Cognitive Memory™
For Language, Learning, and Reasoning

Arun K. Majumdar and John F. Sowa
Kyndi, Inc

10 February 2018
Contents

1. Cognitive Memory™ (CM) ........................................... 3
   CM supports efficient, human-like methods of learning, reasoning, and language understanding.

2. Using CM for evaluating student answers. ............... 19
   Case-based reasoning to evaluate and comment on free-form English answers to exam questions about high-school mathematics.

3. From perception to cognition. ................................. 31
   Common methods of machine learning recognize perceptual patterns. For complex reasoning, cognition must go beyond perception.

4. Three applications of Kyndi technology ................... 45
   Information extraction. Compare software implementations to their specifications. Derive an ontology by reading textbooks and use it to analyze research reports and to answer questions about them.
1. Cognitive Memory™ (CM)

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 is central to all aspects of AI and NLP methods: formal or informal, crisp or fuzzy, symbolic or subsymbolic.
Kyndi Technology

A cognitive architecture for language, learning, and reasoning. Conceptual graphs for mapping to and from languages and notations of any kind — natural or artificial, formal or informal.

A generalized parser for natural or artificial languages.
- Learning by reading documents with no prior annotations or markup.
- Using the new knowledge to analyze texts in any language.
- Translating the results to any desired format.

Cognitive Memory™ for associative retrieval of graphs.
- With N graphs, find approximate matches in log(N) time.
- Basis for the Kyndi parser, analogy engine, and reasoning modules.

Flexible Modular Framework (FMF) for integrating modules:
- Message passing among heterogeneous agents that may use an open-ended variety of methods for learning and reasoning.
- Influenced by Marvin Minsky’s Society of Mind, John McCarthy’s Elephant 2000, and David Gelernter’s Linda.
Cognitive Memory™

High-speed associative memory for all Kyndi modules
Structure Preserving

A cognitive signature™ encodes the exact structure of a graph.
- It is a lossless encoding, similar to a Gödel numbering. *
- For unlabeled graphs, integers are sufficient for a cognitive signature.
- For example, 0 maps to and from an empty graph with no nodes or arcs.
- 1, 2, 3, 4, 5, and 6 can be mapped to and from the following graphs:
  - ![Graph 1](image1)
  - ![Graph 2](image2)
  - ![Graph 3](image3)
  - ![Graph 4](image4)
  - ![Graph 5](image5)
  - ![Graph 6](image6)
- To encode the structure of conceptual graphs in Cognitive Memory, the cognitive signatures are based on generalized combinatorial maps. **

By contrast, a word vector encodes labels, but not structure.
- A word vector is based on a “bag of words” that ignores the connectivity.
- Word vectors are useful for measuring the similarity of texts.
- But they discard the structural information necessary for reasoning, question answering, and language understanding.

* For methods of encoding and mapping graphs and other data types, see Tarau (2009) “An embedded declarative data transformation language”: article and slides.
** For Cognitive Memory, see the Kyndi patents by Arun Majumdar.
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.
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.
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 support 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.
Each row of each table maps to a conceptual relation that is linked to as many concepts as there are columns in the table.
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?*

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cat</strong></td>
<td><strong>Car</strong></td>
</tr>
<tr>
<td>Head</td>
<td>Hood</td>
</tr>
<tr>
<td>Eye</td>
<td>Headlight</td>
</tr>
<tr>
<td>Cornea</td>
<td>Glass plate</td>
</tr>
<tr>
<td>Mouth</td>
<td>Fuel cap</td>
</tr>
<tr>
<td>Stomach</td>
<td>Fuel tank</td>
</tr>
<tr>
<td>Bowel</td>
<td>Combustion chamber</td>
</tr>
<tr>
<td>Anus</td>
<td>Exhaust pipe</td>
</tr>
<tr>
<td>Skeleton</td>
<td>Chassis</td>
</tr>
<tr>
<td>Heart</td>
<td>Engine</td>
</tr>
<tr>
<td>Paw</td>
<td>Wheel</td>
</tr>
<tr>
<td>Fur</td>
<td>Paint</td>
</tr>
</tbody>
</table>

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.
Computational Complexity

Research by Falkenhainer, Forbus, & Gentner: *

- Pioneers in finding analogies with the Structure Mapping Engine.
- Showed that SME algorithms take time proportional to $N^3$, where $N$ is the number of frames or graphs in the knowledge base.
- MAC/FAC approach: Use a search engine to narrow down the number of likely candidates before using SME.

Kyndi methods:

- Encode graph structure and ontology in a Cognitive Signature™.
- For any graph, find closely matching signatures in log($N$) time.
- Only graphs with similar signatures are likely candidates.
- Log-time algorithms scale to the size of the WWW. CM itself is as fast as a typical search engine — and far more precise.

* See [http://www.qrg.northwestern.edu/papers/papers.html](http://www.qrg.northwestern.edu/papers/papers.html)
Logarithmic Algorithms

Graphs of organic molecules are similar to conceptual graphs:

- Atoms ⇒ concept nodes labeled by the name of the element.
- Chemical bonds ⇒ relation nodes labeled by the name of the bond type.
- But conceptual graphs have many more types of concepts and relations.

Chemical graphs inspired Peirce’s existential graphs as representations of “the atoms and molecules of logic.”

Some of the largest and most sophisticated systems for graph processing were developed by chemists.

Logarithmic search algorithms for chemical graphs:


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 $\varepsilon$, CM can find all graphs within the distance $\varepsilon$ 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.
- See Section 2 for the examples of case-based reasoning.
- The examples in Section 4 use approximate matches more often than exact matches.
2. Evaluating Student Answers

A publisher developed a textbook for high-school math.
• The teacher’s manual would contain exam questions.
• And the publisher wanted software for evaluating the answers.
• Multiple-choice questions are easy to evaluate.
• Long essays are often evaluated by statistical methods.
• But English sentences about mathematics are much harder.

Two companies competed to develop the software.
• One company used statistical methods.
• The other used logic-based methods.
• Neither one could correctly evaluate the student answers.

After those failures, the publisher wanted another option.
• Kyndi solution: Use CM and case-based reasoning.
• A working prototype was implemented in two weeks.
• It exceeded the publisher’s expectations.
A Typical Problem

Sample question:

The following integers 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.

A correct answer:

To show that \( N \) is 1 more than a square, show that \( N-1 \) is a square. Find some integer \( x \) whose square is somewhat less than \( N-1 \). Compare \( N-1 \) to the squares of \( x, x+1, x+2, x+3, ... \), and stop when some square is equal to or greater than \( N-1 \). If this square is \( N-1 \), then \( N \) is 1 more than a square.

Requirements: Design and implement software that would

- Determine whether an answer is correct, incorrect, or partially correct.
- If incorrect or partially correct, determine what is wrong and print a helpful message to guide the student.
The Competition

The publisher had previously worked with two companies.

- C1 proposed statistics and word vectors for this task.
- C2 proposed logic, ontology, and an English parser.
- Neither method met the requirements.

After those failures, the publisher wanted another option.

- Arun Majumdar and John Sowa proposed an alternative.
- John suggested CM plus case-based reasoning.
- Arun implemented a working version in two weeks.
- It exceeded the publisher’s requirements and expectations.
- Unfortunately, the publisher’s project manager died in a traffic accident, company C1 appealed to upper management, and the software project was canceled.

Although this project was canceled, it illustrates methods that Kyndi has successfully used in many other projects.
Why Company C1 Failed

Statistics are often used to evaluate student essays:
- The statistics for grammar, style, and discourse markers are based on human evaluations for thousands of essays.
- The content of an essay is represented by a word vector derived from the frequencies of various words in the essay.
- For comparison, an expert writes a “gold standard” essay that would receive a grade of A+ as an answer to the question.
- A student essay is evaluated by comparing it to the gold standard.

But statistics are unreliable for short texts about math:
- A short text doesn’t have enough data for reliable statistics.
- As in slide 20, the same words tend to occur in the question and in the answers, both correct and incorrect.
- Word vectors, which ignore the word order, cannot distinguish them.
- For precise reasoning, a Cognitive Signature™ combines ontology and structure, but a word vector throws away the structure.

The methods proposed by C1 were unacceptable.
Why Company C2 Failed

C2 used AI technology that preserves the logical structure:
  ● A large general-purpose ontology, which is adapted to each task.
  ● A parser that uses the ontology to translate English to logic.
  ● A rule-based inference engine to derive conclusions from the logic.

But the system was too complex and fragile:
  ● Every new problem required special-purpose ontology and rules.
  ● The skills for writing the ontology and rules were beyond the ability of most high-school math teachers.
  ● The parser could handle some well-written, well-edited examples, but it could not parse most of the answers written by students.
  ● The inference engine could not evaluate most of the answers.
  ● Even when it could distinguish some correct and incorrect answers, it could not detect and comment on partially correct answers.

The methods proposed by C2 were unacceptable.
Why Kyndi Technology Succeeded

It could use the data that the publisher had already gathered.

To evaluate new exam questions, the publisher would give the exam to several groups of students.

For each problem, they would get about 50 different answers:

- Some are completely correct
  — but stated in different ways.
- Some are partially correct
  — and the teacher must say what is missing.
- Others are wrong
  — in many different ways.

Some teacher would evaluate each answer (correct, incorrect, or partially correct) and write a short comment.

Result: 50 pairs of student answer and teacher response.

Each answer-response pair is a case for case-based reasoning.
Case-Based Reasoning (CBR)

Given the same cases, analogy takes one step to derive an answer, but induction and deduction take multiple steps.

Analogy is fast and flexible, but a theory is valuable if it can be used and reused in multiple applications.
Using Case-Based Reasoning

Translate all student answers to conceptual graphs (CGs):

- The Kyndi parser starts with a base ontology that is extended with some application-related conceptual graphs. (See the next slide.)
- New CGs derived by the Kyndi parser can also be stored in CM.
- With the high-speed search by CM, the Kyndi parser can extend its ontology with any or all the CGs it had previously derived.

Compare each student answer to the 50 previous cases:

- Compare the CG for the new answer to the CGs stored in CM.
- If there is a good match, print out the teacher’s previous response.
- Otherwise, send the new student answer to some teacher to evaluate.
- Add the new answer-response pair to the collection of cases in CM.

In effect, the high-school teachers train Cognitive Memory™:

- They don’t need to learn any new skills.
- They just evaluate student answers in the same way they always did.
- Every new evaluation by a teacher improves CM’s ability to evaluate more student answers without asking for help.
Conceptual Graph for Multiply

This CG is part of the ontology for the concept type Multiply:

[Someone] multiplies a number by a number to get a product.

The diamond node, called an actor, represents a function that multiplies two numbers to compute their product.
“Multiply a number by itself to get a square.”

The English command causes a change in the mathematics:

- The two concept nodes of type Number are joined (merged).
- That join causes both arguments of Multiply to be the same.
- For simple problems, there is no need for a special inference engine.
- For more complex problems, the graphs may be translated to special notations for mathematics, or programming languages.
Results

The analogy engine found a good match for most answers.

Even though many answers were poorly written, the CGs stored in CM made the parser more robust —

- By resolving ambiguities and showing expected combinations,
- By correcting and compensating for errors in syntax,
- By suggesting defaults for missing arguments, and
- By relating multiple fragments to form complete sentences.

For any answer with a close match to a previous answer,

- The previous teacher evaluation was correct,
- And the previous comments were appropriate.

If no match within a small semantic distance,

- The student answer could be sent to a teacher for evaluation,
- The new answer-response pair would be added to CM,
- Over time, the coverage and accuracy would improve.
Learning by CM is Human-like

Cognitive Memory™ can learn from a small number of examples:
- For case-based reasoning, a single example is enough to start.
- Each new case broadens the selection and improves the results.

CM can produce explanations in natural language:
- Every item in CM is a CG, which can be translated to a sentence in NL.
- For any response, the most similar cases can be translated to sentences with an estimate of their similarity to the input case.

A typical explanation in English:
- “I did X because whenever my friends and I did something like X, we got good results, such as A, B, C.”

CM can be used in conjunction with other AI technologies:
- The CGs stored in CM can be translated to or from other versions of logic, programming languages, or database languages.
- Patterns learned by neural nets can also be mapped to or from CGs.
Deep neural nets (DNNs) were designed for perceptual learning.

- Require huge amounts of data to learn patterns.
- Learn complex patterns as well or better than humans.
- But they can’t explain or reason about the patterns they learn.

Traditional AI tools were designed for cognition:

- Parsers and translators for natural and artificial languages.
- Inference engines and rule-based systems for formal logics and many kinds of informal or fuzzy reasoning.
- These tools can be used with DNNs in hybrid systems.

Cognitive Memory™ does learning and reasoning for cognition.

- Analogies for induction, abduction, and case-based reasoning.
- Learn cognitive structures from small amounts of data — even “one-shot learning” from a single example.
What Makes People Intelligent?

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. §30.8

Language and culture enhance the diversity.

• 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 adopt technology from another culture, they also adopt and adapt new words and ways of thinking.

See also Push Singh & Marvin Minsky (2005) An architecture for cognitive diversity.
Most ML methods learn to approximate a function $f: x \rightarrow y$, where $x$ and $y$ are vectors of observable features or attributes. *

Unsupervised learning begins with a set of pairs of the form $(x,y)$ and computes an estimated probability $p(x,y)$ for any $x$ and $y$.

For classification, $p(x \mid y)$ is the probability that something described by the feature vector $y$ belongs to a class $x$.

For prediction, $p(y \mid x)$ is the probability that a state described by a vector of features $x$ will be followed by a state described by $y$.

Such functions represent the kind of learning that psychologists analyzed and described by stimulus-response (S-R) theories.

But S-R theories could only explain the early stages of perceptual learning. They could not explain complex language and reasoning.

* Henry Lin & Max Tegmark (2016) *Why does deep and cheap learning work so well?*
Deep Neural Networks

DNNs are highly efficient versions of artificial neural networks.

- Early ANNs with multiple hidden layers were slow to learn the weights on the links that define the functions.
- The improved algorithms for DNNs enable them to learn complex functions from larger volumes of data in much less time.
Machine Learning Applications

<table>
<thead>
<tr>
<th>Stimulus $x$</th>
<th>Response $y$</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Picture</td>
<td>Are there human faces? (0 or 1)</td>
<td>Photo tagging</td>
</tr>
<tr>
<td>Loan application</td>
<td>Will they repay the loan? (0 or 1)</td>
<td>Loan approvals</td>
</tr>
<tr>
<td>Ad plus user information</td>
<td>Will user click on ad? (0 or 1)</td>
<td>Targeted online ads</td>
</tr>
<tr>
<td>Audio clip</td>
<td>Transcript of audio clip</td>
<td>Speech recognition</td>
</tr>
<tr>
<td>English sentence</td>
<td>French sentence</td>
<td>Language translation</td>
</tr>
<tr>
<td>Sensors from hard disk, plane engine, etc.</td>
<td>Is it about to fail?</td>
<td>Preventive maintenance</td>
</tr>
<tr>
<td>Car camera and other sensors</td>
<td>Position of other cars</td>
<td>Self-driving cars</td>
</tr>
</tbody>
</table>

Learning a function $f: x \rightarrow y$ is the basis for perception.

- Observation by Andrew Ng: Current ML methods automate tasks that take less than one second of mental effort by humans. *
- Every one of Ng’s examples recognizes a pattern.
- None of them do cognitive reasoning or language understanding.

Limitations of DNNs

Deep neural nets, by themselves, do perceptual learning.

Using DNNs to learn Atari games: *

- Outperform all other machine-learning methods on 6 of 7 games.
- But far from human performance on Q*bert, Seaquest, and Space Invaders — because those games require long-term strategy.

Comparison with Art Samuel’s checker-playing system (1959):

- DNNs are much more powerful than Samuel’s learning algorithm.
- But lookahead methods are necessary for long-term strategy.
- Samuel’s system was a hybrid that combined his learning method with alpha-beta search for long-term strategy (cognitive reasoning).
- It was good enough to beat the Connecticut state checker champion.

Human-like intelligence requires cognition.

AlphaGo by DeepMind

Major breakthrough in learning to play Go.

- Computationally, the game of Go is far more challenging than chess.
- For both games, pattern recognition and tree search are important for high-level play.
- But Go has about 250 options at each step, and chess has about 37.
- The search strategies used for chess programs are inadequate for playing a good game of Go.

AlphaGo won 9 of 10 games with Go masters. *

- A hybrid system with DNNs and Monte Carlo Tree Search (MTCS).
- But the DNNs were trained on millions of games, far more than any human could play in a lifetime.
- And the MTCS search strategy is also used in other Go systems.

A Hybrid with DNNs and Parse Trees

Geometry Problem Solver (GeoS)

GeoS solves typical problems on the Geometry SAT exam. * It’s a hybrid that relates imagery to language and mathematics.

* Developed by the Allen Institute for AI and the University of Washington.
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.
14 participants studied how four devices work: bathroom scale, fire extinguisher, disc brake system, and trumpet. *

- Subjects: college students who were not science or engineering majors.
- Multiple training sessions and test sessions with each of the four devices.
- During test sessions, an fMRI scanner recorded patterns of brain activity.
- An early training session just showed pictures and named the parts: A bathroom scale consists of a spring, a lever, a ratchet, and a dial.
- Later sessions explained structural and causal relations: The spring pulls a ratchet which rotates a gear attached to a measurement dial.

Cognitive Learning

Neural activity in the right hemisphere during test sessions:

- All 14 students showed similar neural activations.
- Questions about the objects and parts activated the visual cortex, the occipital lobes in the back of the brain (image #1).
- Questions about structural relations activated the parietal lobes, which link vision to all sensory and motor regions (image #2).
- Questions about the causal effects of someone operating the system activated the frontal lobes and connections across the brain (image #3).
- Summary: Cognitive learning involves structural and causal relations that link and coordinate perception, action, and reasoning.
Long-Distance Connections

Most neurons have short links to nearby neurons.

But others make long-distance connections.

The diagram shows long-distance connections among areas of the brain involved in language. *

The blue boxes represent areas in the frontal lobes; yellow for an area in the parietal lobe; green for an area in the temporal lobe.

Patterns can be learned and recognized in one area.

But cognition relates many areas across the brain.

* Diagram adapted from MacNeilage (2008).
4. 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
<table>
<thead>
<tr>
<th><strong>COMPOUND</strong></th>
<th><strong>CURIE TEMP.</strong></th>
<th><strong>SOURCE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mn₃[Cr(CN)₆]·16H₂O</td>
<td>89 K</td>
<td>A solid-state hybrid density functional theory study.</td>
</tr>
<tr>
<td>Sr₃Ir₂O₇ in Sr₃Ir₂O₇ single-cr</td>
<td>~ 280 K</td>
<td>Canted antiferromagnetic ground state in Sr₃Ir₂O₇.</td>
</tr>
<tr>
<td>PrPt₂B₂C</td>
<td>6 K</td>
<td>Coexistence of superconductivity and magnetic order.</td>
</tr>
<tr>
<td>La₀:3Nd₀:7Pt₂:1-</td>
<td>2.8 K</td>
<td>Coexistence of superconductivity and magnetic order.</td>
</tr>
<tr>
<td>NdPt₂:1B₂:4C₁:2</td>
<td>3 K</td>
<td>Coexistence of superconductivity and magnetic order.</td>
</tr>
<tr>
<td>NdPt₁:5Au₀:6B₂C</td>
<td>3 K</td>
<td>Coexistence of superconductivity and magnetic order.</td>
</tr>
<tr>
<td>SmNiC₂</td>
<td>17.7 K</td>
<td>Commensurate charge-density wave with frustrated magnetic order.</td>
</tr>
<tr>
<td>Co₀.2Zn₀.8Fe₂O₄. in CdₓCo₁-xO</td>
<td>~ 780 K</td>
<td>Does Ti+4 ratio improve the physical properties of Co?</td>
</tr>
<tr>
<td>Zn₀.8Co₀.12O in ZnO</td>
<td>~ 540 K</td>
<td>Effect of Co doping on the structural; optical and magnetic properties.</td>
</tr>
<tr>
<td>La in Sr₂-xLaxFeMoO₆</td>
<td>425 K</td>
<td>Effect of La doping on the properties of Sr₂-xLaxFeMoO₆.</td>
</tr>
<tr>
<td>Fe in Sr₂-xLaxFeMoO₆</td>
<td>~ 1040 K</td>
<td>Effect of La doping on the properties of Sr₂-xLaxFeMoO₆.</td>
</tr>
<tr>
<td>FeSe</td>
<td>~ 305 K</td>
<td>Electronic and magnetic properties of FeSe thin films.</td>
</tr>
<tr>
<td>Ni-Mn-Ga</td>
<td>376 K</td>
<td>Electronic and structural properties of ferromagnetic materials.</td>
</tr>
<tr>
<td>LaFexSi₁-x in La₁-zPrz₃(Fe₀.81)</td>
<td>~ 190 K</td>
<td>Enhancement of magnetocaloric effects in La₁-zPrz₃.</td>
</tr>
<tr>
<td>LaFe₀.88Si₀.1213 in La₁-zPrz₃</td>
<td>~ 195 K</td>
<td>Enhancement of magnetocaloric effects in La₁-zPrz₃.</td>
</tr>
<tr>
<td>Co₂MnGa in Co₂MnGa</td>
<td>600 K</td>
<td>Ferromagnetic resonance in Co₂MnGa films with vacancies.</td>
</tr>
<tr>
<td>HoCrO₄ in HoCrO₄</td>
<td>17.4 K</td>
<td>Ferromagnetism vs. antiferromagnetism of the dimers.</td>
</tr>
<tr>
<td>Mn₃(HCOO)₆ in Mn₃(HCOO)₆</td>
<td>8.0 K</td>
<td>Guest-induced chirality in the ferrimagnetic nanoparticle.</td>
</tr>
<tr>
<td>NaZn₁₃ in La₀.5Pr₀.5(Fe₀.88</td>
<td>range from 195 K to 185 K</td>
<td>Large isothermal magnetic entropy change after hydration.</td>
</tr>
<tr>
<td>La₂/3Ba₁/3MnO₃ in La₂-3Ba₁</td>
<td>range from 300 K to 250 K</td>
<td>Magnetic and neutron diffraction study of La₂-3Ba₁.</td>
</tr>
<tr>
<td>CuMnSb in Co₁-xCuxMnSb</td>
<td>range from 476 K to 300 K</td>
<td>Magnetic properties of half-metallic semi Heusler compounds.</td>
</tr>
<tr>
<td>Nd₂ in Nd₂-yDy₃Fe₁₇-xSix</td>
<td>range from 61.46 °C to 236 °C</td>
<td>Magnetic properties of iron-rich Nd₂-yDy₃Fe₁₇-xSix.</td>
</tr>
<tr>
<td>Tb₂Fe₁₇ in Nd₂-yDy₃Fe₁₇-xSix</td>
<td>~ 80 °C</td>
<td>Magnetic properties of iron-rich Nd₂-yDy₃Fe₁₇-xSix.</td>
</tr>
</tbody>
</table>
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.
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.
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

00004: The Annecy Sandstone (Gres d’Annot) of southeast France and its correlative deposits (e.g., the Chamrousse Sandstone) form a widespread unit of lower Tertiary turbidites deposited in the Alpine foreland basin. This is an ideal system in which to characterize sandstone geometries developed against confining slopes, because the basin floor was bathymetrically complex, being divided into a series of discrete subbasins. This division is related to the development of a piggyback basin, and the Tertiary subbasins are interpreted as the surface expression of a thrust system within the underlying Mesozoic section. In the Maritime Alps, mild post depositional deformation and good exposure aid the characterization of pinch-out geometries at the margins of these subbasins. The outcrop studies detailed here focus on confining slopes preserved at the margins of the Annecy and Peira Cava subbasins. Our analysis is divided into two sections: characterization of sandstone geometries developed against the confining slope and characterization of facies changes observed approaching the slope.

00008: The basin margin bounded the subbasin preserved around the village of Annecy intracalinal highs related to ramps in the underlying thrust system separated it from other subbasins. This subbasin contains at least two temporally distinct turbidite systems, of which the older Oligocene Braux system is included in this article. The Braux system constitutes a moderately sandy sheet complex, point sourced in the east, that has a sand/shale ratio of about 2:1 overall. The section described in this article was deposited after earlier sandstones had buried the initial basin-floor topography, so the turbidity currents were able to expand across a relatively flat basin floor until confined by an east-northeast-dipping slope on the southwest side of the subbasin. This basin-margin slope provides an example of oblique frontal confinement. Its gradient before compaction has been estimated at about 12°.
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.
For AI, human-like cognition is the ultimate goal.
- DNNs can simulate perceptual learning by the cerebellum.
- But cognitive learning is performed by the cerebral cortex.
- Both are necessary, but they serve different roles.

To see the difference, consider a shark and a dolphin:
- They are about the same size, and they hunt the same prey.
- A shark has a large cerebellum, but a tiny forebrain.
- A dolphin has a huge cerebellum and a huge cerebral cortex.
- A group of sharks devour their prey in a food frenzy.
- But a group of dolphins can communicate, organize a hunt, and systematically surround and trap their prey.
- Dolphins also train their young, care for each other, and are friendly with humans — desirable traits in AI systems.

For more about the neuroscience of sharks, dolphins, humans, and AI, see the slides on the virtual reality of the mind: http://www.jfsowa.com/talks/vrmind.pdf
Research that established the foundations for Kyndi technology:

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
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