Relating Knowledge Graphs To Logic, Language, and the World

John F. Sowa
10 May 2020
Relating KGs to Logic and Language

Abstract: Knowledge graphs are simplified semantic nets, originally designed for information retrieval and question answering. UML diagrams and the Semantic Web have greater precision and expressive power. The DOL standard for distributed ontology, modeling, and specification maps those notations to Common Logic (CL). The new CLIP dialect makes CL more readable. For the RDF and OWL subsets, CLIP is as readable as Turtle. For the full expressive power, CLIP resembles predicate calculus. Tools based on CLIP and DOL can relate knowledge graphs to any language or logic, natural or artificial.

Contents:

1. What is a knowledge graph?
2. DOL for knowledge graphs
3. CLIP for linear and graphic logics
4. Relating KGs to natural languages
5. Metalanguage and metadata
6. Automated and Semi-automated Tools
1. What is a Knowledge Graph?

Knowledge graphs are simplified semantic networks.

- Google introduced KGs in 2012 for question answering.
- DBpedia and other freely available resources provide the data.
- The KGs are stored as triples, and Google added various AI methods for learning and reasoning.

For the Jeopardy challenge, IBM Watson also used DBpedia.*

Watson added a wide range of AI technology: English parsers, question classification, question decomposition, automatic source acquisition and evaluation, entity and relation detection, logical form generation, statistics, machine learning, knowledge representation, and several methods of reasoning.

Goal for the future: Automated and semi-automated tools to make these systems easier to design and implement.

* See https://www.aaai.org/Magazine/Watson/watson.php
Knowledge graphs can represent symbolic models that are directly related to ontology, language, and mental models. Tools for logic and language can relate them to reasoning, action, and the world.
Implementing the Hexagon

The corners of the hexagon represent aspects of knowledge.

1. The world is everything we encounter in space and time.
2. Mental models represent everything we experience or imagine.
3. Symbolic models consist of words related by words to other words.
4. Ontology is a catalog of all the words and whatever they refer to.
5. Reasoning includes all our ways of thinking about anything.
6. Action is what our thinking leads us to do in and on the world.

Natural languages represent conscious knowledge.

- They can represent and relate all six corners of the hexagon.
- Every artificial language, notation, or diagram is a simplified or stylized version of something that could be said in a natural language.
- But the nervous system contains an enormous amount of unconscious knowledge that supports the basic operations of the human body.

Challenge for AI: Design tools to support all the above.
2. DOL for Knowledge Graphs

DOL is a standard for integration and interoperation among distributed ontologies, models, and specifications (OMS). *

- Knowledge graphs (KGs) are as diverse as the many kinds of OMS, and they can benefit from the OMS tools.
- UML and the Semantic Web logics are supported by DOL.
- Tools based on DOL can map KGs to and from those logics.

DOL is formally defined by logic and mathematics.

- Logic is essential for guaranteeing precision.
- DOL uses Common Logic (CL) to define, relate, and integrate heterogeneous OMS.
- But people can still use their familiar diagrams and notations.

* OMG Standard for DOL: Distributed Ontology, Modeling, and Specification Language: [https://www.omg.org/spec/DOL/1.0](https://www.omg.org/spec/DOL/1.0)
Unified Modeling Language (UML)

A family of diagrams for representing database and computer system designs.

Originally specified as informal notations without a precise definition in logic.

The Object Management Group (OMG) standardized formal UML by definitions stated in Common Logic.*

By mapping UML diagrams and SW logics to CL, DOL can facilitate data sharing among applications in any field.

* See https://www.omg.org/spec/FUML/1.4
Mapping UML and the Semantic Web to CL

The DOL standard specifies formal mappings from the Semantic Web logics and UML diagrams to TPTP, CASL, and Common Logic.

Arrows show the mappings from less expressive logics to more expressive logics.

TPTP notation (for Thousands of Problems for Theorem Provers) is a version of many-sorted logic, of which classical first-order logic is a single-sorted subset.

CASL is the logic for HeTS (the Heterogeneous Tool Set), which implements the mappings.
Using DOL With KGs

Usage scenarios for DOL may be adapted to data from KGs. *

- Interoperability between OWL and FOL ontologies
- Module extraction from large ontologies
- Interoperability between closed-world data and open-world metadata
- Verification of rules translating Dublin Core into PROV
- Maintaining different versions of an ontology in languages with different expressivity
- Metadata within OMS repositories
- Modularity of specifications
- Refinement of specifications
- Consistency among UML models of different types
- Refinements between UML models of different types, and their reuse
- Coherent semantics for multi-language models

* See Section 7 of the DOL standard (pp. 33 to 48).
Relating and Integrating Everything

CLIP can relate legacy systems to the latest AI tools,
  • Freely mixing and matching any notations supported by DOL.
  • Anyone may continue to use their favorite notations indefinitely.

Semantic Web annotations may be replaced by CLIP:
  • Any URI, enclosed in quotes, is a valid CLIP name.
  • An annotation that uses the full expressive power of CLIP is written
    `<clip> (one or more CLIP sentences) </clip>`
  • Any annotation written in a Semantic Web logic x may be rewritten
    `<clip logic=x> (one or more CLIP sentences) </clip>`
  • For tools that do not support CLIP, a preprocessor may translate CLIP annotations to the corresponding SW logic.

For integrating legacy systems with AI technology,
  • Any software that is described or specified in any UML or SW notation can take advantage of tools that process CLIP.
Computability and Decidability

The logics for UML and the Semantic Web are decidable.

- But decidability is a property of the problem, not the notation.
- The TPTP tools use syntactic checks to determine which algorithm to use for any particular problem.
- For the same problems, TPTP theorem provers are faster than the tools designed for the restricted logics of the Semantic Web.

Restricting expressive power cannot improve performance. *

- It just makes certain problems impossible to state.
- Natural languages are more expressive than any version of logic.
- But the only people who can state an undecidable sentence are those who have studied advanced logic and mathematics.
- The FOL theorems of *Principia Mathematica* are trivial for TPTP.

CLIP is a dialect of Common Logic that has a direct mapping to and from predicate calculus and the graph notations for logic. The existential graphs (EGs) by C. S. Peirce are the simplest graph logic with the full expressive power of Common Logic.*

EG rules of inference and the methods for mapping EGs to and from CLIP may be adapted to any graph logic.

Design goals for CLIP:

- Immediately readable by anyone who knows predicate calculus.
- As readable as Turtle for the RDF and OWL subsets.
- As readable as any notation for if-then rules.
- Serve as a linearization for UML diagrams.
- Query option: Select (list of names) where (any CLIP sentence).
- Support mappings of KGs ↔ logics and languages.

How to say “A cat is on a mat.”

Gottlob Frege (1879):

Charles Sanders Peirce (1885):

Giuseppe Peano (1895):

Existential graph (1897):

Conceptual graph (1976):

CLIP dialect of Common Logic:

$(\exists x \ y) (\text{Cat}(x) \land \text{On}(x, y) \land \text{Mat}(y))$
Existential Graphs

A line for existence. An oval for negation. Conjunction is implicit.

Existence: —

Negation: 

Relations: Cat- -On- -Under- -With- -Mat

A cat is on a mat: \( \text{Cat—On—Mat} \)

Something is under a mat: \( —\text{Under—Mat} \)

Some cat is not on a mat: \( \text{Cat—On—Mat} \)

Some cat is on something that is not a mat: \( \text{Cat—On—\text{Mat}} \)
The Core CLIP Notation

Core CLIP has a one-to-one mapping to and from EGs.

Existence:  \((\exists x)\) or \((\text{Exists } x)\)

Negation:  \(\neg[ \ ]\)

Relations:  \((\text{Cat } x), (\text{On } x y), (\text{Under } x y), (\text{Mat } y)\)

A cat is on a mat:  \((\exists x y) (\text{Cat } x) (\text{On } x y) (\text{Mat } y)\)

Something is under a mat:  \((\exists x y) (\text{Under } x y) (\text{Mat } y)\)

Some cat is not on a mat:  \((\exists x) (\text{Cat } x) \neg[ (\exists y) (\text{On } x y) (\text{Mat } y) ]\)

Some cat is on something that is not a mat:
\((\exists x y) (\text{Cat } x) (\text{On } x y) \neg[ (\text{Mat } y) ]\)
EGs Without Negation

These examples represent existence and relations:

- A line of identity states that something exists. In CLIP, that is \((\exists x)\).
- Relations in CLIP are represented as (king x), (on x y), (under x y).

Translating the above EGs to CLIP:

\((\exists x)\) (man x). \(\text{There is something, which is a man.}\)

\((\exists x)\) (king x). \(\text{There is something, which is a king.}\)

\((\exists x)\) (man x) (king x). \(\text{There is something, which is a man and a king.}\)

An option that uses a relation name to restrict the quantifier:

\((\exists x \text{'man})\) (king x). \(\text{Some man is a king.}\)
One of Peirce’s Examples

Peirce’s translation to English: “There is a Stagirite who teaches a Macedonian conqueror of the world and who is at once a disciple and an opponent of a philosopher admired by Fathers of the Church.”

A translation to CLIP:

$$\exists x \ y \ z \ (\text{"is a Stagirite" } x) \ (\text{teaches } x \ y) \ (\text{"is a Macedonian" } y)$$
$$\ (\text{"conquers the world" } y) \ (\text{"is a disciple of" } x \ z) \ (\text{"is an opponent of" } x \ z)$$
$$\ (\text{“is a philosopher admired by church fathers” } z).$$

Without negation, CLIP can represent the content of a relational database or the graph databases of the Semantic Web.
Nested Ovals

An oval nested in an even number of negations is unshaded.

- Since a double negation is positive, evenly nested areas are positive.
- A nest of two ovals represents an if-then statement.

With two or more negations, an EG may be translated to CLIP or to English in several equivalent ways:

- $\sim[\exists x (\text{man } x) \sim(\text{king } x) ]$. \hspace{1cm} It’s false that there is a man who is not a king.
- $[\text{If } (\exists x ' \text{man}) [\text{Then } (\text{king } x) ]].$ \hspace{1cm} If there is a man, then he is a king.
- $(\forall x ' \text{man}) (\text{king } x).$ \hspace{1cm} Every man $x$ is a king.
Representing Functions

An example in mathematical notation: \( y = 7 \div (x + 1) + \sqrt{7}. \)

In EGs, a function may be represented as a relation with an arrow for its last line of identity. The four functions may be named \(+1, \div, \sqrt{}, +.\)

A direct mapping of the EG to CLIP:

\[
(\exists (x \ y \ u \ v \ w) \text{real}) \ (+1 \ x \rightarrow u) \ (\div 7 \ u \rightarrow v) \ (\sqrt{7} \rightarrow w) \ (+ \ v \ w \rightarrow y).
\]

Another option eliminates the need for the names \(u, v, w:\)

\[
(\exists (x \ y) \text{real}) \ ( = y \ (+ \ (\div 7 \ (+1 \ x)) \ (\sqrt{7})))).\]
Quantifying Over Functions and Relations

CLIP allows quantified names to refer to functions and relations.

English:  *Bob and Sue are related.*
CLIP:     (related Bob Sue).

English:  *There is a familial relation between Bob and Sue.*
CLIP:     (∃ r) (familial r) (relation r) (r Bob Sue).

English:  *Every numeric function maps numbers to numbers.*
CLIP:     (∀ f) (numeric f) (function f)
          (∀ x y) [If (f x → y) [Then (number x) (number y) ]].

Literal translation of CLIP to English:  *For any numeric function f and any x and y, if f maps x to y, then x is a number and y is a number.*

Note: Higher-order logic with a hierarchy of infinite sets is extremely inefficient for computer processing. But Common Logic uses a version of second-order logic that is as efficient as FOL.
EG lines connected by a ligature refer to the same thing.

- In the EG on the left, two lines, which may be named x and y, are connected by a ligature inside the negated area.
- An EG ligature is represented by a CLIP identity, (= x y). In this example, the ligature is negated by shading, and the CLIP is (∃ x y) ~(= x y).
- To emphasize the ligature, the EG in the middle has an optional dot on the connection. Both EGs assert “There exist two different things.”
- The EG on the right asserts “There exits God x, and if there exists God y, then x is y.” In CLIP, (∃ x'God) [If (∃ y'God) [Then (= x y) ] ].

An EG ligature or a CLIP identity may relate multiple names: (= JQP "John Q. Public" "Jack Public" "http://somewebsite.com/emp_no/87926").
4. Relating KGs to Natural Languages

For computers, informal mappings must be formalized.
- Informal mappings to natural languages (NLs) are OK for humans.
- Informal mappings to RDF and OWL are OK for simple KGs.
- But anything a computer does is formal.

Discourse Representation Theory specifies a subset of NLs.*
- DRT is widely used for natural language processing (NLP).
- Discourse representation structures (DRSs) support full FOL.
- Every DRS has a precise mapping to and from CLIP.

Semi-automated translation of NLs and KGs to and from CLIP.
- Computer translation of NL → CLIP is error prone.
- Computer translation of CLIP → NL is precise, but verbose.
- Human translation depends on the human.
- Simpler and more reliable: Human-aided computer translation.

Mapping DRS to CLIP

Peirce and Kamp independently chose isomorphic structures.

- Peirce chose nested ovals for EG with lines to show references.
- Kamp chose boxes for DRS with variables to show references.
- But the boxes and ovals are isomorphic: they have the same choice of primitive operators, and they support equivalent operations.

Example: If a farmer owns a donkey, then he beats it.

The EG and DRS can be represented by exactly the same CLIP:
\[
\text{If } (\exists x \ y) (\text{farmer } x) (\text{donkey } y) (\text{owns } x \ y) \ [\text{Then } (\text{beats } x \ y) \ ]
\]
Mapping Language to Logic

Hans Kamp observed that the features of predicate calculus do not have a direct mapping to and from natural languages.

Pronouns can cross sentence boundaries, but variables cannot.

- Example: *Pedro is a farmer. He owns a donkey.*
- PC: $\exists x (\text{Pedro}(x) \land \text{farmer}(x))$. $\exists y \exists z (\text{owns}(y,z) \land \text{donkey}(z))$.
- There is no operator that can relate $x$ and $y$ in different formulas.

In English, quantifiers in the if-clause govern the then-clause.

- Example: *If a farmer owns a donkey, then he beats it.*
- But in predicate calculus, the quantifiers must be moved to the front.
- CLIP supports both options: English-like and PC-like.
  
  If $(\exists x y) \ (\text{farmer} \ x) \ (\text{donkey} \ y) \ (\text{owns} \ x \ y) \ [\text{Then} \ (\text{beats} \ x \ y) \ ] \ ]$.
  
  $(\forall x y) \ If \ (\text{farmer} \ x) \ (\text{donkey} \ y) \ (\text{owns} \ x \ y) \ [\text{Then} \ (\text{beats} \ x \ y) \ ] \ ]$.

Note: Proper names are rarely unique identifiers. Both Kamp and Peirce represented names by monadic relations.

EG:

DRS:

CLIP:

$(\exists x y z) \ (\text{Jones}(x)) \ (\text{Smith}(y)) \ (\text{Cooper}(z))$

$[\text{Or} \ [ (\exists u) \ (\text{owns}(x,u)) \ \text{"book on semantics"}(u)]$

$[ (\exists v) \ (\text{owns}(y,v)) \ \text{"book on logic"}(v)]$

$[ (\exists w) \ (\text{owns}(z,w)) \ \text{"book on unicorns"}(w)]$.
EGs With Negation

A shaded oval states that the nested graph or subgraph is false:

- When a line of identity is extended into an oval enclosure, existence is declared in the area that contains the outermost point of the line.
- In CLIP, a shaded oval is represented by a tilde ~ and square brackets [ ].

Translations to CLIP (with two options for the EG on the right):

- ~[ (∃x) (man x) ].  \text{It's false that there is a man.}
- (∃x) ~(man x).  \text{There is something which is not a man.}
- (∃x) (king x) ~(man x).  \text{There is something which is a king and not a man.}
- (∃x x'king) ~(man x).  \text{Some king is not a man.}
Translating the Word *is* to Logic

Three different translations in English or CLIP:

- **Existence:**  *There is x.*  $\leftrightarrow (\exists x)$
- **Predication:**  *x is a cat.*  $\leftrightarrow (\text{Cat } x)$
- **Identity:**  *x is y.*  $\leftrightarrow (= x y)$

Do these three translations imply that English is ambiguous?

Or is the syntax of linear notations too complex?

In EGs, all three uses of the word *is* map to a line of identity:

- **Existence:**  *There is x.*  $\leftrightarrow$ —
- **Predication:**  *x is a cat.*  $\leftrightarrow$ —Cat
- **Identity:**  *x is y.*  $\leftrightarrow$ —— (a ligature of two lines)

As Peirce said, EGs are more iconic than predicate calculus: they show relationships more clearly and directly.
Linking Existential Quantifiers

To relate existential quantifiers in different statements, EGs (left) and DRSs (right) support equivalent operations:

After connecting EG lines or merging DRS boxes,

CLIP: \((\exists x\ y\ z)\ (\text{Pedro}\ x)\ (\text{farmer}\ x)\ (= x y)\ (\text{owns}\ y\ z)\ (\text{donkey}\ z)\).

The names x, y, and z make DRS and CLIP more verbose than EGs.
Metalanguage is language about language, natural or artificial.

- To define semantics, Tarski (1933) used logic as a metalanguage for defining the truth value of any statement in logic.
- Like logic, KGs can state information or talk about information.
- A logic for KGs should also support metalevel KGs.

The IKL extension to Common Logic supports metalanguage. *

- IKL enables CLIP to comment on anything expressed in CLIP.
- It can represent metadata about the sources and reliability of data.
- It can support reasoning about metaphor, modality, and the issues of vague, fuzzy, missing, erroneous, or fraudulent information.

Any Unicode strings may be used for names. Metacomments in CLIP may even be expressed by emojis.

* For the IKL documents, see http://jfsowa.com/ikl/.
Metalanguage in Existential Graphs

A metalevel EG by Peirce (1898):

Peirce’s English: “That you are a good girl is much to be wished.”

A shaded oval negates the nested EG. Without shading, the EG expresses a proposition that is neither asserted nor negated.

The same proposition in CLIP: [ "You are a good girl" ].

History: From 1898 to 1914, Peirce wrote extensively about metalanguage, modality, and intentionality. Those writings had a strong influence on logicians, philosophers, linguists, and AI researchers. For references, see http://jfsowa.com/pubs/5qelogic.pdf
Metalanguage About Situations

The drawing on the right may be interpreted in three ways.

1. Actual: Pierre is thinking of Marie, who is thinking of him.
2. Modal: Pierre is thinking of Marie, who may be thinking of him.
3. Intentional: Pierre hopes that Marie is thinking of him.

In the second clause of #1, the verb is implies that Pierre’s thought is true.

In #2, the verb may implies that his thought is a possible proposition.

In #3, the object of the verb hopes is a situation Pierre intends in some way.
Propositions and Situations

Three ways of relating a proposition $p$ to a situation $s$.

Actual: $p$ is true or false about $s$.
Modal: $p$ is related to $s$ in some manner or mode $m$.
Intentional: Some agent $x$ relates $p$ to $s$ for some reason $r$.

Relating an agent $x$, a proposition $p$, and a situation $s$.

Actual: $p$ is expressed by a CLIP sentence that describes $s$.
Modal: $m$ is represented by a dyadic relation ($m \ p \ s$).
Intentional: $r$ is represented by a triadic relation ($r \ x \ p \ s$).

CLIP notation for propositions and situations.

Propositions enclosed in brackets without a tilde: $[ \ (any \ CLIP \ sentence) \ ]$.
Definition: $(\forall \ s'situation) \ (\exists \ p) \ (= \ p \ [ \ (some \ CLIP \ sentence) \ ]) \ (dscr \ p \ s)$.
With this definition, the keyword 'Situation' may be placed inside the brackets to show the intended use: $[Situation \ (some \ CLIP \ sentence) \ ]$. 
Examples in CLIP

Peirce’s example of 1898 represents an intended situation.

English:  That you are a good girl is much to be wished [by someone].
CLIP:  ("is much to be wished" [Situation "You are a good girl"]).
The relation implies an agent x, and the situation implies a description p.

English and CLIP for the sentences about Pierre.

English:  Pierre is thinking of Marie, who is thinking of him.
CLIP:  (thinkingOf Pierre Marie) (thinkingOf Marie Pierre).

English:  Pierre is thinking of Marie, who may be thinking of him.
CLIP:  (thinkingOf Pierre Marie) (possible [ (thinkingOf Marie Pierre) ]).

English:  Pierre hopes that Marie is thinking of him.
CLIP:  (hopesFor Pierre [Situation (thinkingOf Marie Pierre)]).

IKL does not support modal logic, but metalanguage can be used to define modality in terms of laws and facts.*

* See http://jfsowa.com/pubs/worlds.pdf
Conceptual graphs (CGs) express the same logic as EGs, but they are designed to represent the details of NL semantics. *

English: “If a farmer owns a donkey, then he beats it.”

CLIP: [If (∃ x 'farmer y 'own z 'donkey) (Expr y x) (Thme y z) [Then (∃ w 'beat) (Agnt w x) (Ptnt w z) ]].

Unlike EGs, quantifiers in CGs are represented by boxes, not lines.
Names may refer to concept boxes that represent verbs.
The *semantic or thematic roles* used in linguistics relate verbs to nouns: experiencer (Expr), theme (Thme), agent (Agnt), and patient (Ptnt).

* See “From EGs to CGs”, [http://jfsowa.com/pubs/eg2cg.pdf](http://jfsowa.com/pubs/eg2cg.pdf)
The CGs above show two of the three interpretations of the sentence *Tom believes that Mary wants to marry a sailor*:

- *Tom believes a proposition that Mary wants a situation in which there exists a sailor whom she marries.*
- *There is a sailor, and Tom believes that Mary wants to marry him.*

For the third interpretation, the concept box for *Sailor* would be moved to the area of the proposition:

- *Tom believes that there is a sailor whom Mary wants to marry.*
Three Interpretations of Tom’s Belief

First, the sailor exists only in the situation that Mary wants:

English: Tom believes a proposition that Mary wants a situation in which there exists a sailor whom she marries.

CLIP: (Person Tom) (∃ x'Believe) (Expr x Tom) (Thme x [ (Person Mary) (∃ y'Want) (Expr y Mary) (Thme y [Situation (∃ z'Marry) (∃ w'Sailor) (Agnt z Mary) (Thme z w) ]])]).

Second, the sailor exists in reality:

English: There is a sailor, and Tom believes that Mary wants to marry him.

CLIP: (∃ w'Sailor) (Person Tom) (∃ x'Believe) (Expr x Tom) (Thme x [ (Person Mary) (∃ y'Want) (Expr y Mary) (Thme y [Situation (∃ z'Marry) (Agnt z Mary) (Thme z w) ]])]).

Third, the sailor exists in the proposition that Tom believes:

English: Tom believes that there is a sailor whom Mary wants to marry.

CLIP: (Person Tom) (∃ x'Believe) (Expr x Tom) (Thme x [ (∃ w'Sailor) (Person Mary) (∃ y'Want) (Expr y Mary) (Thme y [Situation (∃ z'Marry) (Agnt z Mary) (Thme z w) ]])].
6. Automated and Semi-automated Tools

Computers should ask humans for help and explanations.

- The experts in any field rarely have a PhD in computer science. Even computer scientists don't know the jargon of every system.
- Computers should accept any language or notation people prefer, and they should read documents without requiring prior annotations.
- If a computer can't understand some text, it should ask people for help. People should answer in their own language.
- Computers may annotate texts, but human assistance is necessary when a computer is uncertain about the interpretation.

Analogies can support informal, case-based reasoning:

- Cognitive Memory can efficiently find analogies in Big Data.

Formal reasoning is based on a disciplined use of analogy:

- Theorem provers use a kind of analogy called unification.
Cognitive Memory (CM)

CM is an associative memory for large volumes of graphs. *

- Approximate pattern matching for analogies and metaphors.
- Associative storage and retrieval of graphs in log(N) time.
- Precise pattern matching (unification) 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) part of a new case with some rule or axiom.
- Abduction: Form a hypothesis based on aspects of similar cases.

KGs for Help and Explanations

CM has been used to support NL dialog and explanations.

- CM uses CLIP as the basic representation.
- But CM can also use any KGs that can be mapped to CLIP.
- That includes any KGs that use UML or Semantic Web notations.
- Tools for CLIP and IKL can support natural language and logic. *

Recommendation for a new generation of development tools:

- Integrate all systems, including legacy systems, with logic-based methodologies.
- Enable subject-matter experts to review, update, and extend their knowledge bases with little or no assistance from IT specialists.
- Provide tools that support collaboration, review, and testing by people with different levels and kinds of expertise.
- An open-ended variety of software, including NNs, may be used.

* Slides about natural logic:  http://www.jfsowa.com/talks/natlog.pdf
Formal Concept Analysis (FCA)

A theory and tools for semi-automated ontology design:
- **Theory.** Define a minimal lattice that shows all inheritance paths among a set of concept types, each defined by a list of attributes.
- **Algorithms.** Efficient ways for computing a minimal lattice from a list of terms and defining features.

Applications:
- **Ontology development and alignment; classification methods; machine learning; defining concepts used in other logics.**
- **FCA tools are often used to check whether ontologies specified in OWL and other notations are consistent.**
- **They can also be used to detect inconsistencies among two or more independently developed ontologies.**

The FCA Homepage: [http://www.upriss.org.uk/fca/fca.html](http://www.upriss.org.uk/fca/fca.html)
For deriving lattices from lexical resources: [http://www.upriss.org.uk/papers/jucs04.pdf](http://www.upriss.org.uk/papers/jucs04.pdf)
Generating Lattices Automatically

FCA tools used the data in Roget’s Thesaurus to generate this lattice for the word 'happy' and its hypernyms (supertypes).

To generate this or similar lattices, enter 'happy' or any other word at the web site [http://www.ketlab.org.uk/roget.html](http://www.ketlab.org.uk/roget.html)
Classifying Resources by Purpose

FCA tools may use a variety of criteria for classification.

- For ontology, the usual criterion is type/subtype.
- But a person who asks a question has some purpose in mind.
- The lattice above classifies resources by purpose, not type.
Advice From Two Logicians and a Poet

Alfred North Whitehead:

“Human knowledge is a process of approximation. In the focus of experience, there is comparative clarity. But the discrimination of this clarity leads into the penumbral background. There are always questions left over. The problem is to discriminate exactly what we know vaguely.”

Charles Sanders Peirce:

“It is easy to speak with precision upon a general theme. Only, one must commonly surrender all ambition to be certain. It is equally easy to be certain. One has only to be sufficiently vague. It is not so difficult to be pretty precise and fairly certain at once about a very narrow subject.”

Alfred North Whitehead:

“We must be systematic, but we should keep our systems open.”

Robert Frost:

“I’ve often said that every poem solves something for me in life. I go so far as to say that every poem is a momentary stay against the confusion of the world... We rise out of disorder into order. And the poems I make are little bits of order.”

(To make the comparison, replace every occurrence of poem with theory.)
Engineers: “All models are wrong, but some are useful.”
  • Discrete symbolic models can be clear, sharp, and precise.
  • But the world is continuous, disordered, and fuzzy.

Natural languages are flexible. They can adapt to anything.
  • They can be as vague or precise as the situation requires.
  • KG tools should be flexible: Detailed levels must be precise, but the top-level ontology must be “sufficiently vague.”
Perception and classification take one second or less.
  • Artificial NNs are very good for learning and recognizing patterns.
  • Ontology tools are good for classifying what is recognized.
  • Result: “gut feel” – rapid perception-action-action responses.

Analysis, planning, and explanation take much more time.
  • AI requires many more tools than just NNs + OWL.
Related Readings


For other references, see the general bibliography, http://www.jfsowa.com/bib.htm