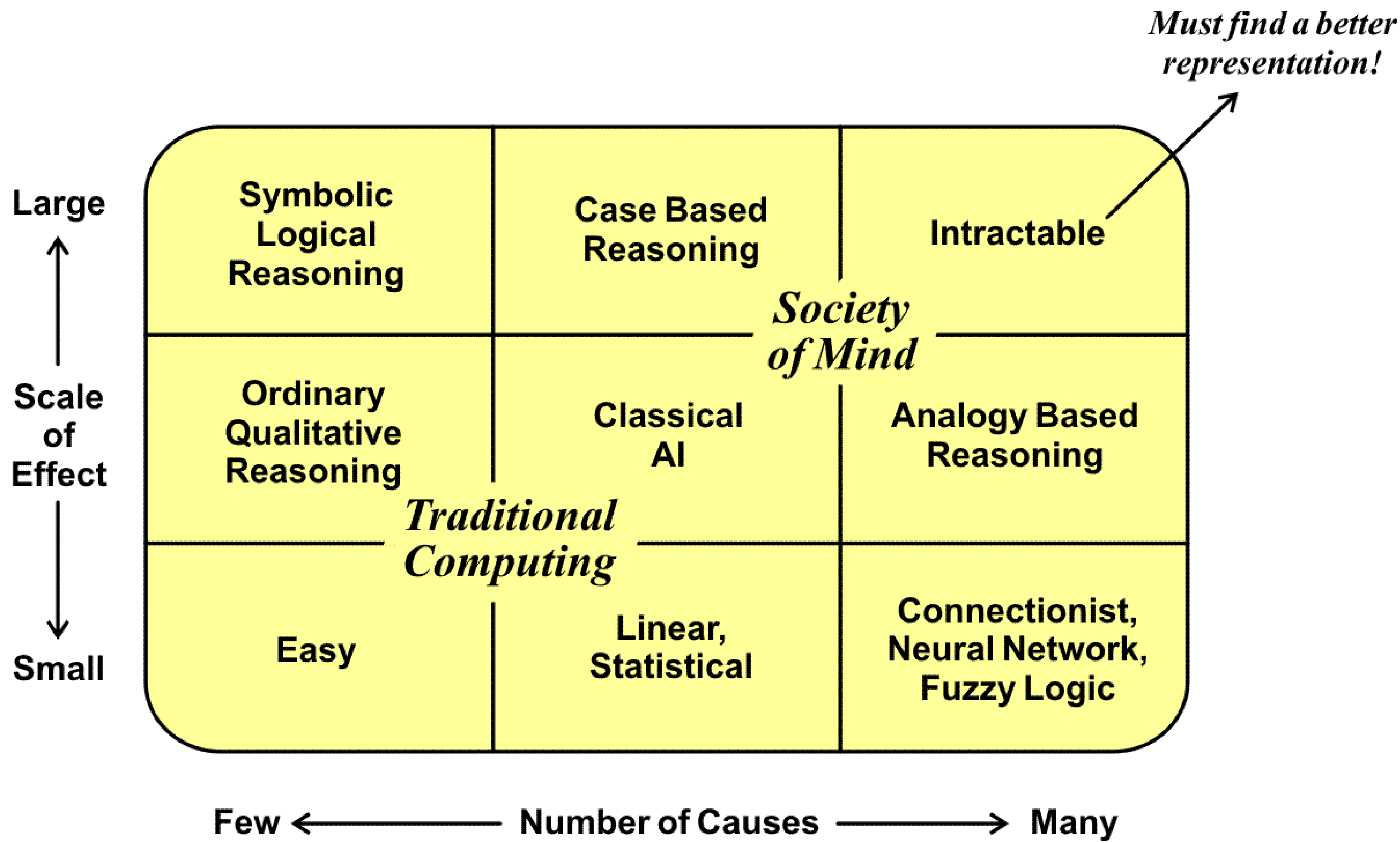
Instead of answering one question, as in Jeopardy!, Watson Paths does extended reasoning to generate a more complex assertion graph.

The reasoning may continue until the assertion graph satisfies some task-dependent criteria.



* See Lally et al. (2014) Watson Paths.

Minsky's Challenge



Adapted from a diagram by Minsky, Singh, & Sloman (2004).

Meeting the Challenge

As Minsky's diagram shows, AI methods cannot process large numbers of causes and complex effects as efficiently as humans.

Statistical methods and neural networks can relate many causes (input variables), but only small-scale effects (simple outputs).

Logic can reason about complex effects (multiple interrelated phenomena), but only with simplified causes (few axioms).

In his *Society of Mind* and *Emotion Engine*, Minsky proposed systems of heterogeneous, interacting modules or agents:

- **Can those agents improve computational efficiency?**
- **Can psycholinguistics and neuroscience guide the design of agents?**
- **What kind of logic, reasoning, and semantics would they support?**
- **Would they use symbolic, statistical, or image-like representations?**
- **Or would they use an open-ended variety of representations?**

Kyndi Technology

Cognitive Memory™ is a key component of Kyndi technology:

- Associative storage and retrieval of graphs in log(N) time.
- Approximate pattern matching for analogies and metaphors.
- Precise pattern matching for logic and mathematics.

Analogies can support informal, case-based reasoning:

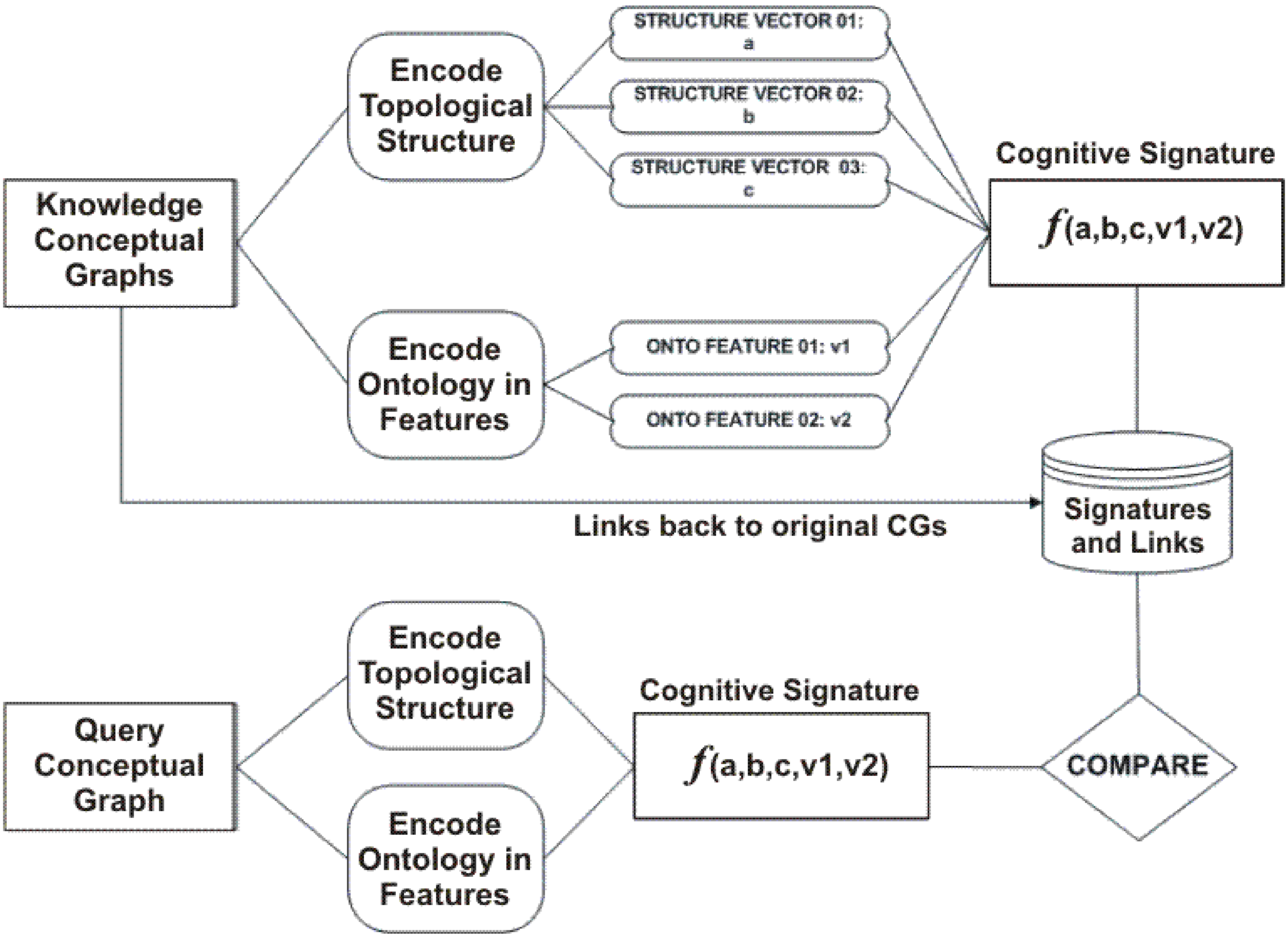
- CM can store large volumes of previous knowledge and experience.
- Any new case can be matched to similar cases in long-term memory.
- Close matches are ranked by a measure of semantic distance.

Formal reasoning is based on a disciplined use of analogy:

- Induction: Generalize multiple cases to create rules or axioms.
- Deduction: Match (unify) a new case with part of some rule or axiom.
- Abduction: Form a hypothesis based on aspects of similar cases.

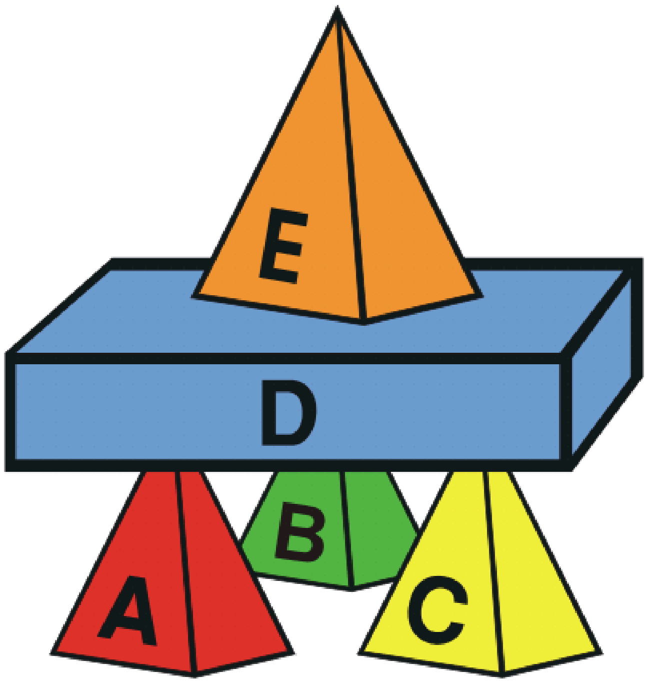
CM can support a wide range of AI and NLP methods: formal or informal, crisp or fuzzy, symbolic or subsymbolic.

Cognitive Memory™



High-speed associative memory for all Kyndi modules

Describing Things in Different Ways



How can we describe what we see?

In ordinary language?

In some version of logic?

In a relational database?

In the Semantic Web?

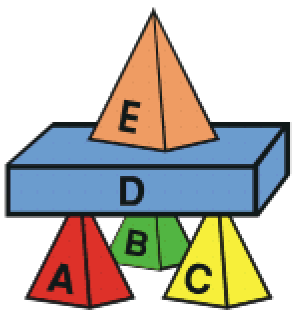
In a programming language?

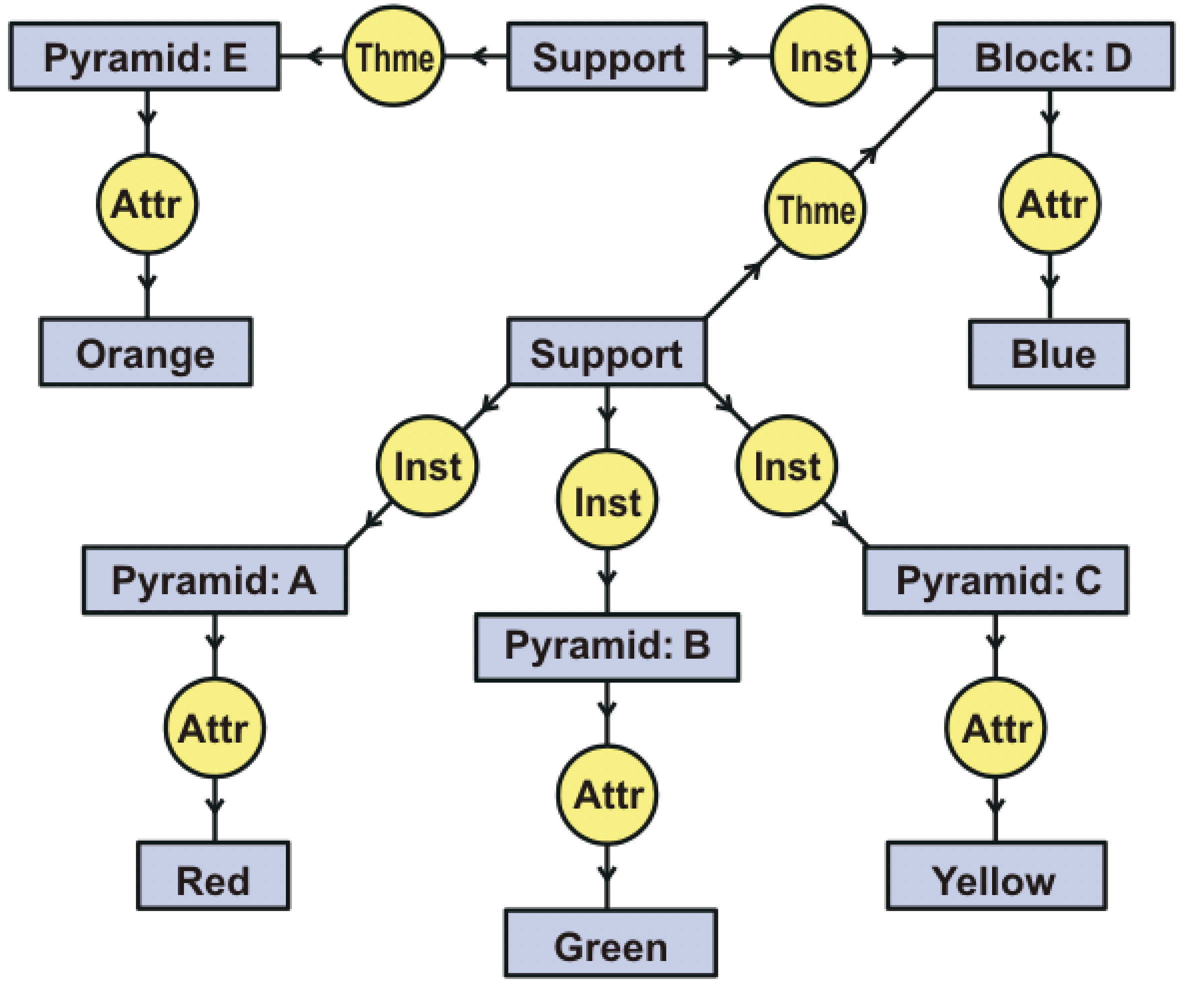
Even when people use the same language, they use different words and expressions.

How could humans or computers relate images and notations, linear or graphic, to one another?

Answer: Find analogies that map one to another.

Mapping English to a Conceptual Graph



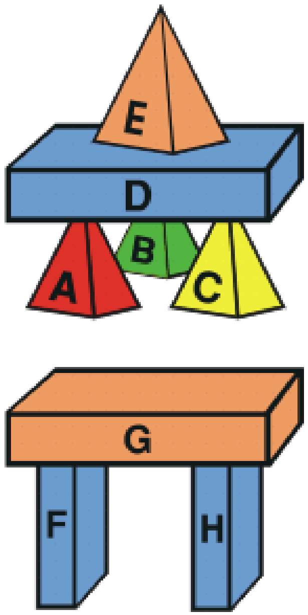


A red pyramid A, a green pyramid B, and a yellow pyramid C support a blue block D, which supports an orange pyramid E.

The concepts (blue) are derived from English words, and the conceptual relations (yellow) from the case relations or thematic roles of linguistics.

Objects, Tables, and Descriptions

Objects represented in database tables:



Objects

Entity	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

The database contents described in English:

> ***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 blue block F and a blue block H suport an orange block G.***

The database is called structured, and English is called unstructured.

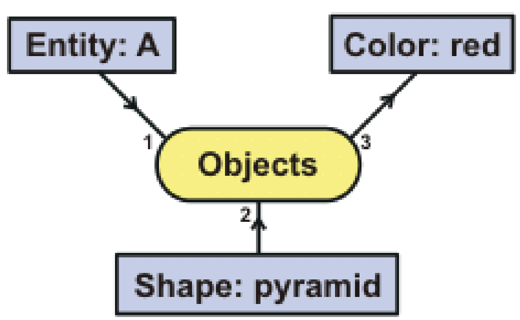
Yet English has more structure, but of a very different kind.

Mapping a Relational DB to Conceptual Graphs

Objects

Entity	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

⇕



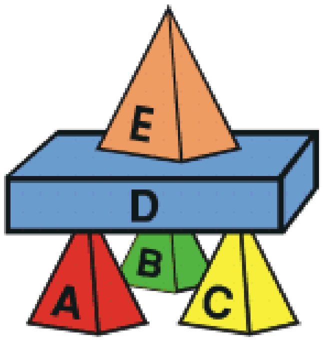
Supports

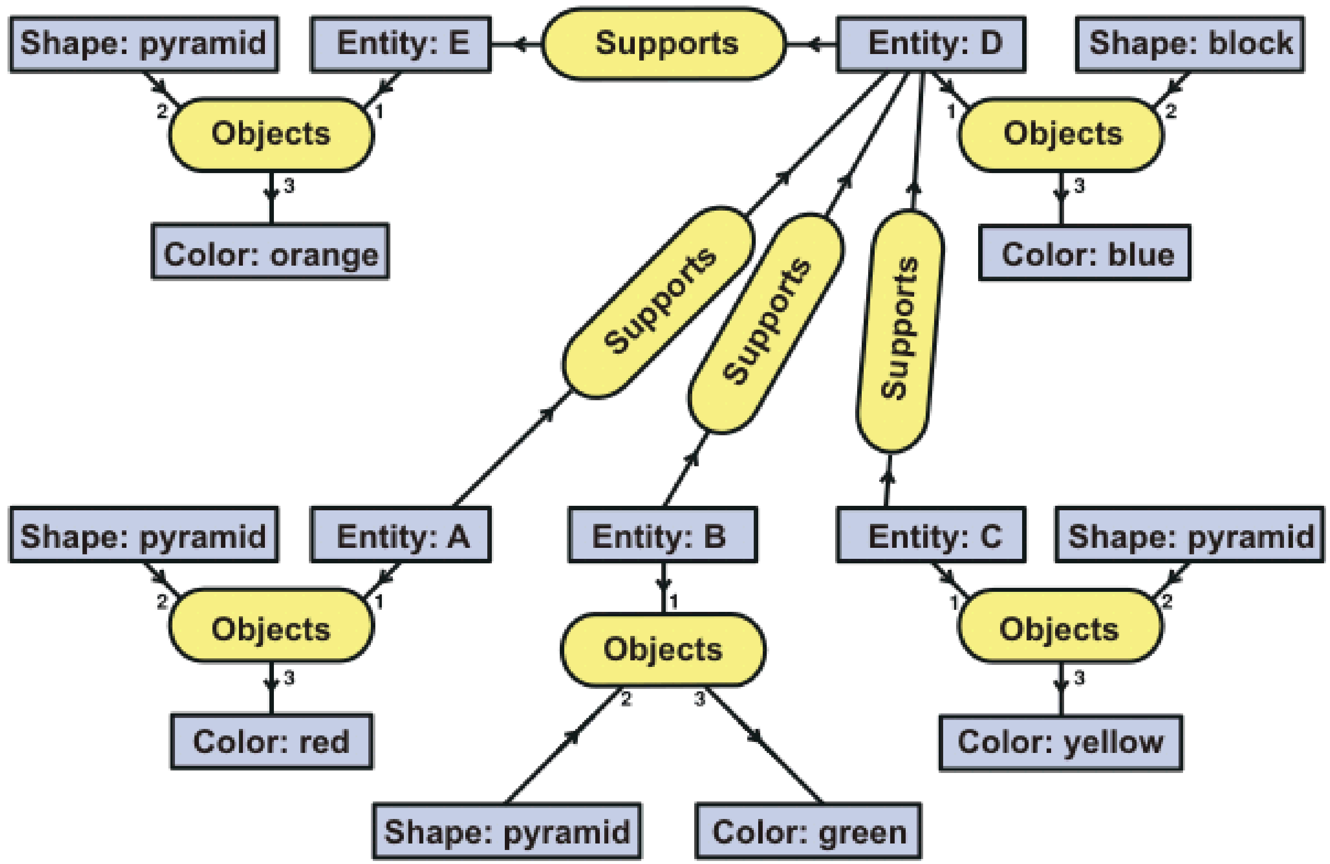
Supporter	Supportee
A	D
B	D
C	D
D	E
F	G
H	G

⇕

Each row of each table maps to a conceptual relation that is linked to as many concepts as there are columns in the table.

Conceptual Graph Derived from the Database

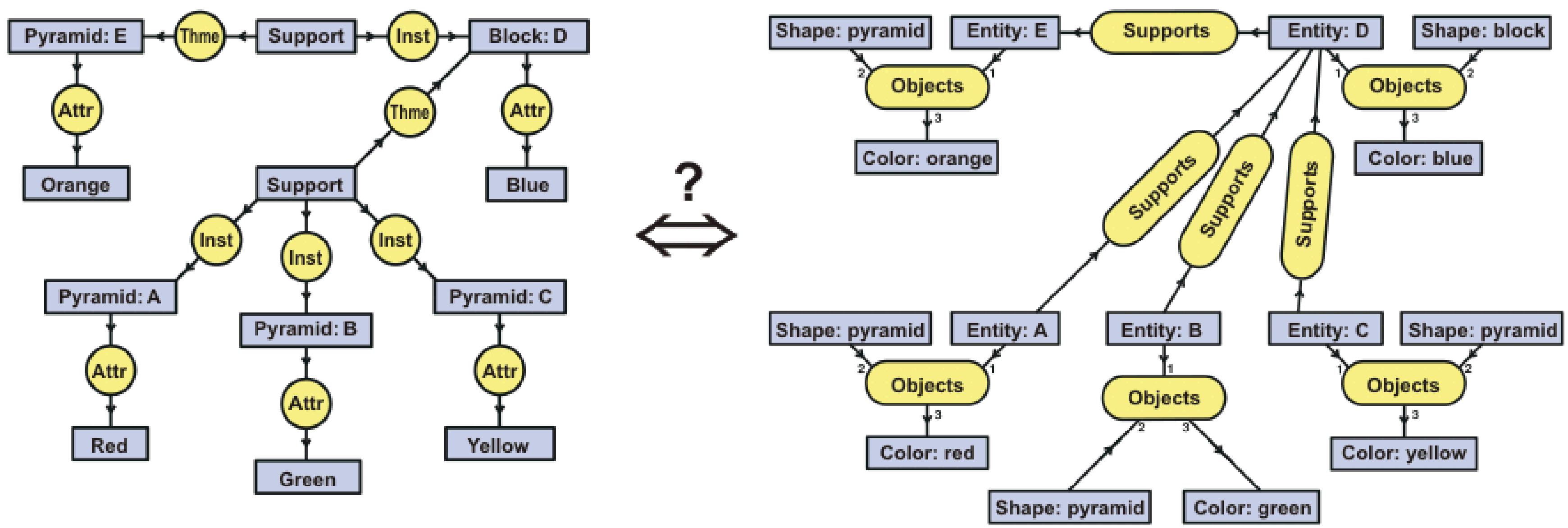




Join concept nodes that refer to the same entities.

Connected graphs represent structures of related objects.

Mapping Two Related Graphs



Very different graphs: 12 concept nodes vs. 15 concept nodes, 11 relation nodes vs. 9 relation nodes, no similarity in type labels.

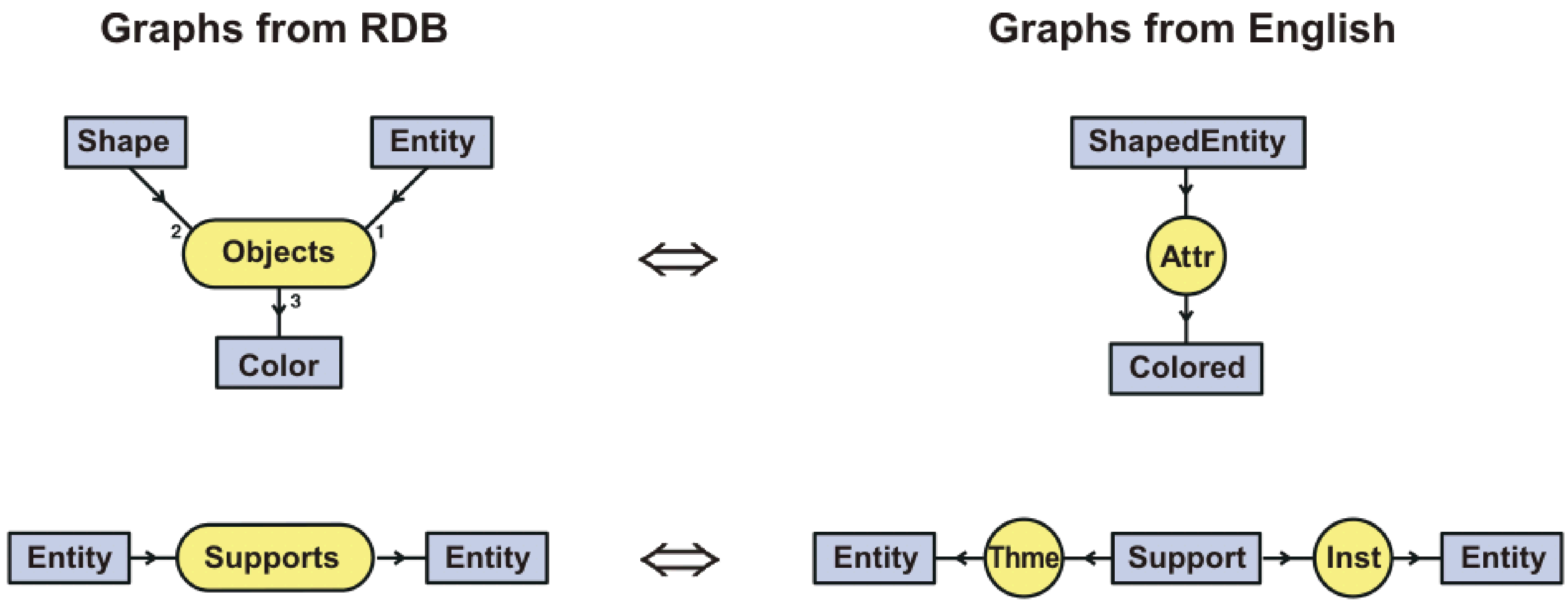
The only commonality is in the five names: A, B, C, D, E.

People can recognize the underlying similarities.

How is it possible for a computer to discover them?

Answer: Use the Kyndi Analogy Engine.

Aligning Ontologies by Structure Mapping



Repeated application of these two transformations perform an exact mapping of all the nodes and arcs of each graph to the other.

This mapping was done by hand in an example by Sowa (2000), Ch 7.

The Kyndi Analogy Engine found the same mapping automatically.

Approximate Mapping for Analogies

Example: *How is a cat like a 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

Answers in this table were found by the Kyndi Analogy Engine.

Approximations

Two factors determine the semantic distance between graphs:

Ontology: Similarity in the types of concepts and relations.

Structure: Similarity in the pattern of nodes and arcs.

Ontology is supplemented by common associations:

- Eyes and headlights are related to light, and there are two of each.
- Heart and engine are internal parts with a regular beat.
- Skeleton and chassis are frames for attaching parts.
- Paws and wheels support the body, and there are four of each.

One-to-one structure matching is preferred:

- Head → Eyes → Cornea.
- Hood → Headlights → Glass plate.

Approximate matches may skip some nodes (marked in red):

- Mouth → Esophagus → Stomach → Bowel → Anus.
- Fuel cap → Fuel tank → Combustion chamber → Muffler → Exhaust pipe.

Exact and Approximate Matching

For logic, CM can find an exact match that unifies two graphs:

- **For example, match a graph from English to a graph from SQL.**
- **These matches are essential for rule-based inference engines.**
- **They are also important for comparing programming statements.**

But CM also finds approximate matches for analogies:

- **Given a graph *g* and a small semantic distance ε, CM can find all graphs within the distance ε from *g* — in log(N) time.**
- **This option is essential for natural languages, which rarely have exact matches, even for sentences with the “same” meaning.**

Applications of Kyndi Technology

Cognitive learning, reasoning, and language understanding.

Cognitive Memory™ uses knowledge from any source:

- Associative retrieval of background knowledge in log(N) time.
- Approximate pattern matching for analogies and metaphors.
- Precise pattern matching for science, mathematics, and logic.

Using CM to derive a domain ontology by reading books:

- Kyndi resources include a general-purpose base ontology.
- But every discovery or innovation creates new ontology.
- CM enables anything learned from one text to be used to interpret other parts of the same text or any other texts.

Three examples of natural language projects:

- Extract information from research reports and map it to a relational DB.
- Legacy re-engineering: Analyze 40 years of legacy software and relate it to the documentation — manuals, reports, memos, and comments.
- Oil and gas exploration: Answer English queries by extracting domain ontology and information from textbooks and research reports.

Information Extraction Project

The next slide shows a table derived from research reports.

To extend the semantics, an ontology for chemistry was added to the basic Kyndi ontology.

Then for each report,

- **Map each sentence to a conceptual graph (CG). ***
- **Analyze anaphoric references to link pronouns to named entities.**
- **The result is a large CG that represents every sentence in the document.**
- **Store that graph (including subgraphs) in Cognitive Memory.**
- **Query Cognitive Memory for the data in each row of the table.**
- **Store the answers in the table.**

In a competition among twelve NLP systems,

- **The Kyndi system got 96% of the entries correct.**
- **The second best score was 73%. Most scores were below 50%.**

*** For an overview of CG methods, see http://www.jfsowa.com/pubs/template.pdf**

Information Extracted from Documents

COMPOUND	CURIE TEMP.	SOURCE
Mn3[Cr(CN)6]2 ·16H2O	89 K	A solid-state hybrid density functional theory study
Sr3Ir2O7 in Sr3Ir2O7 single-cr	~ 280 K	Canted antiferromagnetic ground state in Sr3Ir2O7
PrPt2B2C	6 K	Coexistence of superconductivity and magnetic ord
La0:3Nd0:7Pt2:1-	2.8 K	Coexistence of superconductivity and magnetic ord
NdPt2:1B2:4C1:2	3 K	Coexistence of superconductivity and magnetic ord
NdPt1:5Au0:6B2C	3 K	Coexistence of superconductivity and magnetic ord
SmNiC2	= 17.7 K	Commensurate charge-density wave with frustrate
Co0.2Zn0.8Fe2O4. in CdxCo1-	~ 780 K	Does Ti+4 ratio improve the physical properties of
Zn0.88Co0.12O in ZnO	~ 540 K	Effect of Co doping on the structural; optical and m
La in Sr2-xLaxFeMoO6	425 K	Effect of La doping on the properties of Sr2-xLaxFe
Fe in Sr2-xLaxFeMoO6	~ 1040 K	Effect of La doping on the properties of Sr2-xLaxFe
FeSe	~ 305 K	Electronic and magnetic properties of FeSe thin film
Ni-Mn-Ga	= 376 K	Electronic and structural properties of ferromagnet
LaFexSi1 - x13 in La1-zPrz(Fe	~ 190 K	Enhancement of magnetocaloric effects in La1-zPrz
LaFe0.88Si0.1213 in La1-zPrz(	= 195 K	Enhancement of magnetocaloric effects in La1-zPrz
Co2MnGa in Co2MnGa	600 K	Ferromagnetic resonance in Co2MnGa films with va
HoCrO4 in HoCrO4	17.6 K	Ferromagnetism vs. antiferromagnetism of the dim
Mn3(HCOO)6 in Mn3(HCOO)6	8.0 K	Guest-induced chirality in the ferrimagnetic nanopo
NaZn13- in La0.5Pr0.5(Fe0.88	range from 195 K to 185 K	Large isothermal magnetic entropy change after hy
La2/3Ba1/3MnO3 in La2-3Ba1	range from 300 K to 250 K	Magnetic and neutron diffraction study of La2-3Ba1
CuMnSb in Co1-xCuxMnSb	range from 476 K to 300 K	Magnetic properties of half-metallic semi Heusler C
Nd2 in Nd2-yDyyFe17-xSix	range from 61.46 °C to 236 °	Magnetic properties of iron-rich Nd2-yDyyFe17-xSix
Tb2Fe17 in Nd2-yDyyFe17-xSi	~ 80 °C	Magnetic properties of iron-rich Nd2-yDyyFe17-xSix

Application to Legacy Re-engineering

Analyze the software and documentation of a corporation.

Programs in daily use, some of which were up to 40 years old.

- **1.5 million lines of COBOL programs.**
- **100 megabytes of English documentation — reports, manuals, e-mails, Lotus Notes, HTML, and program comments.**

Goal:

- **Analyze the COBOL programs.**
- **Analyze the English documentation.**
- **Compare the two to determine:**
  - **Data dictionary of all data used by all programs.**
  - **English glossary of all terms with index to the software.**
  - **Evolution of terminology over the years.**
  - **Structure diagrams of the programs, files, and data.**
  - **Discrepancies between programs and documentation.**

An Important Simplification

An extremely difficult and still unsolved problem:

- Translate English specifications to executable programs.

Much easier task:

- Translate the COBOL programs to conceptual graphs (CGs).
- Those CGs provide the ontology and background knowledge.
- The CGs derived from English may have ambiguous options.
- In parsing English, use CGs from COBOL to resolve ambiguities.
- The COBOL CGs show the most likely options.
- They can also provide missing information or detect errors.

The CGs derived from COBOL provide a formal semantics for the informal English texts.

Excerpt from the Documentation

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.

Note that none of the files or COBOL variables are named.

By matching graphs derived from English to graphs derived from COBOL, all names of files and COBOL variables were determined.

Interpreting Novel Patterns

Many documents contain unusual or ungrammatical patterns.

They may be elliptical forms that could be stored in tables.

But some authors wrote them as phrases:

- ***32 — loss on unbilled***
- ***72 — gain on uncollected***
- ***85 — loss on uncollected***

The dashes were represented by a default relation (Link):

[Number: 32]→(Link)→[Punctuation: "–"]→(Link)→[Loss]→(On)→[Unbilled]

This CG, which was derived from an English document, matched CGs derived from COBOL programs:

- **The value 32 was stored as a constant in a COBOL program.**
- **The phrase "loss on unbilled" was in a comment that followed the value 32 in that program.**

Results

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

- **Four weeks for customization:**

  Design, ontology, and additional programming for I/O formats.

- **Three weeks to adapt the software that used Cognitive Memory:**

  Matches with strong evidence (close semantic distance) were correct.

  Weak matches were confirmed or corrected by Majumdar and LeClerc.

- **One week to produce a CD-ROM with the desired results:**

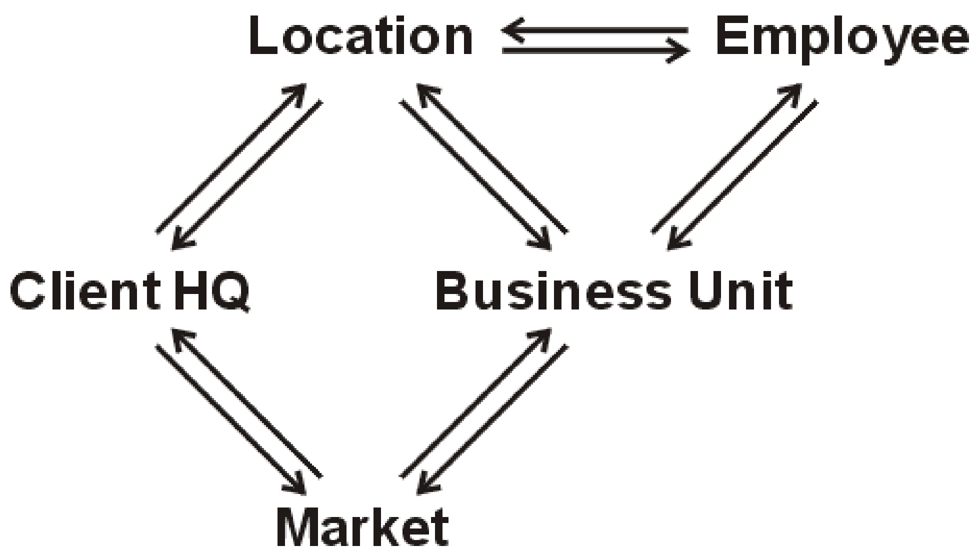
  Glossary, data dictionary, data flow diagrams, process architecture diagrams, system context diagrams, and list of errors detected.

A major consulting firm estimated that the job would take 40 people two years to analyze the documentation and find all cross references.

With Cognitive Memory, the task was completed in 15 person weeks.

Discrepancy Detected

A diagram of relationships among data types in the database:



Question: Which location determines the market?

- **According to the documentation: Business unit.**
- **According to the COBOL programs: Client HQ.**

For many years, management had been making decisions based on incorrect assumptions.

Contradiction Detected

From the ontology used for interpreting English:

- 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.

The software discovered two computer “employees.”

Relating Formal and Informal CGs

The legacy-reengineering task required two kinds of processing.

Precise reasoning:

- **Analyzing the COBOL programs and translating them to CGs.**
- **Detecting discrepancies between different programs.**
- **Detecting discrepancies between programs and documentation.**

Indexing and cross references:

- **Creating an index of English terms and names of programs.**
- **Mapping English documents to the files and programs they mention.**

Conceptual graphs derived from COBOL are precise.

- **But CGs derived from English are informal and unreliable.**
- **Informal CGs are adequate for cross-references between the English documents and the COBOL programs.**
- **All precise reasoning was performed on CGs from COBOL or on CGs from English that were *corrected* by CGs from COBOL.**

Application to Oil and Gas Exploration

Source material:

- 79 documents, ranging in length from 1 page to 50 pages.
- Some are reports about oil or gas fields, and others are chapters from a textbook on geology used as background information.
- English, as written for human readers (no semantic annotations).
- Additional data from relational DBs and other structured sources.
- Lexical resources derived from WordNet, CoreLex, IBM-CSLI Verb Ontology, Roget's Thesaurus, and other sources.
- An ontology for the oil and gas domain written in controlled English by geologists from the University of Utah.
- More ontology derived from the textbooks by Kyndi technology.

Queries:

- A paragraph that describes a potential oil or gas field.
- Analogies compare the query to the documents.

Answering Questions

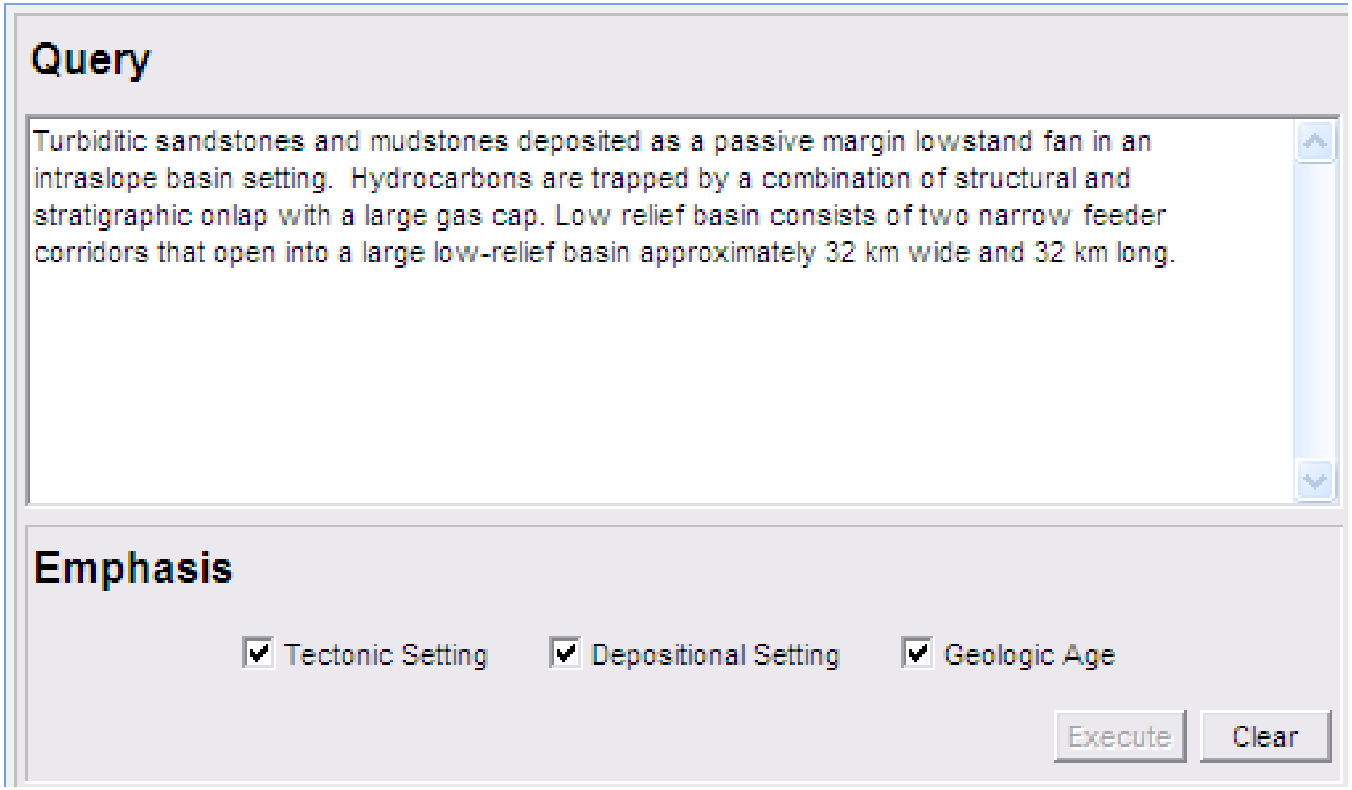
For the sources, either NL documents or structured data:

- **Translate the text or data to conceptual graphs.**
- **Translate all CGs to Cognitive Signatures™ in time proportional to (N log N), where N is the total number of CGs.**
- **Store each Cognitive Signature in Cognitive Memory™ with a pointer to the original CG and the source from which that CG was derived.**
- **Use previously translated CGs to help interpret new sentences.**

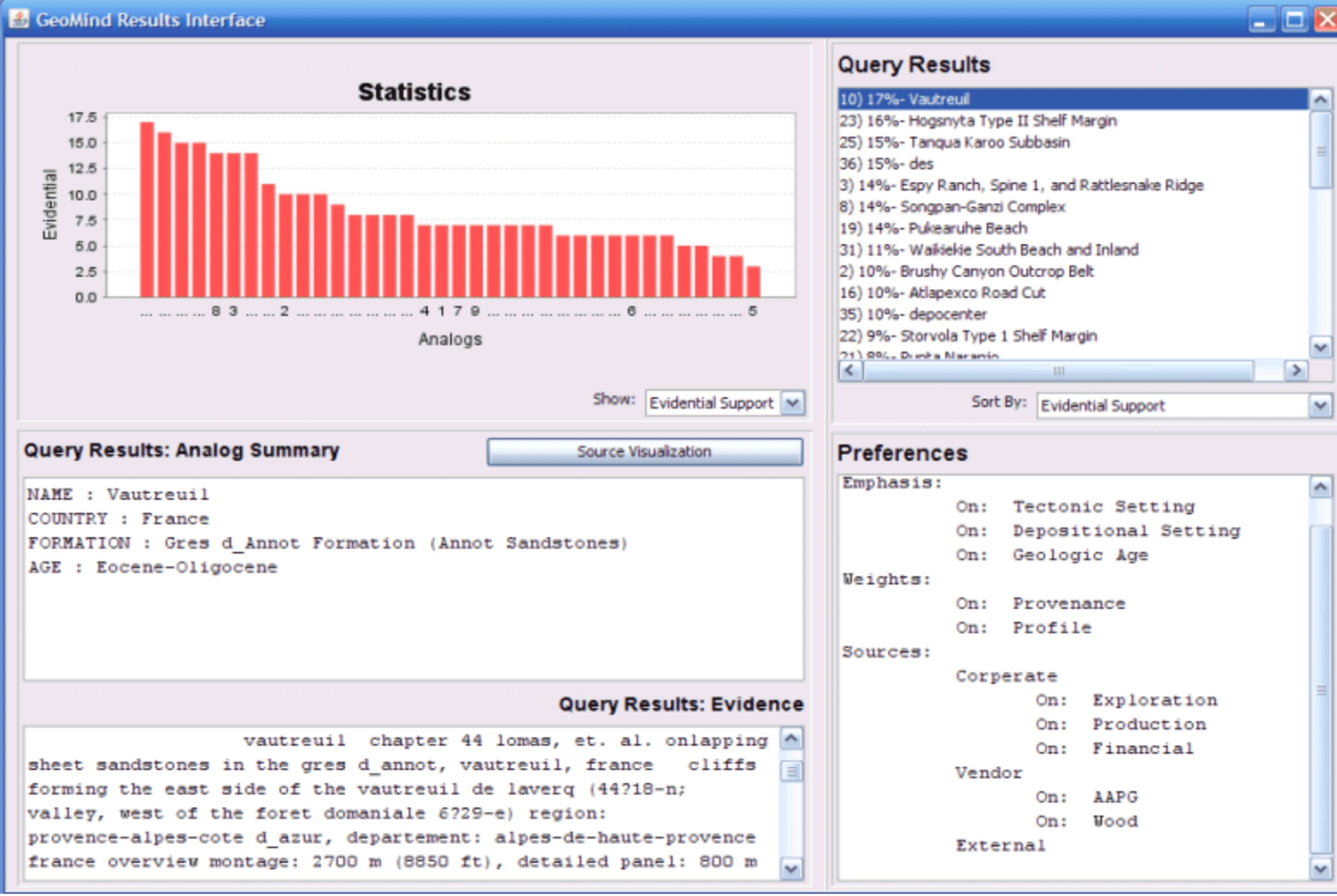
For a query stated as an English sentence or paragraph,

- **Translate the query to conceptual graphs.**
- **Find matching patterns in the source data and rank them in order of semantic distance. The time is proportional to (log N).**
- **For each match within a given threshold, use structure mapping to verify which parts of the query CG match the source CG.**
- **As answer, return the English sentences or paragraphs in the source document that had the closest match to the query.**

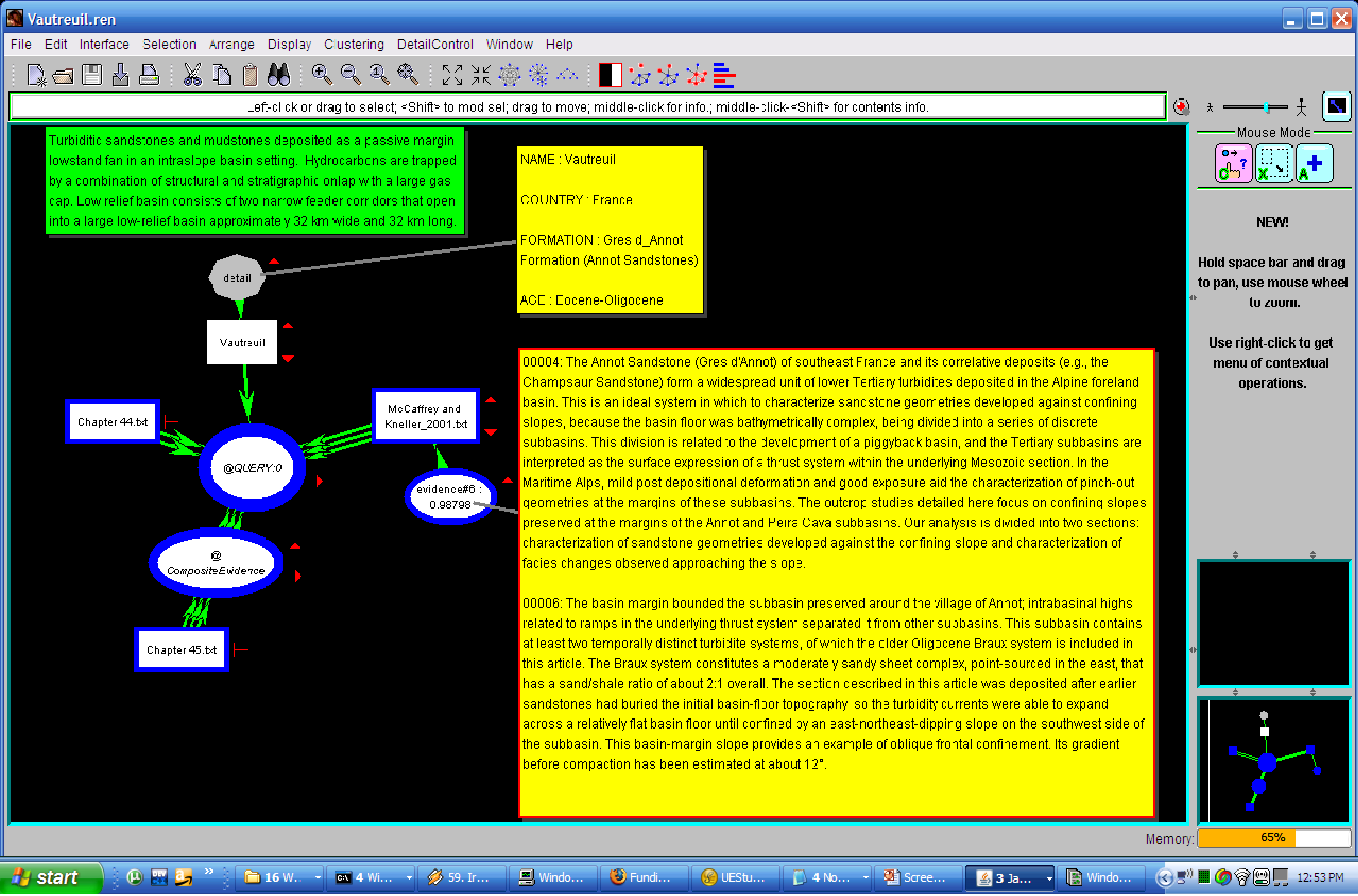
A Query Written by a Geologist



Turbiditic sandstones and mudstones deposited as a passive margin lowstand fan in an intraslope basin setting. Hydrocarbons are trapped by a combination of structural and stratigraphic onlap with a large gas cap. Low relief basin consists of two narrow feeder corridors that open into a large low-relief basin approximately 32 km wide and 32 km long.



Details of the closest matching hydrocarbon fields



Linking the query to the paragraphs that contain the answer

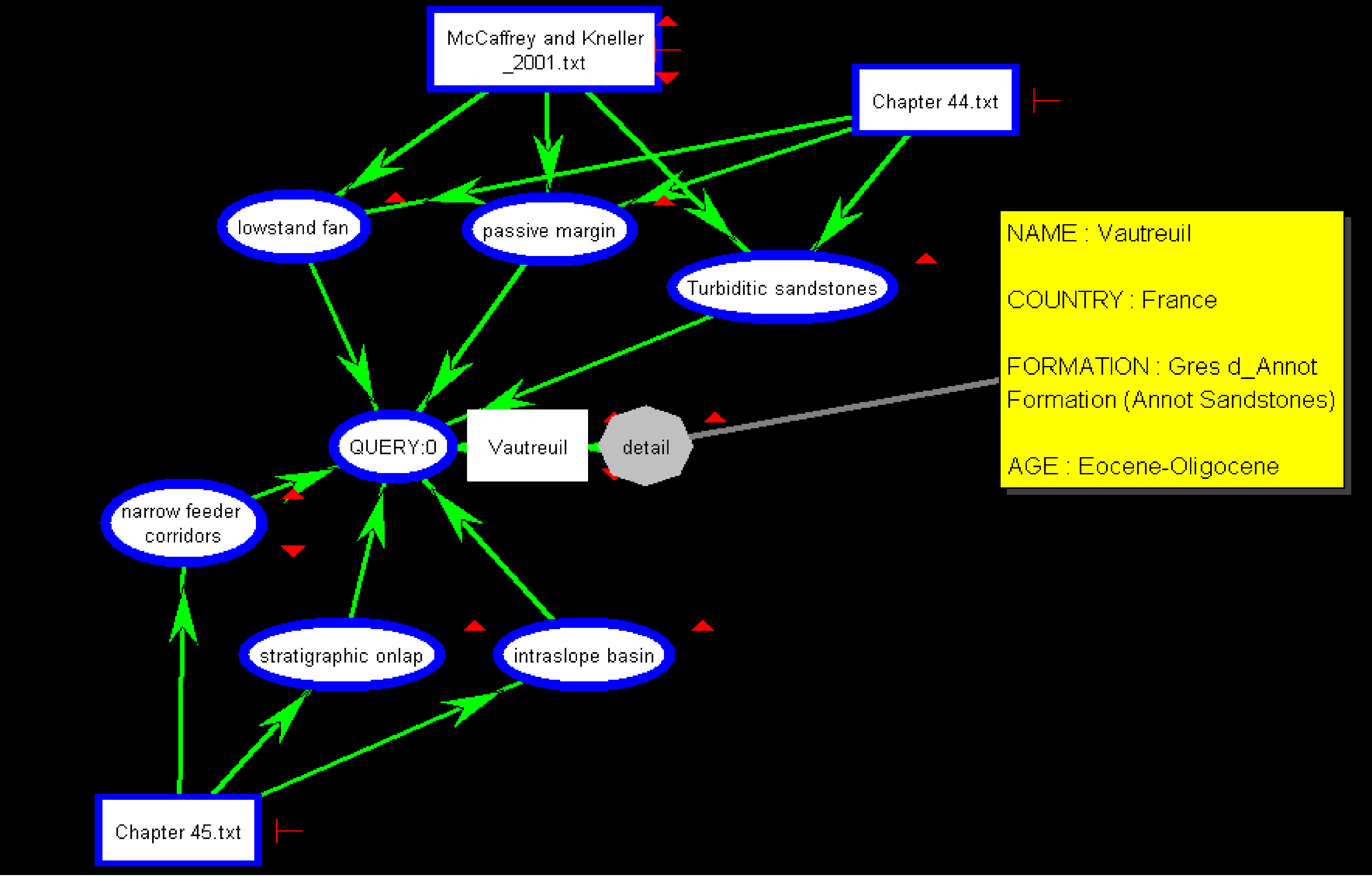
What the Screen Shots Show

Information shown in the previous screen shot:

- **The query in the green box describes some oil or gas field.**
- **The data in the small yellow box describes the Vautreuil field.**
- **The large yellow box shows the paragraphs in a report by McCarthy and Kneller from which that data was extracted.**

The next screen shot shows how the answer was found:

- **Many terms in the query were not defined in the ontology: *lowstand fan, passive margin, turbiditic sandstones, narrow feeder cables, stratigraphic onlap, intraslope basin*.**
- **Generate tentative CGs for these phrases and look in Cognitive Memory to find similar CGs derived from other sources.**
- **Chapters 44 and 45 of the textbook on geology contained those CGs as subgraphs of larger graphs that had related information.**
- **Patterns found in the larger graphs helped relate the CGs derived from the query to CGs derived from the report that had the answer.**



Using background knowledge from a textbook to find the answer

Emergent Knowledge

When reading the 79 documents,

- Translate the sentences and paragraphs to CGs.
- But do not do any further analysis of the documents.

When a geologist asks a question,

- Look for related phrases in Cognitive Memory.
- To connect those phrases, further searches may be needed.
- Some sources may be textbooks with background knowledge that may help interpret the research reports.
- The result consists of CGs that relate the query to paragraphs in research reports that contain the answer.
- The new CGs can be added to Cognitive Memory for future use.

By a “Socratic” dialog, a geologist can lead the system to explore novel paths and discover unexpected patterns.

References

Research that established the foundations for Kyndi technology:

Majumdar, Arun K. (2013) Relativistic concept measuring system for data clustering, US Patent 8,526,321 B2.

Majumdar, Arun K. (2015) Cognitive memory encoding networks for fast semantic indexing, storage, and retrieval, US Patent 9,158,847 B1.

Majumdar, Arun K., John F. Sowa, & John Stewart (2008) Pursuing the goal of language understanding, http://www.jfsowa.com/pubs/pursuing.pdf

Majumdar, Arun K., & John F. Sowa (2009) Two paradigms are better than one and multiple paradigms are even better, http://www.jfsowa.com/pubs/paradigm.pdf

Majumdar, Arun K., & John F. Sowa (2018) Relating language, logic, and imagery, http://jfsowa.com/pubs/relating.pdf

Sowa, John F. (2002) Architectures for intelligent systems, http://www.jfsowa.com/pubs/arch.pdf

Sowa, John F., & Arun K. Majumdar (2003) Analogical reasoning, http://www.jfsowa.com/pubs/analog.htm

Sowa, John F. (2005) The challenge of knowledge soup, http://www.jfsowa.com/pubs/challenge.pdf

Sowa, John F. (2006) Worlds, models, and descriptions, http://www.jfsowa.com/pubs/worlds.pdf

Sowa, John F. (2008) Conceptual graphs, http://www.jfsowa.com/cg/cg_hbook.pdf

Sowa, John F. (2010) Role of Logic and Ontology in Language and Reasoning, http://www.jfsowa.com/pubs/rolelog.pdf

Sowa, John F. (2011) Cognitive architectures for conceptual structures, http://www.jfsowa.com/pubs/ca4cs.pdf

Sowa, John F. (2013) From existential graphs to conceptual graphs, http://www.jfsowa.com/pubs/eg2cg.pdf

ISO/IEC standard 24707 for Common Logic (which includes conceptual graphs as one of the dialects), http://standards.iso.org/ittf/PubliclyAvailableStandards/c039175_ISO_IEC_24707_2007(E).zip