This post is about one of the main difficulties of ontology identification - translating between human ontology and AI ontology. See also unsupervised translation, radical translation and radical interpretation.

I want to make the following argument:

1. We can classify all interpretable types of computations.
2. We could, in theory, formally define those types and the metrics of interpretability.
3. Therefore we could, in theory, write an algorithm which finds the most interpretable translation of any computation.

That would be an important theoretical result even if running the translation algorithm requires an unphysically large finite computer.

Making the argument requires a universal high-level theory of semantics. "Universal" means that it can explain the meaning of arbitrary abstractions. "High-level" means that the explanations are interpretable by humans. It's easy to find a theory which is either universal or high-level, but not both at the same time.[1]

More Research Context

90% of this post are examples of my abstraction typology. Only 10% is fleshing out the whole argument. Why? Because...

There's an entire research direction trying to understand human abstractions. The researchers are John Wentworth & David Lorell & Gretta Duleba, Thane Rutheins, Jan Kulveit, Sam Eisenstat.

Most of them already consider likely much stronger claims than the ones I'm arguing for. They consider it likely that *all* useful abstraction types are interpretable; that humans have already discovered a big chunk of *all* useful abstraction types; that abstraction types of different agents tend to converge.

They are stuck somewhere around understanding nouns, but my typology covers much more. So I expect that it's the most important part of this post. If people will agree that it's plausible, a lot will automatically follow. That's why examples are 90% of the post.

Also, John said that a whole crapton of examples might be useful.

---

[1] Let's say our theory of semantics is "the meaning of an abstraction is the set of all inferences it is a part of" (see inferential role semantics, meaning holism). That's a universal theory, but not high-level.

Natural abstraction framework is supposed to be both universal and high-level. But right now it's not high-level and/or not universal (many adjective, verb and noun types are not figured out).

1. Framework

Terminology

The core term I'm gonna use is "computation". By it I mean something like "a well-defined process unfolding in time", "a process which could be described by a program", "a process which could be simulated by a Turing machine" or "a process which maps inputs to outputs". Doesn't matter if the process is real or imaginary; doesn't matter how we map real things to computations.

Now there's the process of computing something, the computational trace, the program describing the process, the input/output... for convenience I'm gonna call all of that "computation", to avoid juggling many terms.

1.1. DCI Ontology

We're surrounded by computations hidden within other computations. For example, Mario (the character) is hidden within Super Mario Bros. (the program) hidden within transistors hidden within the quantum foam hidden within your sensory input stream. To be useful, those hidden computations have to be detectable and influential. If Mario were undetectable and weren't affecting anything, it would be useless to think about him. But he's a cultural icon!

So it makes sense to think about the following triad:

1. **Detection computation.** Detects the content computation within an exterior computation. E.g. detects Mario (the character) within Super Mario Bros. (the program).
2. **Content computation.** Computes something. E.g. "what could Mario do in this situation?"
3. **Influence computation.** Computes how the content computation constrains the exterior computation. E.g. how Mario's actions affect the output of Super Mario Bros.

Same thing in other words:

1. **Detection computation.** Transforms A into B.
2. **Content computation.** Transforms B into C.
3. **Influence computation.** Transforms C into A'.

A can be interpreted as the main problem we're solving and *A'* can be interpreted as a prediction about the solution.

An instance of the above triad is called a **DCI ontology**. A DCI ontology can contain entire sets of detection / content / influence computations.

Examples of DCI ontologies:

- "((2 + 4) · (2 + 4)) + x = y" is the exterior computation. Addition (transforming "(2 + 4) · (2 + 4)" into "6 · 6") is the detection computation. Multiplication table (transforming "6 · 6" into "36") is the content computation. Calculus (transforming "36 + x" into "x") is the influence computation, telling that the value of "6 · 6" plays an exceedingly small role in the value of *y* as *x* grows to infinity.
- The knapsack problem is the exterior computation. A Karp reduction to sudoku is the detection computation. An algorithm for solving sudoku is the content computation. No influence computation is needed, because the solutions to sudoku are the solutions to the knapsack problem.
- A chess position is the exterior computation. The rules of how individual pieces move and interact is the content computation. Heuristics - material balance, development/mobility, center control, king safety - are the influence computations. No detection computation is needed, because a chess position already contains the information about where/what the pieces are.
- A text is the exterior computation. Your ability to recognize words is the detection computation. Your ability to understand the meanings of individual words is the content computation. Your understanding of context (e.g. Gricean implicature and narrative techniques), allowing to decide which meanings matter more/less, is the influence computation.

A couple more examples:

- A graph is the exterior computation. The calculation of *g(n)* (the cost of the cheapest path from the starting node to *n*) is the content computation. The calculation of *h(n)* (a heuristic function that estimates the cost of the cheapest path from *n* to the goal) is the influence computation. I'm talking about "f(n) = g(n) + h(n)" in the A star algorithm. No detection computation is needed.
- Simple properties of natural numbers are the exterior computation. Primality tests are the detection computations. The definition of prime numbers is the content computation. Stuff like Euclid's theorem, the fundamental theorem of arithmetic and the prime number theorem are the influence computations - telling how primes influence the entirety of natural numbers.

It's not directly implied by the definition, but most of the time by "influence computations" I mean "shortcuts for predicting the outcome of a more complicated computation".

Technically, craving a computation into detection / content / influence is completely arbitrary. But some cravings are more meaningful than others. I chose this ontology only because it's natural for talking about computations inside computations and because it - *with additional specifications* (see §2.1) - gives a nice explanations of human abstractions.

The next two sections will be very confusing, but they'll become clearer after we apply my framework to human abstractions (in §2-4).

1.2. Expanded Ontology

If we don't restrict what "content" means, then anything can be described in terms of A→B, B→C and C→A'.

But if we do restrict what "content" means, then we might need to consider more types of computations:

1. **A→B.** Detection.
2. **A→C.** Direct prediction of content.
3. **A→A'.** Direct prediction of the exterior.
4. **B→C.** Content.
5. **B→A'.** Detection influencing the exterior.
6. **C→B.** Content influencing detection.
7. **C→A'.** Content influencing the exterior.

For any computation there also can be a *meta-level computation* which helps to predict or create it. For example, a *meta-level content computation* helps to predict the output of B→C or create B→C.

Oftentimes it's convenient to just call everything except (meta-level) A→B and (meta-level) B→C *"(meta-level) influence computations"*.

Finally, there are *micro-* and *macro-* computations. *Micro-computations* are small parts of detection / content / influence computations. Detection / content / influence computations are small parts of *macro-computations*.

For example, let's say predicting an individual human (Bob) is our computation. Then predicting agents in general is a meta-level computation. While predicting a single atom within Bob is a mirco-computation (even if we can predict Bob without modelling individual atoms). While predicting Bob's country is a macro-computation (even if we can predict the country without modelling individual humans).

Micro- and macro- computations can be meta-level too. So in principle we can talk about stuff like a *meta-level micro-computation in B→C* (a computation predicting the output of a small part of B→C). But we won't need to get this detailed most of the time.

1.3. Completeness

Why am I introducing meta-level and micro/macro computations? Why stop there? Why not introduce more stuff?

From a philosophical POV, my framework tries to answer the following question —

"What does a set of computations (let's call it an 'ontology') 'see' inside and around itself?" or "what *could* it 'see' inside and around itself, if it had more cognitive power?"

Detection / content / influence computations describe what an ontology "sees" directly. Meta-level computations generalize the ontology. While micro/macro computations describe alternative ontologies. Like in the example about Bob - we can use a normal ontology (= model him as a person) or a generalized ontology (= model him as an agent). Or we can use alternative ontologies (= model his atoms or the geopolitical forces affecting him).

So I claim that the stuff we already introduced is enough to describe everything an ontology could "see" inside and around itself.

2-3. Semantics

2.1. Human Ontology

Let's say we're in a non-tropical forest. We want to predict sensory patterns with well-defined spatiotemporal behavior[2] (e.g. animals, plants, fungi, rocks, rivers, clouds, animals, rain, the day/night cycle) and predict their interactions. Predictions of individual patterns are the *content computations*. Predictions of interactions are the *influence computations*. Interactions themselves are the *exterior computation*.

When you notice a bear, it's a *detection computation*. When you predict the movement of the bear, it's a *content computation*. When you predict a high likelihood of death, it's an *influence computation*.

(Most of the time by an "influence computation" I mean "a shortcut for predicting the outcome of a complicated interaction between sensory patterns" or, put simply, "a shortcut for predicting a complicated situation".)

After fixing what we mean by "detection / content / influence" and "exterior", we can analyze what makes different human abstractions useful. Let's call this setup **human!DCI ontology** or just **human!DCI**.

2.2. Brief Analysis

Let's analyze an adjective, "big".

---

[2] By "well-defined behavior" I mean relatively simple behavior which isn't too varied. "Frog" is a sensory pattern with well-defined behavior. "Green spot" isn't a sensory pattern with well-defined behavior, because there are many green things with completely different behavior.

"Big" is useful for many *detection computations*. A big animal/mountain/river is easier to notice. A big animal/mountain/river is also easier to *recognize*, because "big" is a rare quality. Same logic applies to many other adjectives - e.g. "loud", "shiny", "smelly", "orange", "long".

"Big" is also useful for many *content computations*. I.e. predicting individual sensory patterns.

- Trees and bear cubs grow very big.
- A big tree/animal/mountain/river usually won't become smaller.
- A big animal won't go through a narrow space, unless it's flexible and motivated.

Many other adjectives (e.g. "fast", "aggressive", "carnivorous", "furry", "underground") are about content too.

"Big" is also useful for many *influence computations*. I.e. quick predictions of complicated situations.

- A big animal is hard to defeat. Just meeting it might mean that you're already dead, no matter what you do.
- A big mountain/river is hard to cross, no matter what you do.
- A big forest fire is impossible to put out, no matter what you do.

Big things are influential in different ways, that's why "big" became synonymous with "important". Many other adjectives (e.g. "aggressive", "smart", "trapped", "hungry", "sick") are influential too.

Similar analysis can be done for verbs and nouns. Even for prepositions.

- *Running* animals are easier to notice than stationary or sneaking ones. I.e. "run" is useful for detection.
- Birds can *fly*. I.e. "fly" is a part of the content of birds.
- If you get *poisoned*, you suffer later no matter what else happens, unless you get a cure. I.e. "poison" is an influential verb. Some other influential verbs are die, kill, nuke, trigger, grow, learn, convince, win, catch, control, want, fear, sleep.
- *Footprints* are useful for detecting animals.
- If you see a *bear*, you might be doomed no matter what you do. I.e. "bear" is an influential noun. Some other influential nouns are fire, tornado, night, darkness, stranger, lover, leader, god, shit (something bad or unimportant), maze, paradox, home, fate.
- Hoof fungus grows *on* trees. I.e. "on" can be a part of detection computations (helping to find hoof fungus) and content computations (describing how hoof fungus behaves).
- If wolves are *around* you, you're probably trapped and doomed; if unscalable walls are *around* you, you're probably trapped and doomed. I.e. "around" can be an influential preposition.

2.3. Detailed Analysis

Abstractions like "the color of fox fur", "the color of water", "the color of grass" are useful for detecting sensory patterns with well-defined spatiotemporal behavior (i.e. foxes, water, grass).

But what about abstractions like "the orange color in general", "the blue color in general", "the green color in general"? They are much worse at detecting sensory patterns with well-defined behavior (grass and crocodiles are both "green", yet their behavior is completely different). But they are still useful because, often enough, patches of different color correspond to sensory patterns with different behavior. If you see two patches of different color, you can predict that those patches will behave differently. This means abstract colors help to predict detection computations.[3] I.e. abstract colors help *meta-level detection computations*. Ditto abstractions like "texture", "shape", "size", "change", "salience".

Specific colors (like "the color of grass") and abstract colors (like "the green color in general") are both abstractions over sensory input, but the latter is useful for meta-level reasons. Meanwhile "salience" is a meta-level abstraction useful for meta-level reasons.

Now, consider abstractions like "object", "property", "action", "behavior". Why are they useful if they are completely vacuous? They are useful because of statistical facts like the following:

- Many sensory patterns have object permanence.
- Many sensory patterns have only a couple of most important behaviors.
- Many sensory patterns have only a couple of most important properties.
- etc.

Those statistical facts help to predict and create content computations (i.e. help to analyze old and novel objects and processes). They help *meta-level content computations*.

Next, consider abstractions like "cause/effect", "event", "outcome", "situation", "condition". And "object", "property", "action", "change" again. Why are they useful? They are useful because of statistical facts like the following

- Usually an interaction between sensory patterns can be decomposed into just a couple of main changes / causes & effects / objects / conditions / properties.
- Usually the end of an interaction between sensory patterns (i.e. the end of an action/event/situation) can be easily recognized. Which is useful for stuff like quiescence search.
- The biggest changes in sensory patterns are usually very important, no matter what causes the changes (e.g. noticing a big animal, losing your vision, entering a new area, a tree falling, a geyser erupting or lighting striking).
- etc.

[3] Imagine a world where everything was covered in patches of random color. In this world, knowing abstract colors (red, orange, yellow, green, blue, violet, etc.) would be much less useful.

Those statistical facts help to predict influence computations (i.e. help to analyze old and novel complicated situations). They help *meta-level influence computations*.[4]

3.1. Interpretability

Here are four metrics of interpretability of a computation:

1. **Locality.** How many situations (= inputs) do you need to analyze to understand how the computation works? How much does the output of the computation vary on different inputs? The less, the better.
2. **Generality.** How general is the computation, i.e. in how many situations does it apply? The more, the better.
3. **Simplicity.** How simple/short is the computation?
4. **Genericness.** Is the computation similar to many others or is it unique? The former is better. Because it allows us to understand what the computation means by analogy.

Those metrics should be formalizable in principle.

The types of computations described in §2.2/2.3 are local, general, simple and generic.

3.2.

I think the analysis in §2.2/2.3 is

...plausible. It can't be too false - human abstractions *do* have detection/content/influence components and humans *do* experience them and we *can* get at least rough estimates of those components.

...logically complete. It's hard to imagine any other interpretable type of computation.

...universal. I.e. analyzes human abstractions using a very abstract language, generalizable to any type of computation (see §1.1).

Plausibility, completeness and universality are evidence that we've classified "all interpretable types of computations".[5]

---

[4] Note that e.g. both "big" and "property" help meta-level influence computations, the latter is just more general.

[5] Remember that human!DCI ontology was defined for a specific goal (predicting interactions of sensory patterns with well-defined behavior). Would choosing a different goal lead to a completely different human ontology?

I think not. The goal we've chosen is general enough to subsume most other goals. Also, empirically it just doesn't seem like human abstractions are dramatically sensitive to goals (I mean among humans - humans with different goals don't end up with completely different ontologies).

I'm not claiming that the analysis in §2.2/2.3 is fully true. But I am claiming that human abstractions are useful *and* interpretable *only to the extent to which an analysis like that is true*.

3.3. Automatic Translation

So here's my argument:

1. The setup above (human!DCI) classifies all interpretable types of computations.
2. We could, in theory, formally define those types (i.e. define what "detection" / "content" / "influence" mean in human!DCI) and the metrics of interpretability (§3.1).
3. Therefore we could, in theory, write an algorithm which translates any computation (predicting the same sensory data as human!DCI) into human!DCI and finds the most interpretable translation.

4. More Semantics

4.1

Mathematical abstractions are *micro-computations* (§1.2) in human!DCI. Most individual mathematical abstractions are both too abstract and too specific to be useful. But they combine into useful theorems.

For example, consider the mathematical definition of a set. It's very abstract, but has a bunch of specific quirks:

- Sets can't have repeating elements. Sets can't change in time. It doesn't matter how you create a set, only its elements matter. So set theory won't deal elegantly with sets like "the current president of America", "the number of living people", "Alice's favorite colors", etc.
- There's a difference between {apple} and {{apple}}, as well as {nothing} and {{nothing}}. You can generate uncountable infinities from nothing.
- A collection of things existing together somewhere is often not a set. "Berry" and "herb" exist together inside {apple, berry, herb}, but {apple, berry, herb} doesn't contain {berry, herb}.

Those quirks are not very intuitive and *not even useful* (in human!DCI). But they help to prove theorems. And some of the theorems *are* useful in human!DCI.

In human!DCI, the most abstract micro-computations are universal languages (like math) for describing arbitrary abstractions. While the most abstract *meta-level micro-computations* are about guides for developing those languages and their limitations/capabilities. Gödel's

incompleteness theorems, undecidability of the halting problem, Tarski's undefinability theorem, Rice's theorem, Turing completeness, complexity classes, information theory, category theory are examples of abstractions useful for meta-level micro-computations.

In human!DCI, *macro-computations* are about co-occurring high-level abstractions. Polysemantic concepts and complex fuzzy concepts (such as "friendship", "love", "justice", "aesthetics") are the most abstract macro-computations which we understand at least to some degree. While the most abstract *meta-level macro-computations* are about emergence laws which make some co-occurring abstractions more frequent or more important. Systems theory, cybernetics, chaos theory, selection theorems, subagent theory, shard theory, Critch's boundaries, aesthetics, some theories of semantics (like NAH or this very post) are all attempts to study emergence in the abstract. Intelligence itself is a (meta-level) macro-computation, because it manipulates large numbers of abstractions at once.

4.2. Polysemy

I think polysemantic concepts are clusters of abstractions, connected by strong and weak implications. The individual abstractions and the whole cluster are useful for meta-level influence computations.

I'll analyze "run", "set" and "put" - the most complex verbs in English (TM).

"Run" (wiktionary) is a cluster of approximately the following abstractions:

- Motion, spread, increase, extension in a direction.
- High speed, hurry, urgency, busyness. Free movement (or the opposite - movement through resistance, repeating movement).
- Important entrance / exit / approach. Time limit, temporality. Challenge.
- Something fundamental, global, fixed. E.g. job, functionality, society, (dis)organization in general, family, home, landscape.
- A homogeneous quantity.

Instances of "run" tend to combine multiple of those abstractions. For example, if we say that a mill is "running", we combine ~6 abstractions. Because the mill's *job/functionality is to use (fast) repetitive motion to produce a homogeneous quantity*.

The abstractions are connected by weak and strong implications. For example, many jobs imply repetitive motion which implies repetitive production of a homogeneous quantity; many challenges imply a time limit; many things require a homogeneous quantity (like fuel or food) which disappears with time, implying a time limit; spread is a type of movement; repeated movement is a type of homogeneous quantity; busyness is partially synonymous with a job; etc.

"run" examples

Approximate meanings of "run":

1. "Run into battle", "run and fetch the doctor", "run to mother at every little difficulty", "a run on a bank", "one of the prisoners tried to make a run for it" all mean *"make a hasty, urgent, important approach + have a challenge/problem"*.
2. "Run in the race" means *"compete in a time-limited challenge about fast, hasty motion"*. "Run for mayor" means *"compete in a busy time-limited challenge to get a job"*. "The cops ran the thief down in an alley" means *"cops completed a (time-limited) challenge about fast hasty motion"*.
3. "Give me a rundown" means *"enter the challenge of explaining the information in a fast way"* or *"give me a quantity (= an item-by-item summary) for the challenge of fast information comprehension"*.
4. "The train runs between New York and Washington", "the buses are running late", "a rope runs through the pulley", "a swiftly running grindstone", "file drawers running on ball bearings" all mean *"X has a job about (fast) repetitive motion"*.
5. "Dogs that run deer" means *"dogs have a job about fast, hasty motion"*. "Ran the horse through the fields" means *"made the horse do its job fast"*.
6. "The computer was running", "the computer program was running", "the movie ran over two hours", "the play ran for six months", "the comic ran for 851 issues" all mean *"X did (fast) repetitive operations to do its job"*. "Run a bath" means *"use a (fast) flow to fill the bath with a homogeneous quantity (= water) it uses for its (repetitive) job"*.
7. "Run a store/hotel/restaurant/factory" means *"do a repetitive busy job to manage other repetitive busy jobs"*. "Run errands" means *"do a repetitive busy job"*.
8. "A press run of 10,000 copies" means *"a homogeneous quantity produced by repetitive work"*. Run-of-the-mill, "the average run of students" mean *"not different from a homogeneous quantity, produced repetitively"*.
9. "A run-down car", "a run-down hotel", "ran himself to death at work" all mean *"X became less functional from frequent repetitive influence (on its job)"*.
10. "I'm running out of fuel / supplies / ideas / patience" means *"there's not much left of the (homogeneous) quantity needed to do my (repetitive) job (which gives me a strict time limit before something bad happens)"*.
11. "Rioters run amok", "dogs love to run about and feel free", "chickens run loose" all mean *"X move hastily and freely without a job, without a care about any system/job"*. "He runs guns" means *"he operates with a homogeneous quantity outside of the system (= law)"*.
12. "My stocking is running", "colors run", "my nose is running", "tears are running", "a running sore", "speculation runs rife" all mean *"X disorganizes something by moving"* or *"X's movement/spread is a sign of disorganization"*.
13. "He runs with some decent-size locals", "he runs with a wild crowd" mean *"he does a job (or an anti-job, i.e. something fun or antisocial) with them"*. While "he refuses to run with the pack and buy every new gadget released" means *"he doesn't act (hastily) like the society"*.
14. "The road runs close to the shore", "the boundary line runs east", "my tastes in books run toward science fiction", "musical talent runs in the family", "boys and girls run up rapidly" all mean *"X extends in a direction, establishing something fundamental/fixed (= a road, a boundary, an interest, a talent, a level of maturity)"*.

15. "A run of good/bad luck", "a long run of cloudless days", "to have a good run" all mean *"X is a homogeneous quantity which happens, for a limited time, against the resistance of improbability"*. "Ran his fingers through his hair", "ran a red light", "run the blockade", "ran a sword through the body", "ran his head into a post", "ran into problems" all mean *"did a quick movement through resistance"* or *"was moving quickly/freely until meeting resistance"*.
16. Salmon run means *"fast movement against resistance to a birthplace (= something fundamental)"* or *"fast movement against resistance as a part of a big fundamental mission"*.

"Set" and "put" are synonyms... with sorta opposite meanings. So it's better to analyze them side by side.

"Set" (wiktionary) is a cluster of approximately the following abstractions:

- Gentle, precise, very deliberate, gradual action. Continuous influence.
- Stable, fixed, firm, special, irreversible action / place / position / state.
- Preparing, priming a thing for something.
- Changing into a final position / state.

"Put" (wiktionary) is a cluster of approximately the following abstractions:

- Unpleasant, harmful, forceful action (against resistance). Harsh, imprecise, abrupt, careless action. Action without thinking. Disposing, unloading, relieving action.
- Unstable, arbitrary, temporary place / position / state. (Dis)order. Uncertainty. Continuous effort, control.
- Something very external (foreign or antagonistic). High degree of separation.
- An action important in of itself, a deciding action, an action which directly leads to winning or losing something. An action solving a problem.

"Set" is calmer and more stable, "put" is more aggressive and unstable.

"set" and "put" examples

Approximate meanings of "set":

1. "Set a stone on the grave" means *"accurately place the stone on the special stable position"*. "Set a broken bone" means *"precisely place the bone into a stable position, making it ready to function again"*. "Clear sky set with stars" means *"stars on the sky are in very fixed positions"*.
2. "Set milk for cheese", "the cement sets rapidly", "this glue sets in five minutes" mean *"X is gradually, irreversibly transforms into Y, its final state"*. "The sun sets", "this century sets with little mirth" are about gradual finishing too. "The pudding sets heavily on my stomach" means *"the pudding continuously affects me, gradually finishing existing in my stomach"*.

3. "The bird was treated and then set free", "they accidentally set the house on fire", "the years have set their mark on him", "set the rules for the game", "set a record for the half mile", "war sets brother against brother" are all about semi-irreversible actions.
4. "Set a place for a guest", "set the stage", "set the sails", "set the alarm for 7:00", "set seedlings" (= transplant plants), "set a ladder against the wall", "set a wedding day" all mean *"place X into a very deliberate state/position to prepare for Y"*. "Set an example of generosity" means *"make a deliberate action to 'prepare'/prime someone for generosity"*.

Approximate meanings of "put":

1. "She put her books on the table" means *"she (without thinking) placed her books in an arbitrary, temporal position"*.
2. "The police put him in a cell" means *"the police did something unpleasant, harmful, forceful, harsh to him + disposed of him, controlled him + that was a decisive action + they were an antagonistic force"*. "Put the blame", "put in a box", "put to death", "put down", "put a bullet into", "put foot to ass", "put the fear of God into", "put someone in their place" are similarly violent. "Put your hands up" is a harsh request to make a deciding action.
3. "Put a damper on", "put a check on", "put a cork in it", "put a lid on", "put a special tax on luxuries" mean *"make a deciding action to control/end something with antagonistic force"*.
4. "Put on a happy face", "put on an act", "put lipstick on a pig" mean *"force an exterior completely separate from what's inside"*.
5. "I put £100 on the favorite horse", "what answer did you put for question 3?" are about uncertainty and deciding actions. "That seems like a bad idea put against the restrictions we're working under", "he is putting all his energy into this one task" are about uncertainty, resistance/effort and deciding actions. "Wouldn't put it past him" means an important unpleasant judgement.
6. "The doctor's put me on a strict diet" means *"the doctor (a foreign influence) made me stop careless eating, which is an action important in of itself and requires continuous effort"*. "Put your house in order" means *"make the (abrupt) effort to fix disorder"*.
7. "I put it to you, Sir, that you are a thief and a liar" means *"I give you my foreign opinion, uncertain what you'll do with this information, relieving the burden of those thoughts"*.
8. "Put the poem into English", "put my feelings into words", "lyrics put to music" mean *"solve the problem of translating into (or combining with) a foreign medium"*.

I claim that this is a plausible story and that polysemantic concepts are interpretable only to the extent to which a story like this is true. Because this is the only way to make polysemantic concepts local (§3.1).

4.3 Math

To the extent to which mathematical intuitions are connected to the rest of the human reasoning, they have to be grounded in (meta-level) detection / content / influence computations or in (meta-level) macro-computations.