AN INTUITIVE LOGIC FOR UNDERSTANDING AUTOREGRESSIVE LANGUAGE MODELS

Gaia Carenini, Alexandre Duplessis
Computer Science Department
ENS - PSL Research University
Paris, France
name.surname@ens.psl.eu

Adnan Ben Mansour
Independent Researcher
Paris, France
adnan.ben.mansour.cs@gmail.com

ABSTRACT

Transformer-based language models have shown a stunning collection of capabilities but largely remain black boxes. Understanding these models is hard because they employ complex non-linear interactions in densely-connected layers and operate in high-dimensional spaces. In this article, we address the problem of interpretability of large regressive language models with a principled approach inspired by basic logic. First, we show how classical mathematical logic does not grasp the reasoning system of these models and we propose the *intuitive logic*, which is notoriously asymmetric and redefines the classic logical operators. We then proceed with the localization of the activated areas associated with the conjunction, disjunction, negation, adversive conjunctions and conditional constructions. From the localization results, we obtain topological important information about the network that induces the formulation of a conjecture about the mechanisms underlying the intuitive logic introduced in GPT 2-XL. We test the conjecture through model editing and conclude by laying the foundations for a connectomics for GPT. The code is available at: https://github.com/Adnan-Ben-Mansour/hackathon2022.

***K*eywords** Autoregressive Language Model · GPT 2-XL · Logic & Proofs · Interpretability

1 Introduction

Transformer-based language models have shown a stunning collection of capabilities but largely remain black boxes. Despite this obscurity, language models are increasingly employed in a wide range of applications, spanning from the realization of chat-bots [1] to the development of medical models [2]. This class of applications requires an assessment of possible undesirable behaviors and strong guarantees regarding the predictability of the model, justifying for instance the emergent behaviors (e.g. [3]). In the absence of the latter, security threats might arise (e.g. [4], [5] and [6]).

Mechanistic interpretability attacks these questions providing tools to better analyse this colossal architectures by reverse engineering model computation into human-understandable components (e.g. [7]). In particular, recent breakthroughs have shed light on some basic aspects of the architecture of GPT, such as the storing of factual associations [8] and the circuits performing indirect object identification [9]. However, to the best of our knowledge, no result has yet captured the circuits underlying the logical behavior of large language models. The existence of such circuits is supported by experimental evidence: transformer-based language models can indeed perform numerous tasks involving logic skills, such as selection-inference [10] or automatic theorem proving [11].

Contribution In this paper, we propose a systematic analysis of latent logic at GPT 2-XL. First, we exclude the possibility of using standard logic. After, starting from the basic building blocks consisting of a redefinition of logical connectors ($\wedge, \vee, \neg$), we assess the ability of this infrastructure to intuitively grasp the meaning of formal logic (intuitive logic). Then, based on the method proposed in [8], we proceed with the location of the activation areas corresponding to the above-mentioned operators, this step shows unexpected invariants and properties of the operators that we discuss in details. From there, we conceive a conjecture about the mechanisms underlying the intuitive logic intrinsic in GPT 2-XL and test the conjecture through model editing. Eventually, we conclude by describing future research directions.

Among the remarkable results, we underline the following two the one for which each class of synonym is related to a unique "or" structure and the discovery that the "or" does not possess a centralized structure.

2 An Intuitive Logic for GPT

This first section attempts to analyse the logic engine of GPT 2-XL [12]. Starting from mathematical logic and observing violations of it at the linguistic level, we first attempt to outline the logical traits learnt from language that can motivate GPT's reasoning capabilities.

Remark Full experimental data and results are available in the appendix.

2.1 Standard logic doesn't work

Premise Within classical logic, there are numerous invariants and symmetries (e.g. [13]) that constitute a characterising element and are widely used in combinatorial optimisation and satisfiability problems (e.g. [14]). Proving that such symmetries are not respected by a language model is sufficient to move away from the classical logical approach. Human language, on which GPT is trained, is characterised by being asymmetry with respect to the $\wedge$ operator [15]. This suggests the need to test whether the model itself has captured this structure from the frequency gap in the transpositions.

Experiment The experiment performed focuses on the operator $\wedge$, but gives similar results for $\vee$. We first construct natural language (standard English) prompts of the form s *is* a *and*, where s is a subject of the set $\mathcal{S}$ and a is an adjective relative to the subject of the set $\mathcal{V}$. We therefore ask GPT 2-XL for completion in the form of a probability distribution p defined over a set of adjectives $\mathcal{V}$. This data is then organised in a matrix form M, starting from which, we calculate the difference between M and its transposition. This value well measure the asymmetry of the matrix.

Results The result can be visualized in Figure 1, where we can immediately see that there is no naive symmetry of the $\wedge$ operator.

2.2 Logical content identified

As noted in the previous paragraph, it is necessary to redefine the basic logical operators so that they can be descriptive. We do that below, by introducing an *intuitive logic* for GPT 2-XL. This construction is developed with a view to being able to be extended to other language models and to be consistent with the imperfect logics underlying standard linguistics. The basic elements we are analysing in this context are: weak equality, conjunction, disjunction, negation, the adversative clause and if-then constructs. Moreover, we make a parenthesis to observe an abstraction involving adjectives.

2.2.1 Weak equality

Premise In classical logic, equality commonly denotes a binary relationship of equivalence between two entities, called members of the equality. However, natural language has a multi-layered structure, in particular we can distinguish between a semantic level and an ensemble level. Consider a set of distinct words $\mathcal{W}$ and a pair of distinct words $w, w' \in \mathcal{W}$. From an ensemble point of view, it is clearly evident how w' *is not* w, i.e. belongs to the set $\mathcal{W} - w$. From a semantic point of view, on the other hand, the relation w' *is not* w can be arbitrarily false when w and w' are synonymous [1]. We therefore introduce a notion of weak equality where two elements are weakly equal, with respect to naive logic, if they are synonymous. We test the presence of weak equality in GPT 2-XL with the following experience.

Experiment We first build natural language prompts (more precisely in standard English) of the form s *is* a *and*, where s is a subject of the set $\mathcal{S}$ and a is an adjective relative to the subject of the set $\mathcal{V}$. We therefore ask GPT 2-XL for completion in the form of a probability distribution p defined over a set of adjectives $\mathcal{V}$. Then, we normalize over the restricted vocabulary $\mathcal{V}$ through a *soft maximization* and rewrite the result of the normalization in matrix form $P = (p(a, b))$ where $a, b \in \mathcal{V}$. Eventually, we conclude with a second normalization of the matrix, followed by an averaging with its transposition, i.e. we perform the following operation:

$$\frac{\frac{p(a,b)}{\sum_{a' \in \mathcal{V}} p(a',b)} + \frac{p(b,a)}{\sum_{b' \in \mathcal{V}} p(b',a)}}{2} \tag{1}$$

for every a and b in $\mathcal{V}$. This operation enforces symmetry.

[1] Synonyms are understood in the classical linguistic sense and have been checked through the online generator *Thesaurus*.

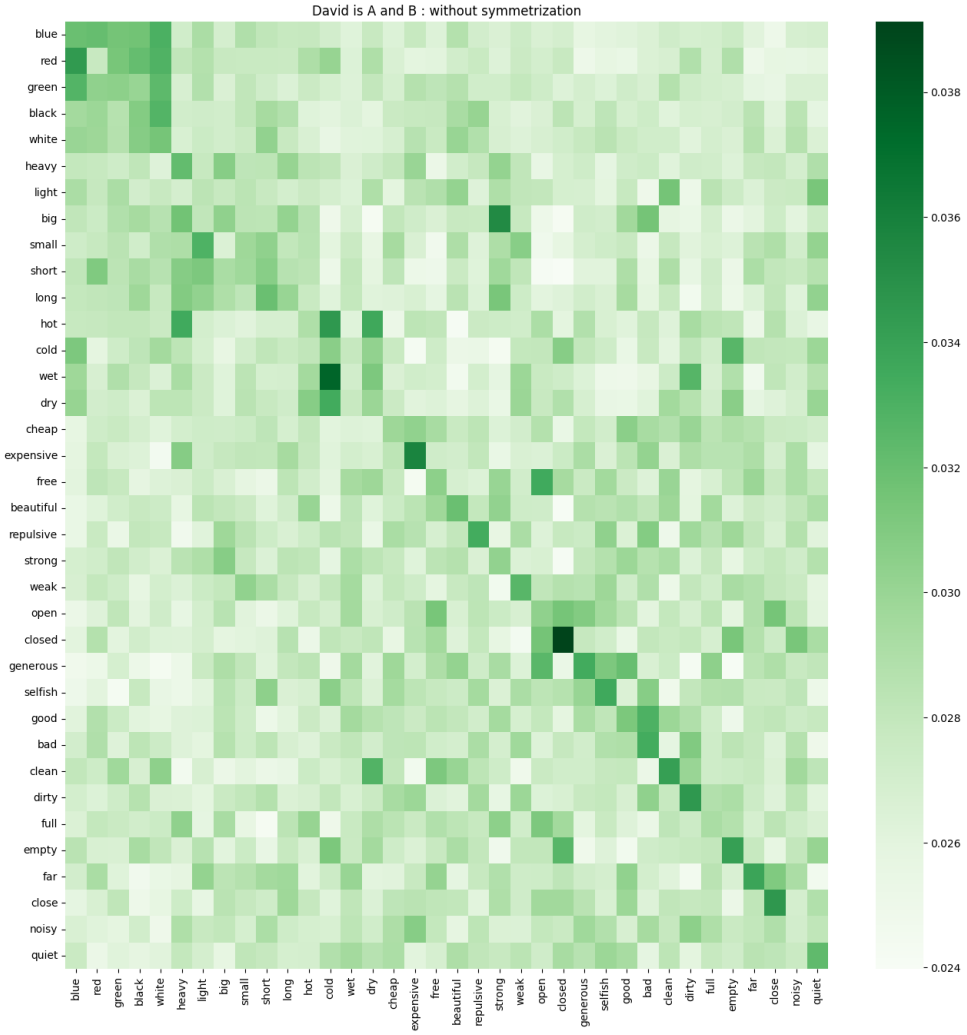

Figure 1: *Analysis of the Symmetry* On the two axes, there are the elements of $\mathcal{V}$. The color of the entries in the matrix is roughly speaking a measure of the compatibility of the adjectives labelling its coordinates.

Results The results can be visualised in Figure 2 where it is clear how adjectives with related meanings, e.g. *kind, loving, friendly*, form clusters that give the matrix the form of a block diagonal matrix. This particular shape of the matrix is obtained thanks to the choice of an appropriate ordering on the axes.

2.2.2 Conjunction

Premise Within natural language, the conjunction "and" generates ambiguity from a logical point of view [16]. For example, apparently contradictory adjectives, such as "black" and "white", can be in the same sentence: *A zebra is black and white*. Therefore, it would seem natural to assume that this conjunction is not a basic construct of the logic of GPT 2-XL.

Experiment The experiment is the same as the one used for weak equality.

Results The results can be visualised in Figure 2 where we observe how the "and" operator of GPT actually takes on the role of a synonym detector. Indeed, it appears to measure logical rather than linguistic compatibility. The maximum for this compatibility is held by the adjective itself as if the model acquires the knowledge s *is* a and completes the sentence after the "and" by answering the question: "What is s?"

2.2.3 Disjunction

Premise Contrary to $\wedge$, the operator $\vee$ is unambiguous from a linguistic and logical point of view. It is however important to notice how in natural language (Standard English) disjunction tends to be exclusive [17]. The associated intuitive "not" operator will therefore partially reflect the properties of the classic $\vee$ operator with a bias making it more exclusive in nature.

Experiment The experiment is the same as the one used for weak equality, the only difference is given by the prompt which in this case is s *is* a *or*, where is a subject of the set $\mathcal{S}$ and a is an adjective relative to the subject of the set $\mathcal{V}$.

Results The results can be visualised in Figure 2 where we note how well the or operator has mastered the disjunction of the form a *or not* a. In particular, this is more evident along the diagonal of blocks. Furthermore, it is possible to observe how colours seem to be recognised as a real class, in particular by donating a prompt of the form s *is* a where s is a subject in $\mathcal{S}$ and a is a colour in $\mathcal{V}$ there is a very high probability that the completion is itself a colour. This phenomena is analyzed more in detail in a dedicated paragraph below.

2.2.4 Negation

Premise In classical logic, negation is understood to be a unitary logical operation, which returns the inverse truth value of a proposition. Clearly from a linguistic point of view, this is no longer true for the observation made about the ensemble level and the semantic level. From the above experiences, one can see a semantic behaviour that allows one to see the intuitive negation operator as the corresponding standard logical operator that nevertheless acts on the vocabulary modulo the weak equality relation rather than the original one.

Experiment The experiment is the same as the one used for weak equality, the only difference is given by the prompt which in this case is s *is* a*, s is not*, where s is a subject of the set $\mathcal{S}$ and a is an adjective relative to the subject of the set $\mathcal{V}$.

Results The results can be visualised in Figure 3 where we can observe the level of masterization of the logical not modulo weak equality. Indeed, it is pretty clear how on the diagonal the antinomials are "paired".

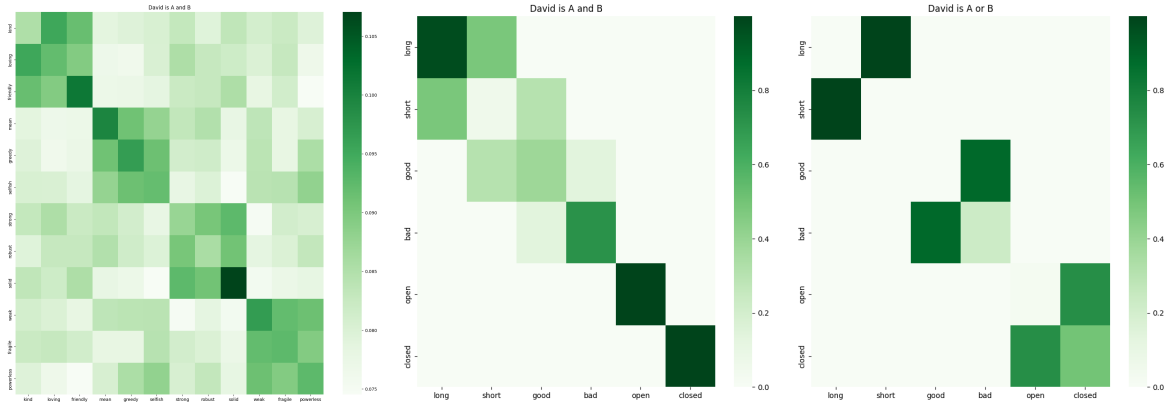

Figure 2: From the left to the right, we find the matrices associated to weak equality, conjunction and disjunction respectively. The color of the entries in the matrix is roughly speaking a measure of the compatibility of the adjectives labelling its coordinates.

2.2.5 Adversive Conjunctions

Premise Adversative conjunctions express opposition or contrast. This position is intuitively associated with negation and therefore seems to be of interest introducing them as well.

Experiment The experiment is the same as the one used for weak equality, the only difference is given by the prompt which in this case is s *is* a *but*, where s is a subject of the set $\mathcal{S}$ and a is an adjective relative to the subject of the set $\mathcal{V}$. In the plotted visualization we took the average over three synonyms for each adjective, this simple transformation allows us to get a sharper result.

Results The results can be visualised in Figure 3 where we can observe a behavior similar to the one observed with negation.

2.2.6 If-then Statements

Premise In natural language, the if-then construct expresses the relationship of consequentiality. The presence of a structure in GPT capable of processing such forms is therefore in order to investigate the inferential basis of such a language model.

Experiment The experiment is the same as the one used for weak equality, the only difference is given by the prompt which in this case is *If s is a, then s is*, where s is a subject of the set $\mathcal{S}$ and a is an adjective relative to the subject of the set $\mathcal{V}$.

Results The results can be visualised in Figure 3 where we can observe the naive principle for which *If s is a, then s is a*.

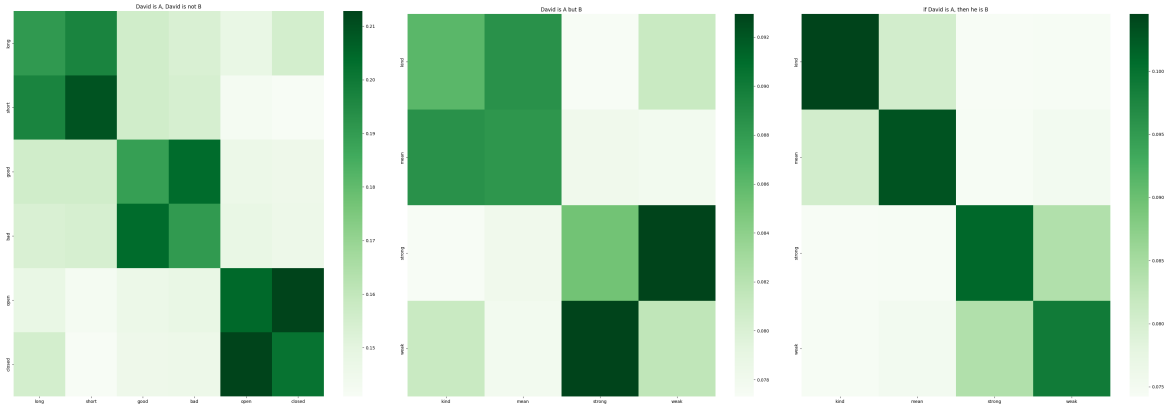

Figure 3: From the left to the right, we find the matrices associated to weak negation, adversative conjunction and if-then construct respectively. The color of the entries in the matrix is roughly speaking a measure of the compatibility of the adjectives labelling its coordinates.

2.2.7 Adjectives Abstraction

The above experiments drew our attention about a possible abstraction effect involving adjectives. With the experiment described below, we investigate this phenomenon further.

Experiment The experiment is the same as the one used for weak equality.

Results The results of that experience can be visualized in Figure 4, where indeed we note that adjectives tend to take on a categorization well compatible with that developed in English grammar.

3 Locating Neurons Involved in Logical Tasks

The results of the previous section show that there is indeed an intuitive logic justifying the behaviour of GPT 2-XL. In this section, we present the localisation of the aforementioned logical functions by taking up ROME's intuition [8].

3.1 Experiment

Our localisation method is almost fully derived from the causal tracing method of [8]. More precisely, we compute the average indirect effect (AIE) over different positions in the sentence and different model components including individual states, MLP layers, and attention layers (see [8] for more details).The main difference is that we do not study triples of the form *(subject, relation, object)* but rather quadruples of the form *(subject, adjective, relation, adjective)*.

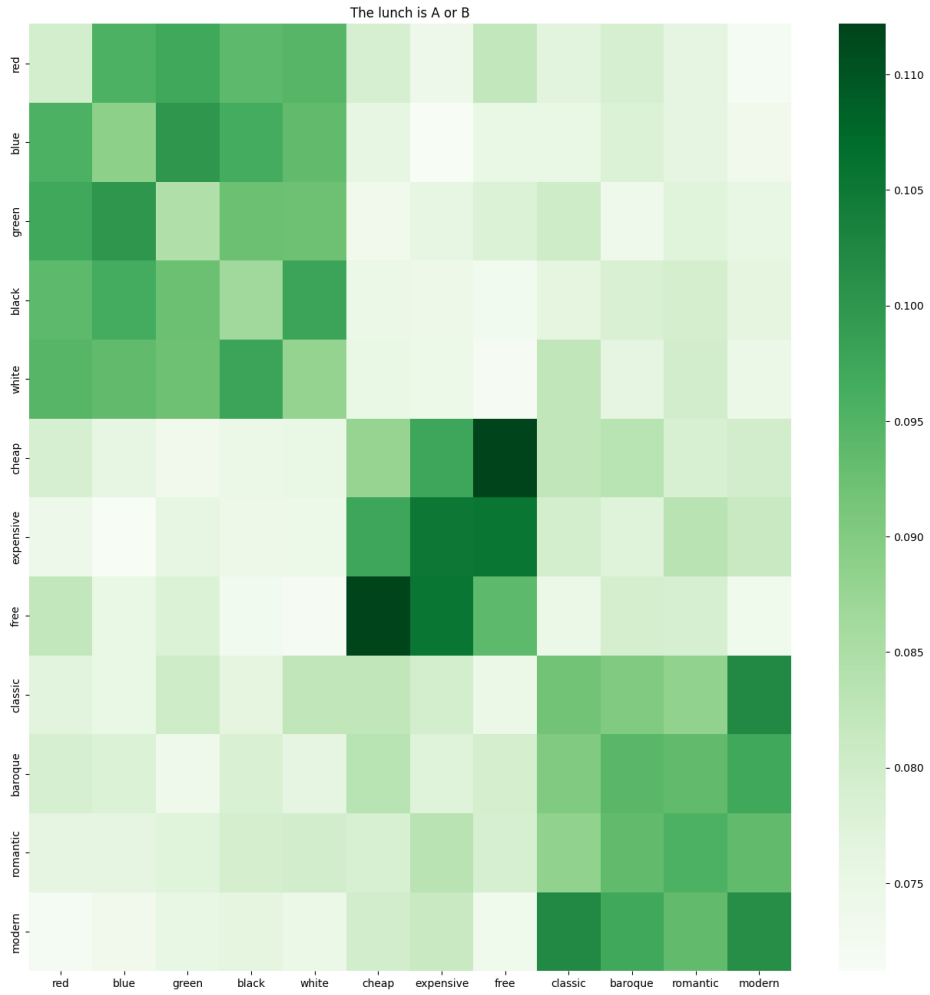

Figure 4: *Analysis of the Adjective Categorization* On the two axes, there are the elements of $\mathcal{V}$. The color of the entries in the matrix is roughly speaking a measure of the compatibility of the adjectives labelling its coordinates.

3.2 Results

In the following, we call MLP / Attention sensitive sites the parts of the network that have an influence on the output recovering (see diagrams). In general, as shown in [8], we get a late site immediately before the prediction which is not surprising, but also an early site at the last token of the corrupted part. The MLP is classically seen as a key-value mapping recalling facts.

3.2.1 Invariants

Our first localization experiments evidence the existence of two important invariants.

Invariance of the localization of the MLP and Attention sensitive sites under subject change To prove this property we compared the locations of each of the two sites for prompts of the form s *is a and* where $s \in \mathcal{S}$. This result has remarkable consequences, in fact, it means that whatever structure encodes "and" (or "or"...), its architecture and location do not depend on the subject (not even on the type of subject).

Invariance of the localization of MLP and Attention sensitive sites under change of adjective More precisely, when using prompts s *is a and* where $s \in \mathcal{S}$, we maintain the same sites for logical operator "or", "but", or "not". This is an even more crucial result since it tends to show that each logical operator has a specific Attention site (indirectly pointing to a storage MLP).

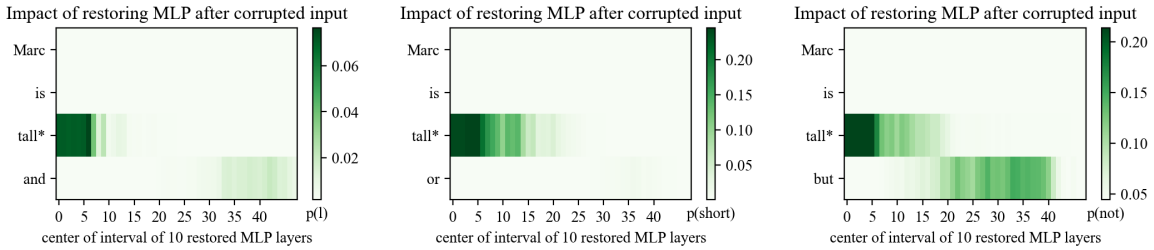

Figure 5: From the left to the right, we find the sensitive sites on MLPs layers for the "and", "or" and "but" logical operators, the query is "Marc is tall [operator]".

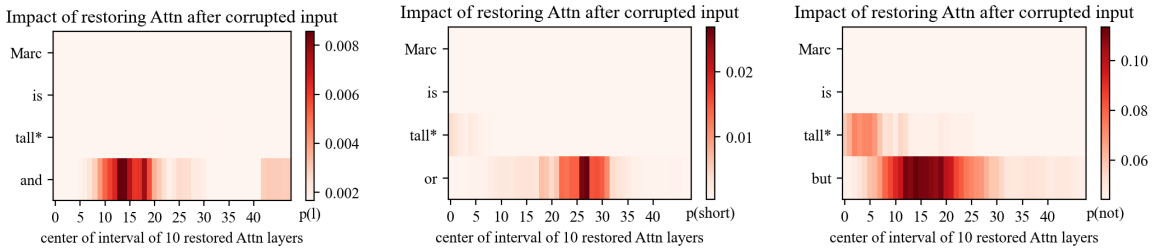

Figure 6: From the left to the right, we find the sensitive sites on attention heads for the "and", "or" and "but" logical operators, with the same query "Marc is all [operator]".

3.2.2 Clustering of and in the different Attention

After comparing the location of the Attention sensitive sites for different adjectives, we deduce that each synonymy class is related to a unique "or" structure. Moreover, we observe that each synonymy class is "located" at the same place as it is its opposite class (i.e. the class of the antonyms). At the same time, we remark that several disjoint classes can have the same Attention site location.

3.2.3 Relationships among Logical Operators

One other interesting observation arises when we compare the locations of $\wedge$ and $\vee$ relationships. We notice in fact that the MLP sensitive layers of $\wedge$ are a subset of the ones of $\vee$. This can be interpreted thanks to the previous results by remarking that since "and" basically links compatible tokens, and "or" incompatible ones. Therefore the intuitive logical "and" can be seen as a sub-routine of "or".

4 Conjecture on the Structure of the Intuitive Logic

The previous results enable us to make precise conjectures on the structure of each of the logical operators.

4.1 And operator

As evidenced earlier, the "and" operator is a compatibility operator, i.e. the more a and b have similar meanings, the more "a and b" is expected to appear.

The intuitive way for this operator to function would be to store the data of each compatibility class. In view of the key-value interpretation of MLP layers in transformers, it seems plausible that this data is stored in MLP layers (see next part for further justification/validation). Note that this table has be proven to be independent on the subject. Finally, combining this with the localization results, we may plausibly argue that the "and" operator might work thanks to several attention heads associated to groups of synonyms and antonyms, that indirectly make a link with the actual storage.

4.2 Or operator

As mentioned earlier, the "or" of GPT complements the logical one but is at all from a bias that causes it to be typically exclusive. For this reason it seems reasonable to think, that the or needs access to a compatibility table, which in the view of this conjecture we estimate to be precisely associated with the "and" operator.

4.3 Not

At this time, the "not" operation structure is not clear, and we would need further experiments (see next part) to come up with a rigorous idea. However in view of what was said earlier (i.e. the fact that adjectives are stored at the same place than their antonyms in the MLP layers), we can guess that the "not" operator accesses the same table, and the structure of this table may be a bit more complex than first expected.

5 Model Editing

To confirm the conjecture on the structure of the intuitive logical operations, we have started to edit the model (GPT 2-Medium) in order to alter this logic.
More concretely, we describe below one of the experiments that we have conducted in order to confirm our intuitions about "and" intuitive operator. As already said, our model is expected to complete a prompt in natural language such as *Marc is tall and* with adjectives close in meaning to "tall". The aim here is to try edit the model in order to integrate "small" as part of the synonyms of "tall". To do that, we directly apply ROME ([8]) algorithm to the key-value pair *(tall, small)* in the MLP layers identified in the localization part.
We actually observe that:

- using ROME algorithm to update the "and" table works (the probability to get "small" increases significantly, while and reaches values comparables to other synonyms);
- this modification is intrinsic since it does not depend on the subject (i.e. we can replace "Mark" with any subject);
- at this point it is not really clear whether this modification only affects the "and" operator or not (we need to further analyze the probability variations and compare with other operators in order to be able to assess the hypothetical interrelationships).

6 Discussion and Further Work

With this work, we open the door to a systematic logical approach to the study of large language patterns. We show in particular how these architectures capture aspects of the language's intrinsic semantics, for example regarding synonyms and negation. We also show how localization and editing techniques can actually shed light on the "implementation" of logical operators by highlighting in particular invariances and distributions in the model.
Some experiments, especially those related to editing, could be easily improved by a rigorous study of the underlying probability distributions and work still suffers from some weaknesses mainly related to the limited data on which the experiments were carried out. However, overall, the proposed approach offers a seemingly general framework that could be valid in broader models such as GPT-J which opens up new research perspectives including the one briefly developed below.
The logical operators presented although basic are the only elements necessary to introduce important formal tools including, for example, the system of proof called Resolution which is an inference rule that leads to a technique of proving theorems via propositional logic and first-order logic. For propositional logic, the systematic application of the resolution rule acts as a decision procedure for unsatisfiability of formulas, solving the problem of Boolean satisfiability. In light of this observation, it would be interesting to investigate in follow-up work potential hidden circuits in the combinations of logical operators introduced (perhaps starting with the resolution itself).

As for now we didn't have the time to make edit experiments to confirm or deny the conjectured structures for "not" and "or".

References

[1] Guendalina Caldarini, Sardar Jaf, and Kenneth McGarry. A literature survey of recent advances in chatbots. *Information*, 13(1), 2022.

[2] Angela Zhang, Lei Xing, James Zou, and Joseph C Wu. Shifting machine learning for healthcare from development to deployment and from models to data. *Nature biomedical engineering*, July 2022.

[3] Dennis Wei, Rahul Nair, Amit Dhurandhar, Kush R. Varshney, Elizabeth M. Daly, and Moninder Singh. On the safety of interpretable machine learning: A maximum deviation approach, 2022.

[4] Samuel N. Cohen, Derek Snow, and Lukasz Szpruch. Black-box model risk in finance, 2021.

[5] Yavar Bathaee. The artificial intelligence black box and the failure of intent and causation. *Harvard Journal of Law & Technology*, 31:889, 2018.

[6] Dan Hendrycks and Mantas Mazeika. X-risk analysis for ai research, 2022.

[7] Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks, 2021.

[8] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2022.

[9] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small, 2022.

[10] Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning, 2022.

[11] Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving, 2020.

[12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[13] Carlos Areces, Guillaume Hoffmann, and Ezequiel Orbe. Symmetries in modal logics. *Electronic Proceedings in Theoretical Computer Science*, 113:27–44, mar 2013.

[14] Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström. Certified symmetry and dominance breaking for combinatorial optimisation. In *AAAI*, 2022.

[15] Guglielmo Cinque. The fundamental left-right asymmetry of natural languages. 2009.

[16] Maja Popovic and Sheila Castilho. Are ambiguous conjunctions problematic for machine translation? 09 2019.

[17] Miguel López-Astorga. Interpretation and use of disjunction in natural language: A study about exclusivity and inclusivity. *Revista Lengua y Habla*, 25:24 – 33, 2021.

Appendix

Experimental Data

The set of the possible subjects is defined as follows:

$$\mathcal{S} := \{\textit{"Georges", "Mark", "David", "The phone", "The skyscraper", "The lunch"}\}$$

The set of adjectives for the experiment regarding weak equality is defined as follows:

$$\mathcal{V} := \{\ \textit{"kind", "loving", "friendly", "mean", "greedy", "selfish", "strong", "robust", "solid", "weak", "fragile", "powerless", "easy", "effortless", "trivial", "complicated", "difficult", "complex"}\}$$

The set of adjectives for all the other experiments (except the one concerning the abstraction involving adjectives) is defined as follows:

$$\mathcal{V} := \{\ \textit{"blue", "red", "green", "black", "white","heavy", "light", "big", "small", "short", "long", "hot", "cold","wet", "dry", "cheap", "expensive", "free", "beautiful", "repulsive", "strong", "weak", "open", "closed", "generous", "selfish", "good", "bad", "clean", "dirty", "full", "empty", "far", "close", "noisy", "quiet"}\ \}$$

The set of the adjectives for the experiment concerning abstraction involving adjectives is defined as follows: $\mathcal{V} := \{$ *"red", "green", "black", "white", "cheap", "expensive", "free", "classic", "baroque", "romantic", "modern"* $\}$

Architecture

NVidia RTX 3050 with 6GB of VRAM

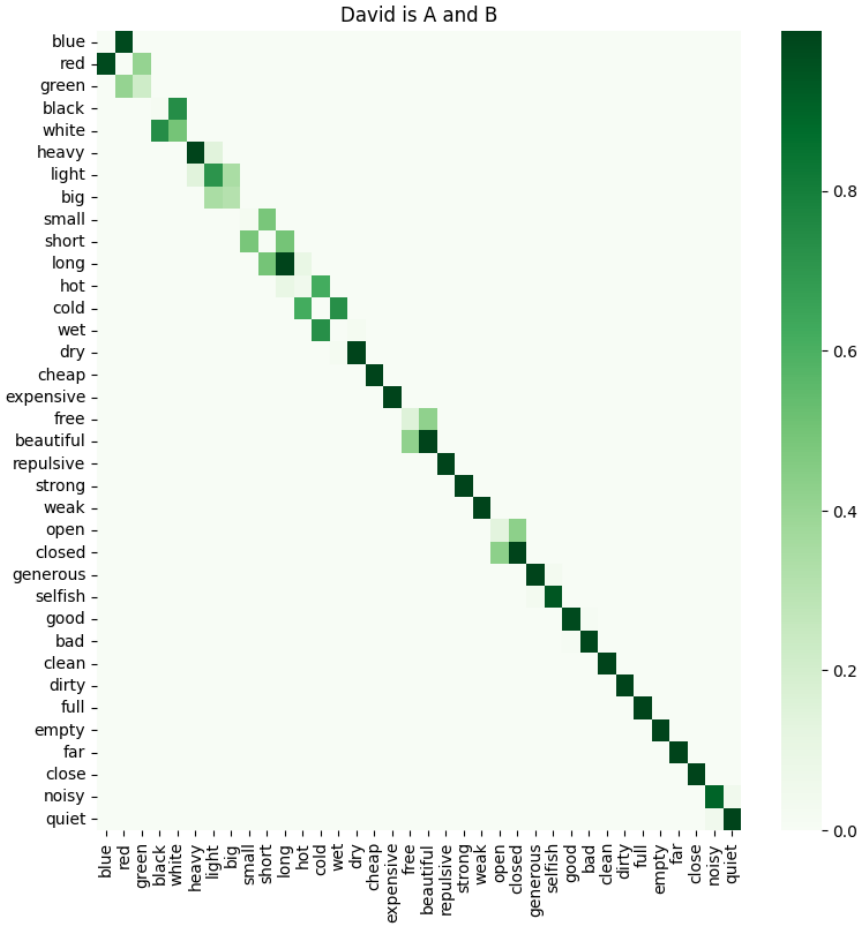

Figure 7: *Analysis of the Conjunction* On the two axes, there are the elements of $\mathcal{V}$. The color of the entries in the matrix is roughly speaking a measure of the compatibility of the adjectives labelling its coordinates.

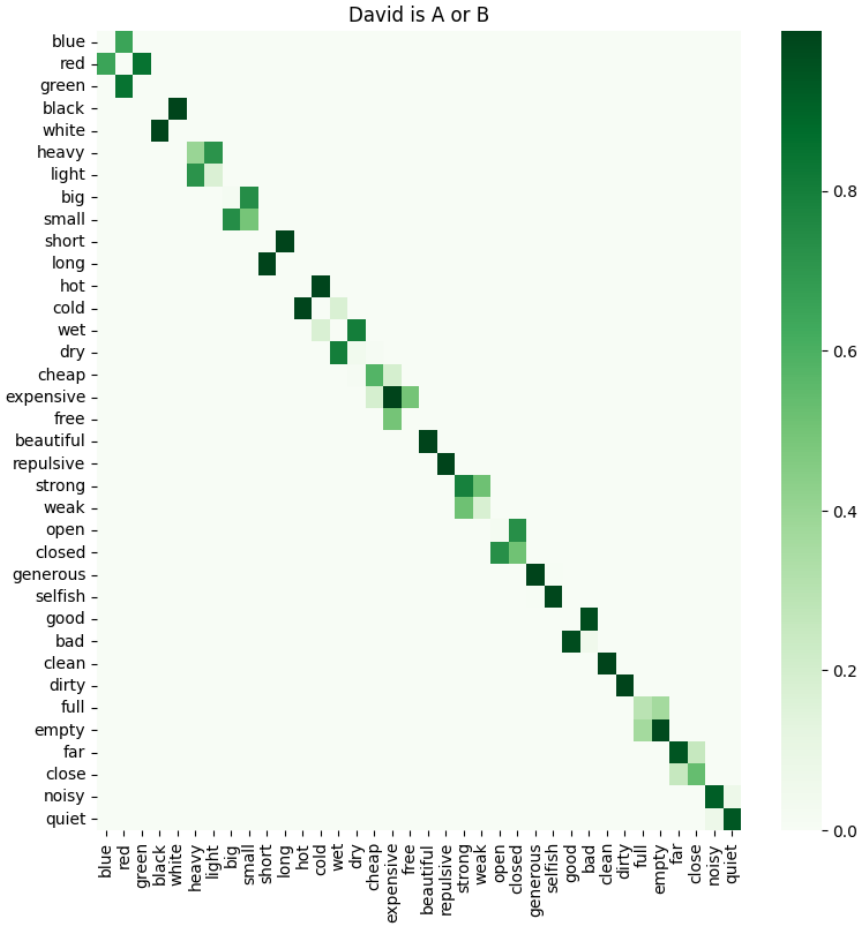

Figure 8: *Analysis of the Disjunction* On the two axes, there are the elements of $\mathcal{V}$. The color of the entries in the matrix is roughly speaking a measure of the compatibility of the adjectives labelling its coordinates.