Towards Symbolic and Language-Agnostic Large Language Models

WALID S. SABA

Institute for Experiential AI (EAI)
Northeastern University
w.saba@northeastern.edu

Abstract. Large language models (LLMs) have achieved a milestone that undeniably changed many held beliefs in artificial intelligence (AI). However, there remains many limitations of these LLMs when it comes to true language understanding, limitations that are a byproduct of the underlying architecture of deep neural networks. Moreover, and due to their subsymbolic nature, whatever knowledge these models acquire about how language works will always be buried in billions of microfeatures (weights), none of which is meaningful on its own, making such models hopelessly unexplainable. To address these limitations, we suggest combining the strength of symbolic representations with what we believe to be the key to the success of LLMs, namely a successful bottom-up reverse engineering of language at scale. As such we argue for a bottom-up reverse engineering of language in a symbolic setting. Hints on what this project amounts to have been suggested by several authors, and we discuss in some detail here how this project could be accomplished.

Keywords: Bottom-up reverse engineering of language, symbolic NLP.

1 Introduction

In general, scientific explanation proceeds in one of two directions: by following a top-down strategy or by following a bottom-up strategy (Salmon, 1989). For a top-down strategy to work, however, one must have access to a set of *general principles* to start with and this is certainly not the case when it comes to thought and how our minds externalize our thoughts in language. Nevertheless, decades of work in natural language processing (NLP) marched on inspired by generative linguistics, where an innate language faculty and a Universal Grammar were postulated (Chomsky, 1956), cognitive linguistics, where it was postulated that we metaphorically build our linguistic apparatus on top of a set of idealized cognitive models (ICMs) (Lakoff, 1987), or model-theoretic semantics (Montague, 1974), where it was postulated that natural languages, like formal languages, can be precisely specified using the tools of mathematical logic. However, in all cases there was very little in terms of established knowledge that these theories started from. In retrospect, then, and lacking any general principles one can speak of about our language (and *the language of thought*) it is no surprise that the bottom-up method succeeded where decades of top-down work in NLP failed to deliver. Moreover, and due to the intricate relationship between language and knowledge, this is perhaps the reason why much work in knowledge representation and ontology also failed (Sowa, 1995; Lenat, 1990), since most of this work amounted to pushing, in a top-down manner, various metaphysical theories of how the world is supposedly structured and represented in our minds, and again without any established general principles to start from.

On the other hand, a little more than a decade of work in bottom-up reverse engineering of language has produced very impressive results. With the release of GPT-4 it has become apparent that large language models (LLMs), that are essentially a massive experiment in a bottom-up reverse engineering of language, have crossed some threshold of scale at which point there was an obvious qualitative improvement in their capabilities[1]. It is our opinion that these capabilities mark a milestone, and not just a computational one, but a theoretical one, and we think it is one that linguists, psychologists, philosophers, and cognitive scientists must reflect on. In particular, we believe that a number of reservations expressed by luminaries in the philosophy of language and the philosophy of mind concerning the possibility of machine understanding are now questionable, if not outright irrelevant. For example, we believe the arguments of Hubertus Dreyfus (1972) who suggested that computers will never know what is *relevant* in a given situation, are not very convincing anymore since GPT-4 certainly replies with 'relevant' content in response to some prompt. Moreover, we believe the thought experiment devised by the philosopher John Searle (1980), one that questioned the possibility of machines exhibiting any semantics, to also be somewhat irrelevant now. While lots of ink has been spilled on what has become known by the Chinese Room Argument (CRA), current capabilities of LLMs clearly demonstrate not only a mastery of syntax but quite a bit of semantics too. Indeed, what the massive experiments that lead to LLMs have shown is that quite a bit of semantics, and even quite a bit of commonsense knowledge, both of which are clearly encoded in our everyday linguistic communication, can be uncovered in a bottom-up reverse engineering process[2]. But, in our opinion, this is where the good news ends for LLMs.

2 Limitations of LLMs

To begin with, and despite their relative success, we should remain cognizant of the fact that LLMs models are not (*really*) 'models of language' but are statistical models of the regularities found in linguistic communication. Models and theories should explain a phenomenon (e.g., $F = ma$) but LLMs are not *explainable* because explainability

[1] GPT stands for 'Generative Pre-trained Transformer', an architecture that OpenAI built on top of the transformer architecture introduced in (Vaswani, et. al., 2017).

[2] While this is not our immediate concern, but we believe this is what John Searle missed in his CRA thought experiment, namely that syntax and semantics are two sides of the same coin, and that mastering syntax implicitly means mastering quite a bit of the semantics that is embedded in the syntax, as has clearly been demonstrated by LLMs. It is for this reason that we can make syntactically valid expressions that are meaningless, but we cannot have a meaningful expression if it was not syntactically valid!

requires structured semantics and reversible compositionality that these models do not admit (Saba, 2023) (see also figure 1). In fact, and due to the subsymbolic nature of LLMs, whatever 'knowledge' these models acquire about language will always be buried in billions of microfeatures (weights), none of which is meaningful on its own. In addition to the lack of explainability, LLMs will always generate biased and toxic language since they are susceptible to the biases and toxicity in their training data (Bender et. al., 2021). Moreover, and due to their statistical nature, these systems will never be trusted to decide on the "truthfulness" of the content they generate (Borji, 2023)[3]. Note that none of these problematic issues are a function of scale but are paradigmatic issues that are a byproduct of the architecture of deep neural networks (DNNs).

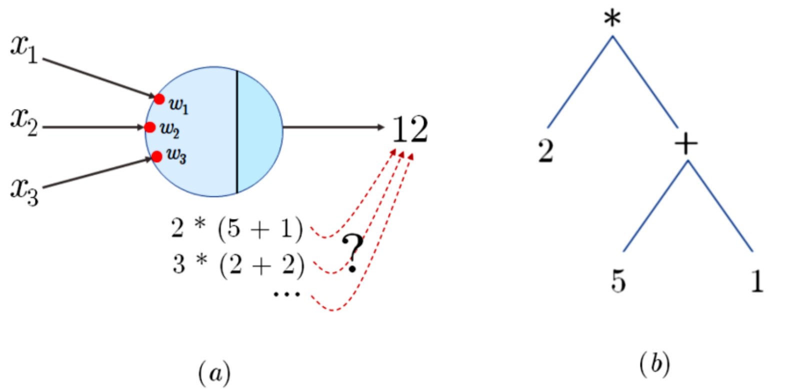

Figure 1. Compositional computation in subsymbolic systems (e.g., DNNs) is not invertible. The decomposition of 12 into its original components is undecidable (*a*), unlike symbolic systems (*b*), where there are structures that 'save' the semantic map of the computation.

There are other limitations that are also not a function of scale, but a byproduct of the underlying architecture. LLMs are based on the architecture of DNNs that do not admit any symbolic representations and are thus purely extensional models and would therefore fail to make the correct inferences in intensional contexts. Consider the example shown in figure 2(a) below. While 'Aristotle' and 'the tutor of Alexander the Great' have the same extension (they both refer to the same object), as objects of cognition they have different intensions (senses), thus their extensional equality should not license a replacement of one for the other as GPT-4 does, resulting in the absurd conclusion that '*perhaps the tutor of Alexander the Great was not the tutor of Alexander the Great*'. In figure 2(*b*) we have a similar situation were replacing Paris with '*the most populous city in France*' – while extensionally valid, also results in a

[3] Truth is not approximate, and not only when it comes to mathematical facts. Much like it is meaningless to speak of the probability of (6 = 2 * 4), it is also meaningless to assign any probability to the result of the database query '*Is John Smith the sales manager in our Chicago branch?*' – facts either are, or they are not.

non-truth, since Mary's stating her desire to visit Paris does not entail Mary's stating her desire to visit the most populous city in France. In addition to failing in intensional contexts LLMs cannot be relied upon in contexts where the resolution of scope ambiguities requires access to subtle commonsense knowledge. For example, in figure 2(c) GPT-4 does not correctly interpret "two museums" to mean "many museums" since, from the standpoint of commonsense, the correct reading should be "every tourist is taken to two museums by some student"[4]. Note that these tests are not exotic or farfetched and can always be reproduced as they are examples that are, in theory, beyond the capabilities of purely extensional LLMs (note that due to some added randomness, to simulate 'creativity', the same prompts may produce slightly different results).

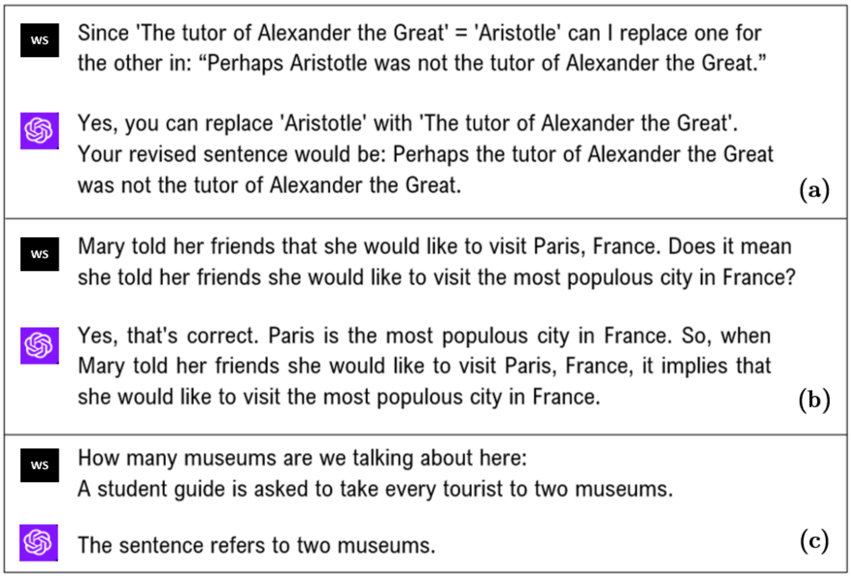

Figure 2. Examples showing how LLMs fail in modal and intensional contexts.

So where do stand now? From our discussion thus far it would seem that the glass is half full and half empty. On one hand, LLMs have clearly proven that one can get a handle on syntax and quite a bit of semantics in a bottom-up reverse engineering of language at scale; yet on the other hand what we have now are unexplainable models that do not shed any light on how language actually works and how we externalize our thoughts in language. Since we believe that the relative success of LLMs is not a

---

[4]A list of such examples involving intensional contexts as well as examples that involve commonsense reasoning (e.g. in the context of quantifier scope) can be found here `https://medium.com/ontologik/a-serious-chat-with-chatgpt-99e7de8d68c2`.

reflection on the symbolic vs. subsymbolic debate but is a reflection on (appropriately) adopting a bottom-up reverse engineering strategy, we think that combining the advantages of symbolic representations with a bottom-up reverse engineering strategy is a worthwhile effort. The idea that word meaning can be extracted from how words are actually used in language is not exclusive to linguistic work in the empirical tradition, but in fact it can be traced back to Frege, although there were more recent philosophical and even computational proposals on what this project amounts to. Below we will discuss these proposals in some detail.

3 Concerning "The Company a Word Keeps"

The genesis of modern LLMs is the *distributional semantics hypothesis* which states that the more semantically similar words are, the more they tend to occur in similar contexts – or, similarity in meaning is similarity in linguistic distribution (Harris, 1954). This is usually summarized by a saying that is attributed to the British linguist John R. Firth that "you shall know a word by the company it keeps". When processing a large corpus, this idea can be used by analyzing co-occurrences and contexts of use to approximate word meanings by word embeddings (vectors), that are essentially points in multidimensional space. Note, however, that this part of the story covers only what is called *lexical semantics*, which is the study concerned with word meanings. In particular, this part of the story does not address modeling syntactic rules nor compositional semantics, by which the meaning of larger linguistic units is obtained as some function of the meaning of the parts and how they appear together. Instead, the meaning of larger linguistic units in this tradition was usually obtained by some weighted vector addition operation, although there were many attempts to combine traditional compositional semantics with vector semantics in what has come to be known by compositional distributional semantics (CDS). See (Baroni, et. al., 2014) for an excellent review of this work.

While word embeddings can approximate lexical semantics (word meanings), it was not until the transformer model (Vaswani, et. al., 2017) that embeddings started the encoding of syntax. That is, what transformers and multiple attention heads did is create embeddings for 'valid' sequences and not just words. But how many of these sequences can one encode? Apparently, it has taken a massive network with over 500 billion encodings to master the syntax of language. In this regard, it is worth mentioning here an astute observation made by Stephen Wolfram (2023) regarding the size of these deep networks, namely that "the size of the network that seems to work well is so comparable to the size of the training data". In other words, it would seem that (roughly) an additional parameter (weight) was required for every additional token in the corpus. If this correlation is not accidental then it is another indication that such models cannot provide an explainable model/theory for how language works since it would mean that what these models are doing, in effect, is encoding (memorizing) all possible combinations of how words may appear in any sequence of words, which is hardly a theory of linguistic communication.

In summary, transformers with attention, along with massive scale, have allowed for a qualitative leap in the linguistic capabilities of LLMs. Still, at the root of this bottom-up reverse engineering of language is the concept of 'the company a word keeps' and the distributional semantics hypothesis that, unlike top-down approaches, "reverse engineers the process and induces semantic representations from contexts of use" (Boleda, 2020). But nothing precludes this ingenious idea from being carried out in a *symbolic* setting. In other words, the 'company a word keeps' can be measured in several ways, some of which, incidentally, have been discussed since Frege. We turn to this subject next.

4 Symbolic Reverse Engineering of Language

In discussing possible models (or theories) of the world that can be employed in computational linguistics Jerry Hobbs (1985) once suggested that there are two alternatives: on one extreme we could attempt building a "correct" theory that would entail a full description of the world, something that would involve physics and all the sciences; on the other hand, we could have a promiscuous model of the world that is isomorphic to the way we talk it about in natural language. Clearly, what Hobbs is suggesting here is a reverse engineering of language itself to discover how we actually use language to talk about the world we live in. In essence, this is not much different from Frege's Context Principal that suggests to "never ask for the meaning of words in isolation" (Dummett, 1981) but that a word gets its meanings from analyzing all the contexts in which the word can appear (Milne, 1986). Again, what this suggests is that the meaning of words is embedded (to use a modern terminology) in all the ways we use these words in how we talk about the world. While Hobbs' and Frege's observations might be a bit vague, the proposal put forth by Fred Sommers (1963) was very specific. Again, Sommers suggests that "to know the meaning of a word is to know how to formulate some sentences containing the word" and this would lead, like in Frege's case to the conclusion that a complete knowledge of some word w would be all the ways w is used and in every possible sentence. For Sommers, the process of understanding the meaning of some word w starts by analyzing all the properties P that can sensibly be said of w. Thus, for example, [*delicious Thursday*] is not sensible while [*delicious apple*] is. Moreover, and since [*delicious cake*] is also sensible, there must be a common type (perhaps **food**?) that subsumes both *apple* and *cake.* This idea seems similar to the idea of type checking in programming languages. For example, the types in an expression such as '$x + 3$' will only unify (or the expression will only 'make sense') if/when x is an object of type number (as opposed to a tuple, for example). As it was suggested in Saba (2007), this type of analysis can be used not only to discover the dimensions of word meanings, but to 'discover' the ontology that seems to be implicit in all natural languages.

Let us now consider the following naïve procedure for some initial reverse engineering of language, a procedure that was initially suggested in Saba (2007):

1. Consider concepts $C = \{c_1, ..., c_m\}$ and properties $P = \{p_1, ..., p_n\}$.

2. Assume the existence of a predicate, **app**(p, c) that holds true iff the property p applies to (makes sense of, or is sensible to say of) objects of type c, where $c \in C$ and $p \in P$.
3. A set $Cp = \{c \mid \mathbf{app}(p, c)\}$ is generated for all concepts $c \in C$ and all property $p \in P$ such that the property p is applicable (or can sensibly be applied to) c.
4. A concept hierarchy is then systematically discovered by analyzing the subset relationship between the various sets generated.

Applying the above procedure on a fragment of natural language and taking, initially, C to be a set of nouns and P a set of adjectives and relations that can sensibly be applied to (or can be said of) nouns in C, would result in something like the following:

R_1 : **app**(OLD, entity)
R_2 : **app**(HEAVY, physical)
R_3 : **app**(HUNGRY, living)
R_4 : **app**(ARTICULATE, human)
R_5 : **app**(MAKE(human, artifact))
R_6 : **app**(MANUFACTURE(human, instrument))
R_7 : **app**(RIDE(human, vehicle))
R_8 : **app**(DRIVE(human, car))

What the above say, respectively, is the following:

R_1 $\rightarrow$ in ordinary language we can say OLD of any entity
R_2 $\rightarrow$ we say HEAVY of objects that are of type physical
R_3 $\rightarrow$ HUNGRY is said of objects that are of type living
R_4 $\rightarrow$ ARTICULATE is said of objects that are of type human
R_5 $\rightarrow$ MAKE holds between a human and an artifact
R_6 $\rightarrow$ MANUFACTURE relates a human and an instrument
R_7 $\rightarrow$ RIDE holds between a human and a vehicle
R_8 $\rightarrow$ DRIVE holds between a human and a car

Note that the above 'findings' would eventually result in a well-defined hierarchy. For example, since a bottom-up reverse engineering of language will ultimately produce **app**(HEAVY, car) and **app**(OLD, car) – that is, since by analyzing our linguistic communication, we would also discover that it is sensible to say 'heavy car' and 'old car' it would seem that car must be a subtype of physical which in turn must be a subtype of entity. Similarly, since it makes sense to say MAKE of everything we can say we MANUFACTURE, an instrument must be a subtype of an artifact. The fragment of the hierarchy that is implicit in R_1 through R_8 is shown in figure 3 below.

Note, also, since **app**(ARTICULATE, human) says that 'articulate' is a property that can be said of objects of type human, we can rewrite this fact as **hasProp**(articulation, human), where ARTICULATE is reified (nominalized) as the trope articulation which is an abstract object of type property (see Moltmann, 2013 for more details on such abstract objects). Using the primitive and linguistically agnostic relation **hasProp** what we now have is a relation between two entities, the property of articulation and a human, which effectively states that articulation is a property that is usually attributed to (or said of) objects that are of type human. The same can be done with R_3, **app**(HUNGRY, living), resulting in **inState**(hunger, living) to say that any living entity can be in a state of hunger. The result of this discovery process (that produces linguistic knowledge such as R_1 through R_8) coupled with the nominalization process and using only primitive relations between entities will be no less than discovering (as opposed to inventing) the ontology that seems to be implicit in ordinary language.

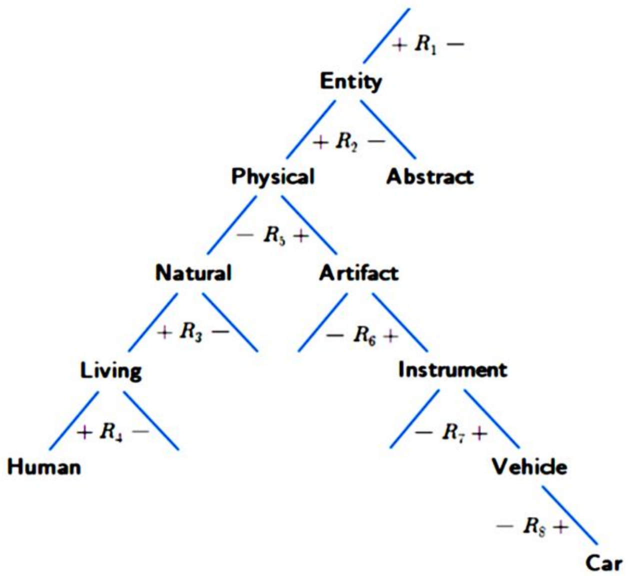

Figure 3. The hierarchy that is implicit in the 'discoveries' R_1 through R_8 above.

Before we discuss the nature of the ontology that seems to be implicit in language use, we need to answer the question of where do these primitive relations come from? That is, how do we discover all these primitive and linguistically agnostic relations, such as **hasProp** and **inState**? The answer to this question lies in the copular ('is' or verb to be). In general, when describing an object or an entity *x* by some property *P* we are, indirectly, making a statement such as '*x* is *P*'. If we analyze the various ways these descriptions can be made, it will lead us to different types of primitive relations, as shown in table 1 below. For example, in saying *Mary is wise*, we are essentially saying that *Mary has the property of wisdom*. Similarly, in saying *Carlos is ill*, we are essentially saying that *Carlos is in the* (physiological) *state of illness*. Analyzing different ways of describing different 'types' of entities would lead us to discover all the language agnostic primitive relations that are summarized in table 2 below.

Here's a summary of the overall process: (*i*) by analyzing a large corpus we can discover all pairs of *c* and *p* for which **app**(*p*, *c*) holds (e.g., **app**(ARTICULATE, human), **app**(HUNGRY, living)); (*ii*) via a nominalization process convert **app**(*p*, *c*) to two entities related by some primitive relation (e.g., **hasProp**(human, articulation), **inState** (human, illness)); (*iii*) construct the ontology implicit in all the discovered relations.

Table 1. Discovering the language agnostic primitive relations.

LINGUISTIC CONTEXT	IMPLICIT PRIMITIVE RELATION
Frido ***is*** a dog	Frido ***instanceOf*** dog
Billy the Kid ***is*** William H. Boney JFK ***is*** John Fitzgerald Kennedy	Billy the Kid ***eq*** William H. Boney JFK ***eq*** John Fitzgerald Kennedy
Mary ***is*** wise Julie ***is*** articulate	Mary ***hasProp*** wisdom Julie ***hasProp*** articulation
Jim ***is*** sad Carlos ***is*** ill	Jim ***inState*** sadness Carlos ***inState*** illness
Sara ***is*** running Olga ***is*** dancing	Sara ***agentOf*** running John ***agentOf*** dancing
Sara ***is*** greeted Sara ***is*** acknowledged	Sara ***objectOf*** greeting Sara ***objectOf*** acknowledgment
John ***is*** 5'10'' tall Dan ***is*** 69 years old	John's *height* ***hasValue*** 5'10'' Dan's *age* ***hasValue*** 69 yrs
Sheba ***is*** running	Sheba ***participantIn*** running (event)
Olga ***is*** dancing	Olga ***agentOf*** dancing (activity)

Table 2. A summary of the language-agnostic primitive relations.

PRIMITIVE RELATIONS	DESCRIPTION
Eq(x, y)	**individual** x is identical to individual y
Part(x, y)	individual x is part of individual y
Inst(x, y)	individual x instantiates **universal** y
Inhere(x, y)	individual x inheres in individual y
Exemp(x, y)	individual x exemplifies property y
Dep(x, y)	individual x depends for its existence on individual y
IsA(x, y)	universal x is a sub-kind of universal y
Precedes(x, y)	individual process x precedes individual process y
HasParticipant(x, y)	individual y participates in individual occurrent x
HasAgent(x, y)	individual y is agent of individual occurrent x
Realizes(x, y)	individual process x realizes individual function y
TypeOf(x, $\mathbf{t}$)= ($x :: \mathbf{t}$)	individual x is an object of type $\mathbf{t}$

5 Dimensions of Word Meanings

What we have suggested thus far is a bottom-up reverse engineering of language using the predicate **app**(p, c) that effectively generates sets for all nouns c that the property p is applicable of. This in turn can be converted into a triple **R**(entity, entity) corresponding to([entity] → (**R**) → [entity]) after all the concepts have been reified, where **R** is a primitive and language agnostic relation. Since every entity can now be defined by primitive relations, these primitive relations would now represent the dimensions of word meanings. In figure 3 we show these dimensions for (one of) the meanings of the word book.

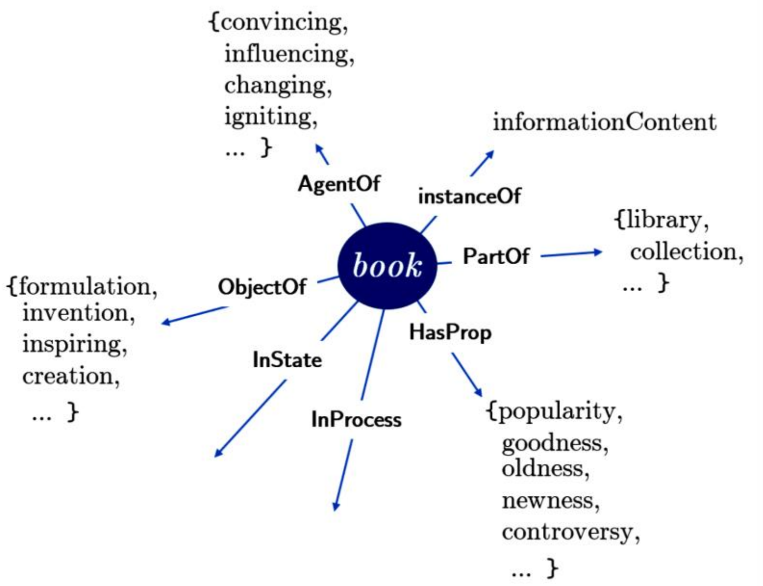

Figure 3. The primitive and linguistically agnostic relations as the dimensions of word meaning (in this case of one meaning of the word book).

As shown in figure 3, one meaning of the word *book* (which in WORDNET is "a written work or composition that has been published") is an entity (*i*) that can be the agent of a changing event (as in '*Das Kapital changed many opinions on capitalism*'); (*ii*) that can have the popularity property (as in '*The Prince is a popular book*'); (*iii*) that can be the object of an inspiring event (as in 'Hamlet inspired many movies'), etc. Note that in our reverse engineering process we have discovered that popularity is a property that books can have, that is,

popularity ∈ book . **hasProp**

We could've instead associated a weight with these entries, for example (w_1, popularity) ∈ book . **hasProp** and (w_2, fame) ∈ book . **hasProp**, and where $w_1 > w_2$ would indicate that popularity is used when describing books more than fame – or that we say '*popular book*' much more than we say '*famous book*'. However, such weights would introduce bias as they represent accidental and temporal measures of the specific text processed at a specific point in time. For the moment let us ignore the weights and let us use GPT-4 to generate some of these vectors along the various dimensions. The data in figure 4 is obtained by asking GPT-4 to provide 25 "plausible" (or "sensible") replacements for the [**MASK**] in the given sentences.

The book has [**MASK**] *millions of people*	*Jon has* [**MASK**] *the book*	*Das Kapital was a very* [**MASK**] *book*
1. influenced	1. wrote	1. influential
2. inspired	2. criticized	2. impactful
3. educated	3. endorsed	3. controversial
4. affected	4. debated	4. challenging
5. engaged	5. read	5. analytical
6. perplexed	6. quoted	6. detailed
7. challenged	7. discussed	7. scholarly
8. reached	8. interpreted	8. complex
9. enlightened	9. appreciated	9. profound
10. motivated	10. translated	10. radical
11. stirred	11. reviewed	11. enlightening
12. provoked	12. studied	12. dense
13. intrigued	13. analyzed	13. thought-provoking
14. alarmed	14. examined	14. significant
15. shaped	15. dismissed	15. rigorous
16. guided	16. understood	16. comprehensive
17. fascinated	17. refuted	17. polemical
18. informed	18. praised	18. controversial
19. captivated	19. digested	19. transformative
20. provoked	20. researched	20. critical
21. challenged	21. referenced	21. pivotal
22. transformed	22. challenged	22. historical
23. touched	23. summarized	23. theoretical
24. awakened	24. defended	24. intricate
25. stimulated	25. bought	25. philosophical

Figure 4. Querying GPT-4 to complete contexts with plausible actions/relations and properties that can plausibly (sensibly) be said of (or apply to) a book. Starting with the first column we can see that as an agent of some action or activity, a book can influence, inspire, motivate, educate, etc. people; as the object of some activity, a book can be translated, interpreted, examined, refuted, etc. and finally, a book can have the property (or can be described as being) significant, critical, historical, influential, controversial, etc.

Using this strategy we can also 'discover' the underlying ontology that seems to be implicit underneath our ordinary language. In figure 5 below we apply masking to generate the most plausible actions that a computer, a car, and a couch can be the object of. Note that while the three types of objects can be the objects of ASSEMBLE (we 'assemble' computers, cars, and couches), we can sensibly say a computer or a car is RUNNING (or that a computer or a car is OFF) but the same is not true of a couch. This tells that while a computer and a car must have some common supertype, these two types seem to eventually belong to a different branch from couch although the three objects must have a common supertype at some level of abstraction since they can all be 'assembled' (see figure 6). Analyzing the sets generated from this bottom-up reverse engineering would easily yield such an ontological structure.

R(x, car)		R(x, computer)		R(x, couch)	
1.	INNOVATE	1.	USE	1.	PURCHASE
2.	CONCEPTUALIZE	2.	OPERATE	2.	ASSEMBLE
3.	OPERATE	3.	PROGRAM	3.	MOVE
4.	UTILIZE	4.	BUILD	4.	UPHOLSTER
5.	SKETCH	5.	DEVELOP	5.	CLEAN
6.	DESIGN	6.	IMPROVE	6.	ADVERTISE
7.	BUILD	7.	ENGINEER	7.	SELL
8.	CRAFT	8.	CREATE	8.	POSITION
9.	CREATE	9.	INVENT	9.	ADMIRE
10.	MODIFY	10.	DESIGN	10.	INSTALL
11.	VISUALIZE	11.	MANUFACTURE	11.	INHERIT
12.	IMPROVE	12.	PRODUCE	12.	REPLACE
13.	DEVELOP	13.	INNOVATE	13.	COVER
14.	ENGINEER	14.	REFINE	14.	PHOTOGRAPH
15.	DRIVE	15.	EVOLVE	15.	REARRANGE
16.	RIDE	16.	ADVANCE	16.	INSPECT
17.	CONSTRUCT	17.	UTILIZE	17.	SIT ON
18.	ASSEMBLE	18.	MAINTAIN	18.	DISLIKE
19.	FABRICATE	19.	UPGRADE	19.	DONATE
20.	EVOLVE	20.	ENHANCE	20.	CHOOSE
21.	REFINE	21.	EXPAND	21.	FIND
22.	PIONEER	22.	OPTIMIZE	22.	DELIVER
23.	NAVIGATE	23.	ASSEMBLE	23.	STAIN
24.	PROTOTYPE	24.	STUDY	24.	VACUUM
25.	PLAN	25.	COMMERCIALIZE	25.	RESTORE

Figure 5. Actions (verbs) that are sensible to say of a car, a computer and a couch.

To appreciate how the ontological structure implicit in ordinary language can be used in handling very complex phenomena in natural language we give one example here that involves what is called metonymy. Consider the following sentence (said by a waiter in some diner, and its commonsense interpretation:

The loud omelet wants a beer.
→ *The loud* [**person eating the**] *omelet wants a beer.*

How is it that speakers of ordinary language manage to uncover the missing and implicitly assumed (bold) text? The answer is as follows:

1. the 'want' relation has the type constraint WANT(human, entity)
2. the text speaks of an omelet wanting a beer, i.e. WANT(omelet, beer)
3. WANT(omelet, beer) can be generalized to WANT(food, entity) since omelet 'is a' food and beer is (ultimately) an entity.
4. what is said, WANT(food, entity), and what is expected, WANT(human, entity), while they unify on the type of the object, the types of the agent (human and food) do not unify, since neither is a subtype of the other.
5. the most salient relationship between human and food must now be retrieved, and that relationship is EAT(human, food)
6. the phrase 'the loud omelet' is now understood to mean 'the loud person eating the omelet'.

For a more detailed discussion on how the ontological structure implicit in ordinary language can be used in handling very complex phenomena in natural language the reader can consult (Saba, 2020).

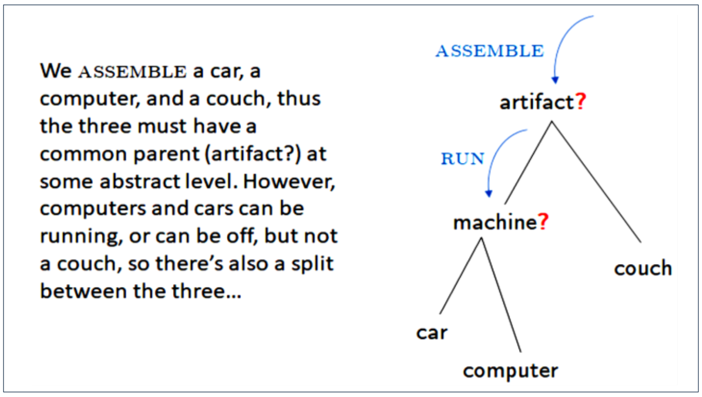

Figure 6. A computer, a car, and a couch can be assembled, so at some level of abstraction the three types must have a common parent (artifact?). However, cars and computers, although not couches, RUN and can be described by being OFF and so they eventually must be in different branches.

6 Concluding Remarks

Large language models (LLMs) have proven that a bottom-up reverse engineering of language at scale is a viable approach. However, due to their subsymbolic nature, LLMs do not provide us with an explainable model of how language works nor how we externalize the thoughts we contemplate in language. The idea of a bottom-up reverse engineering of language, which LLMs proved to be viable approach could however be done in a symbolic setting, as has been suggested previously going back to Frege. The obvious and ideal solution, therefore, would be to combine the advantages of a bottom-up reverse engineering approach with an explainable symbolic representation, as we have done in this paper. How the symbolic dimensions of word meanings we discussed in this paper are used in language understanding would be the subject of future work.

References

1. Baroni, M. Bernardi, R. and Zamparelli, R. (2014), Frege in Space: A Program for Compositional Distributional Semantics, *Linguistic Issues in Language Technology* (LiLT), Volume **9**, pp. 214-346.
2. Bender, E. M., Gebru, T. et. al. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623.
3. Boleda, G. (2020). Distributional Semantics and Linguistic Theory, *Annual Review of Linguistics*, 6:213–34.
4. Borji, A. (2023). A Categorical Archive of ChatGPT Failures, Available online at https://arxiv.org/abs/2302.03494
5. Chomsky, N. (1957). *Syntactic Structures*, Mouton de Gruyter, Berlin, New York.
6. Dreyfus, H. (1972), *What Computers Still Can't Do: A Critique of Artificial Reason, Cambridge*, MA, MIT Press.
7. Dummett, M. (1981). *Frege: Philosophy of Language.* Harvard University Press.
8. Harris, Z. S., (1954). Distributional Structure. *Word* 10:146–62
9. Hobbs, J., (1985). Ontological promiscuity. In *Proc. of the 23rd Annual Meeting of the Assoc. for Computational Linguistics*, Chicago, Illinois, 1985, pp. 61–69
10. Lakoff, G. (1987). *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*, University of Chicago Press.
11. Lenat, D. B., Guha, R.V. (1990). *Building Large Knowledge-Based Systems: Representation & Inference in the CYC Project.* Addison-Wesley.
12. Milne, P. (1986). Frege's Context Principle, *Mind*, Vol. **95**, No. 380, pp. 491-495
13. Moltmann, F. (2013). *Abstract Objects and the Semantics of Natural Language*, Oxford University Press.
14. Montague, R. (1973). The Proper Treatment of Quantification in Ordinary English. In: Kulas, J., Fetzer, J.H., Rankin, T.L. (eds) *Philosophy, Language, and Artificial Intelligence. Studies in Cognitive Systems*, vol **2**.
15. Saba, W. (2022). New Research Vindicates Fodor and Pylyshyn: No Explainable AI Without 'Structured Semantics', *Blog Comm. of the ACM*, September 14, 2022.
16. Saba, W. (2020), Language, Knowledge and Ontology: Where Formal Semantics Went Wrong, and How to Go Forward, Again, *Journal of Knowledge Structures and Systems* (JKSS), **1** (1): 40-62.
17. Saba, W. (2007). Language, logic and ontology: Uncovering the structure of commonsense knowledge, *Int. J. of Human Computer Studies*, **7** (65), pp. 610-623

18. Salmon, W. (1989). Four decades of scientific explanation, in: P. KITCHER & W. SALMON (Eds) *Minnesota Studies in the Philosophy of Science, Vol. XIII* (Minnesota, University of Minnesota Press), pp. 3-21
19. Searle, J. (1980). Minds, Brains, and Programs, *Behavioral and Brain Sciences*, Volume **3**, pp. 417–457.
20. Smith, B. (2005). Against Fantology, In J. Marek and E. M. Reicher (eds.), *Experience and Analysis*, pp. 153–170
21. Sommers, F. (1963). Types and ontology. *Philosophical Review,* **72** (3), 327-363.
22. Sowa, J. F. (1995). *Knowledge Representation: Logical, Philosophical and Computational Foundations*, PWS Publishing Company, Boston.
23. Vaswani, A., Shazeer, N., et. al., (2017), Attention is All You Need, Available online at https://arxiv.org/abs/1706.03762.
24. Wolfram, S. (2023). What is ChatGPT Doing, and Why Does it Work? Available online at https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/