Language Design as Information Renormalization

Ángel J. Gallego
Departament de Filologia Espanyola, Facultat de Filosofia i Lletres,
Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain

Román Orús∗
Donostia International Physics Center, Paseo Manuel de Lardizabal 4, E-20018 San Sebastián, Spain
IMIC, Ikerbasque Foundation for Science, María Dirección 3, E-48013 Bilbao, Spain and
Multiverse Computing, Paseo de Miramón 170, E-20014 San Sebastián, Spain.

Here we consider some well-known facts in syntax from a physics perspective, allowing us to establish analogies between both fields with many consequences. Mainly, we observe that the operation MERGE, put forward by N. Chomsky in 1995, can be interpreted as a physical information coarse-graining. Thus, MERGE in linguistics entails information renormalization in physics, according to different time scales. We make this point mathematically formal in terms of statistical language models. In this setting, MERGE amounts to a probability tensor implementing a coarse-graining, akin to a probabilistic context-free grammar. The probability vectors of meaningful sentences are given by stochastic tensor networks (TN) built from diagonal tensors and which are mostly loop-free, such as Tree Tensor Networks and Matrix Product States, thus being computationally very efficient to manipulate. We show that this implies the polynomially-decaying (long-range) correlations experimentally observed in language, and also provides arguments in favour of certain types of neural networks for language processing. Moreover, we show how to obtain such language models from quantum states that can be efficiently prepared on a quantum computer, and use this to find bounds on the perplexity of the probability distribution of words in a sentence. Implications of our results are discussed across severalambits.

I. INTRODUCTION

Linguistics can be defined as “the scientific study of language, and its form, meaning, and context” [1]. The field itself is a broad science, sometimes even a philosophy, embracing interdisciplinary ideas from a wide variety of contexts: syntax, mathematics, computer science, neuroscience... all in all, there is no common agreement concerning why human language is as it is, or even about its basic defining properties. From the point of view of Artificial Intelligence (AI), for instance, one is worried about developing accurate algorithms for speech recognition/prediction [2]. Additionally, the generative approach led by Noam Chomsky tries to understand the linguistic capacity from a biological perspective, as part of human cognition. As Chomsky et al. observe [3], the point of departure is Descartes’ observation that, among all animal species, only humans appear to possess a mental grammar – an “I-language,” where the “I” stands for intensional, internal, and individual – that allows us to create infinitely many meaningful expressions from a finite stock of discrete units [5, 6]. Within the generative models, the Minimalist Program [7] tries to attribute the properties of human language to what Chomsky [8] calls the “third factor”, namely “to language-independent principles of data processing, structural architecture, and computational efficiency” [8]. This picture is not different from the general study of organic systems, and D’Arcy Thompson’s and Alan Turing’s works on form and morphogenesis can be seen as an example [9]. In this framework, Chomsky proposed a basic operation, called MERGE, to build up linguistic structures [10]. In MERGE, two syntactic objects $X$ and $Y$ are combined to form a new syntactic unit $K$, i.e.,

$$MERGE : X, Y \rightarrow K = \{X, Y\}, \quad (1)$$

where the brackets mean that the information in $K$ is obtained from that in $X$ and $Y$. The operation can be applied recursively, thus having the ability to create different generations of units.

In parallel to this, physics aims to understand how the universe behaves. Some of its subfields search for the fundamental mathematical laws of the building blocks of Nature, such as high-energy physics and quantum gravity. However, the knowledge of such fundamental rules (the so-called reductionism) does not imply a priori the knowledge of the emergent laws for aggregates of many fundamental entities (the so-called emergentism [11]). Typical examples of this are condensed matter and solid-state physics, where the knowledge of the rules governing the fundamental entities at a short length scale (such as atoms and molecules described by Schrödinger’s equation) does not imply, at least directly, the knowledge of the rules governing the collective behavior of aggregates at a large length scale (such as phase diagrams of matter). In famous words of P. Anderson, “more is different” [12].

The key concept in the above discussion is that of

∗roman.orus@dipc.org
**emergence**: the collective properties of aggregates of systems may be, because of different reasons, very different from the ones of the individual systems themselves. The mathematical formalization of this paradigm in physics is achieved by the so-called Renormalization Group (RG), or simply renormalization [13]. Originally developed (mainly) by K. Wilson and L. Kadanoff, renormalization is a strategy for dealing with problems at different physical scales. This scale is typically a length, energy or time scale, and allows for different descriptions of the problem as it changes. For instance, going from short to long length scales corresponds intuitively to “zooming out” the system, effectively going from, e.g., a description of individual atoms (short scale) to a description of a solid with $O(10^{23})$ interacting atoms (long scale). At its root, a renormalization transformation is built in two steps: first, one keeps the most relevant information degrees of freedom to describe the system at a new scale discarding the ones believed not to be relevant, and second, one implements a rescaling of the physical variable and operators/functions in order to maintain the original picture. Physics is full of successful applications of renormalization in different ambits, from condensed matter physics [14] to quantum field theory [15] and quantum information [16] (each one with its own peculiarities), to the point that it has become one of the basic pillars in our current understanding of the laws of Nature. Physical theories that cannot be renormalized are considered wrong or incomplete, in the sense that they may lead to mathematical complications that are usually difficult to handle. As an example, the plain quantization of the Einstein-Hilbert action in gravity is non-renormalizable, and this has been traditionally considered an important problem.

Having said all this, our aim with this paper is twofold. On the one hand, we establish analogies between physics and linguistics in the context of the Minimalist Program (for linguistics) and emergence (for physics) which, once mathematically formalized, turn out to have important consequences in ambits as diverse as AI, theoretical linguistics, computer science, RNA / protein sequencing, quantum many-body systems, quantum computing, and beyond. We make this analogy formal for the case of statistical language models. On the other hand, we strengthen the relation between physics and linguistics, where language is the system to be understood using the machinery of physics, and of information theory in particular.

Let us be more specific: here we observe that MERGE can be understood physically as a type of information coarse-graining according to time scales. Roughly speaking, the linguistic information in sequences of words (short time scale) gets renormalized by successive MERGE operations up to meaningful sentences (long time scale) \(^1\). This simple observation, somehow trivial from a physics perspective, turns out to have deep consequences. In particular we show that language models (i.e., probability distributions over word sequences) [17], widely used in AI applications, admit a natural description in terms of Tensor Networks (TN) [18]. For instance, the simplest MERGE corresponds to a 3-index tensor of components $M_{\alpha\beta\gamma}$ accounting for a probability distribution of three variables $\alpha, \beta$ and $\gamma$. And this is nothing but a Probabilistic (or Weighted) Context-Free Grammar (PCFG), in a way to be made precise later. Probabilities of meaningful sentences with a given syntax tree are naturally given in this framework by (mostly) loop-free TNs made of diagonal tensors. Such mathematical structures have a number of nice properties which make them particularly amenable to manipulations of their information content, as we shall explain. Moreover, we show that they naturally encompass the long-range correlations that are observed experimentally in language, as well as provide arguments in favour of certain types of neural networks for language processing. Then, we move on to show that the TN structure and the particularities of PCFGs allow for the description of the probability distributions in terms of some TN quantum states. Such an exotic description using quantum mechanics is only to be understood at the practical level, but it happens to provide a useful connection between computational linguistics and quantum information and computation, opening the door to unprecedented results and developments in language processing algorithms. As examples of this, we show how such states can be built efficiently on a quantum computer, and prove lower bounds on the perplexity of the probability distribution of a set of words in a sentence \(^2\) by using mathematical tools borrowed from the theory of quantum many-body entanglement [20]. We envisage consequences of our results in several ambits. For instance, one can use the full machinery of TNs and quantum information to validate, simulate, assess, and improve current language models. Moreover, the fact that such probabilistic models can be fed into a quantum computer means that we have, in fact, a quantum algorithm that allows perfect random sampling of

---

\(^1\) A clarification is in order: here we understand “renormalization” as the tool that allows the mathematical description of the information in a system at different physical scales, accounted for by relevant degrees of freedom at every scale. Of course, the implementation of this idea in several contexts leads to different consequences. Well-known examples in physics are the existence of critical systems, critical exponents, universality classes, phase transitions, the $\varepsilon$-theorem, the reshuffling of Hilbert spaces, the $\beta$-function, fixed points, RG flows, scaling laws, relevant / irrelevant / marginal perturbations... the list is unending. In our case, however, we do not assume necessarily the existence of any of these in the case of language (though some of them may also be there), and adopt instead the most fundamental and general perspective of what “renormalization” means at its very basic core at the level of information.

\(^2\) To be defined in Eq.(17).
language, which is simply impossible with classical computing. All in all, and together with other implications, we propose that our physical picture is in fact related to the conjectured “perfect design and economy” of language in Chomsky’s Minimalist Program, as well as to the (also conjectured) efficient processing of linguistic information in the human brain [21].

The structure of this paper is as follows. In Sec.II we introduce our basic analogy, namely, that MERGE in linguistics entails information-renormalization in physics. In Sec.III we explain a direct consequence of this: the quasi-loop-free TN structure of language models. We also derive properties of such structures using tools from TNs. In Sec.IV we prove that our formalism encompasses the observed long-range correlations in language. In Sec.V we establish the novel connection to TN quantum states, and show in Sec.VI how this can be used to derive results on the perplexity of probabilistic language models. Then, Sec.VII considers arbitrary grammars and language models. In Sec.VIII we discuss some implications of our observations in different ambiits. Finally, in Sec.IX we wrap up our conclusions, include a table of the main analogies discussed in the paper, and discuss future perspectives.

II. THE BASIC ANALOGY

Our guiding principle is the following analogy, which we write in the form of an equation:

\[ \text{MERGE} = \text{Coarse-graining} \]

The left hand side of the above expression is a purely linguistic concept. MERGE is a basic operation in syntax, picking up a set of linguistic elements (such as lexical categories) and returning a new element describing the main features of the combination of the original, see Fig.1. On the other hand, the right hand side of the equation is a purely physical concept. A coarse-graining of information means the removal of superfluous degrees of freedom in order to describe a given physical system at a different scale (of, e.g., length, energy or time), see Fig.2. Combined with rescaling, it is the procedure entailing renormalization, by which the rules describing the macroscopic emerge from those describing the microscopic. The above equation establishes that both concepts are, in fact, the same basic idea but in different contexts. Chomsky’s MERGE operation entails then the renormalization of linguistic information. Moreover, this renormalization accounts for different time scales.

To understand better why this is the case, see the example in Fig.3. In terms of information processing, the MERGE operation picks up two information units at a given time scale, namely

\[ [V \text{ eat}] \text{ and } [N \text{ pasta}], \quad (2) \]

and keeps for the next time scale the most relevant information of their coarse-grained combination, i.e., that

\[ [VP [V \text{ eat}] [N \text{ pasta}]] \quad (3) \]

is a verb phrase (i.e., lexical category VP), where we used bracketed phrase markers to represent the syntax.
The linguistic MERGE operation, seen as a physical coarse-graining process. The horizontal axis is time $t$, and the vertical axis is the renormalization scale $z$. In this case, the operation takes a verb (eat) and a noun (pasta), and coarse-grains them into a verb phrase (eat pasta). At the scale $z_2$, all the relevant syntactic information is that the compound object is a verb phrase ($VP$). Unless stated otherwise, we assume that the basic building blocks are words together with their label, as shown in the grey boxes, though one could also interpret them more fundamentally as the result of a MERGE operation between a word and a set of lexical categories. For simplicity, we shall also assume here that no other information is carried by MERGE (such as genre, number, case, etc). In any case, this extra information can always be accounted for with minor trivial modifications of the scheme that we present here. The diagram provides the structure of linguistic correlations in the physical $(z,t)$ plane.

There is information loss, since at the coarsely-grained level ($VP$) we do not have, a priori, the information about the individual constituents (in this case, $V$ and $N$). The operation MERGE is non-associative, as corresponds in general to a coarse-graining $^3$. Moreover, as linguists know very well, grammar rules for individual words are not the same as those governing more complex syntagmas such as noun and verb phrases. So, we have two different descriptions of a system at different scales, and with different linguistic information units. The physical variable with different scales must be time, since language is spoken and thought as time-ordered sequences of concepts, being written language just a graphical representation of this, see the more complex example in Fig.4. Let us also stress that here we did not consider any semantic unit of information, but rather focus only on the syntax. This is the approach that we take in this paper, for the sake of simplicity of explanations. However, it would not be difficult to expand our ideas here also to include semantic information, so that sentences such as “Colorless green ideas sleep furiously” are ruled out naturally on semantic grounds.

This observation is ubiquitous in syntax and, when seen from the perspective of physics, entails the renormalization of linguistic information at different time scales. One can start by renormalizing words into sintagmas, then sintagmas into more sintagmas... an so on. Consider for instance syntax trees (or parse trees, as known in computational linguistics) like the one in Fig.4. Such analysis are of the kind linguist use to describe how different elements (words) come together in order to produce a meaningful (active or passive) sentence, and have since long been widely used in the study of language. In practice, such syntax trees are nothing but the concatenation of several MERGE operations at different scales $^4$: from words to other sintagmas, from these sintagmas to more complex sintagmas... and finally up to a sentence. According to our basic analogy, the syntax tree that one obtains from such analysis is nothing but the graphical representation of the renormalization of the (linguistic) information of a sentence. This is, how the information in different words comes together, hierarchically at different time scales, up to an emergent meaningful sentence that we can interpret semantically.

Moreover, the syntax tree also encodes the structure of physical correlations at different time scales in the sentence. More precisely, because of the local nature of MERGE, correlations in a sentence are (essentially) built locally at different time scales. Of course, it could be possible that other potentially-necessary operations in syntax, different from MERGE, introduce other dependencies (e.g., long-range movement). But still, it should be possible to codify them pictorially in the syntax tree. Worst-case scenario, such extra elements would introduce some loop in the tree. But even in such a case, the renormalization picture still holds, as we shall show in explicit

---

$^3$ Meaning that $[\alpha,[\beta,\gamma]] \neq ([\alpha,\beta],\gamma)$ using the bracketed notation.

$^4$ Other operations can be accounted for by introducing extra links in the graphical representation, as we shall explain, but the renormalization picture still holds.
Importantly, this observation is completely general, and therefore must hold for any reasonable model of language following the Minimalist Program. In particular, theoretical linguistic models trying to account for the observed rules of grammar, as well as probabilistic language models in artificial intelligence accounting for the chances of finding a given sentence in a corpus, should somehow encompass the renormalization of linguistic information. This, in turn, has deep implications in the structure of correlations in syntax. As we shall see in the next section, a direct consequence of our observation is a natural description of probabilistic language models in terms of quasi-loop-free TNS [18] accounting for different time scales.

III. TENSOR NETWORKS AND PROBABILISTIC LANGUAGE MODELS

We now consider the implications of our basic analogy for language models, i.e., probability distributions over sequences of words [17]. Such models produce probabilities $p_{w_1,...,w_n}$ for a sequence of $n$ words, represented by the random variables $w_1,...,w_n$, and are widely used in several technological ambitions such as speech recognition, machine translation, text prediction, and so forth. In practice, such probabilities are obtained by training the model (in this context, computing the frequencies of sequences) with very large corpuses of text. A priori, the model (in this context, computing the frequencies of sequences) with very large corpuses of text. A priori, the structure of the probability distribution depends on the grammatical constraints imposed by language, yet usually one assumes different types of ansatz. Here we focus on the general constraints that renormalization imposes on the structure of these probability distributions. As we shall see, a very natural description in terms of TNS just pops out, linking directly to Probabilistic Context-Free Grammars (PCFG), but not necessarily restricted to them only.

A. The MERGE tensor

To begin with, let us consider the probability distribution that two given linguistic elements $\alpha$ and $\beta$ (e.g., two words) merge into a new element $\gamma$ (e.g., some other syntagma). This probability distribution $M([\alpha, \beta] \to \gamma) = M(\alpha \cap \beta \cap \gamma)$ can in fact be described by a probability map $M$,

$$M : V_{in_1} \otimes V_{in_2} \rightarrow V_{out},$$

with $V_{in_1}, V_{in_2}$, and $V_{out}$ the input and output vector spaces. The coefficients of this map are given by a 3-index probability tensor $M_{\alpha\beta\gamma}$. The entries of this tensor are the probabilities of merging $\alpha$ and $\beta$ (the linguistic input of MERGE) into $\gamma$ (the linguistic output of MERGE). Physically, the tensor coarse-grains the variables $\alpha$ and $\beta$, at a given time scale, and retains the fundamental degrees of freedom of the common object at a different time scale. The result of this coarse-graining is variable $\gamma$.

The tensor $M_{\alpha\beta\gamma}$ obeys the usual normalization condition for probabilities,

$$\sum_{\alpha,\beta,\gamma} M_{\alpha\beta\gamma} = 1,$$

i.e., the sum of all the probabilities is equal to 1. One can also compute residual probability distributions in the usual way, i.e., by summing up over the variables that are discarded. For instance, one could have

$$M'_\gamma = \sum_{\alpha,\beta} M_{\alpha\beta\gamma},$$

with $M'_\gamma$ the residual probability distribution of obtaining $\gamma$ as the output of MERGE, no matter the input.

From a linguistic point of view, the tensor $M_{\alpha\beta\gamma}$ is the implementation, at a mathematical level, of the MERGE operation for a probabilistic language model. If the same tensor is to be used everywhere in a syntactic structure, then this is nothing but the realization of a PCFG [22], i.e., a Context-Free Grammar with probabilities assigned to its merging rules. From the perspective of physics, though, this tensor coarse-grains degrees of freedom $\alpha$ and $\beta$ at a given time scale into a new degree of freedom $\gamma$ at a different time scale. From a mathematical perspective, this tensor describes the probability of obtaining the information codified in $\gamma$ from the information in $\alpha$ and $\beta$. Regardless of the interpretation, this object constitutes the fundamental LEGO® brick of probabilistic language models following the Minimalist Program.

B. Syntactic tensor networks

Next, we notice that the structure of a syntax tree maps directly into a tensor network (TN) [18] for the probability distribution $p_{w_1,...,w_n}$ of the sentence. Specifically, every syntactic MERGE$^{[i]}$ corresponds to a 3-index...
tensor $M_{[i]}^{[1]}$, with $i$ simply a label to identify individual tensors, which could in principle be different. Now let us consider the case in which a variable $\mu$ is the result of merging $\delta$ and $\gamma$, with $\delta$ itself being the result of merging $\alpha$ and $\beta$. In such a case, following the usual mathematical treatment of probabilities, one has that the probability of obtaining $\mu$ from $\alpha, \beta$ and $\gamma$ (i.e., no matter the value of $\delta$) is given by the expression

$$p_{\mu;\alpha\beta\gamma} = \sum_{\delta} M_{\mu\beta\gamma}^{[2]} M_{\alpha\beta\delta}^{[1]},$$

i.e., we sum over all the possible intermediate events represented by $\delta$. This admits a very intuitive diagrammatic representation, see Fig.5. In that figure, every tensor is a shape and every index is a line. Open indices, i.e., those over which there is no sum, are just “free” lines, whereas sums over all the possible values of a common index between tensors are represented by lines connecting the tensors. Such sums are called contractions, i.e., in this example we just contracted index $\delta$. These type of structures, where one has a set of tensors whose indices are contracted according to some network pattern, are called tensor networks (TN) [18], and always admit a convenient diagrammatic representation as in Fig.5. With this in mind, we arrive to the conclusion that syntax trees of sentences map into TNs of MERGE tensors $M_{\alpha\beta\gamma}^{[2]}$, at the level of probabilistic language models. We call such structures syntactic TNs.

Let us be more precise: if the syntax tree does not have long-range dependencies (i.e., it is made only of MERGEs), then the TN is loop-free and corresponds generically to a Tree Tensor Network (TTN) [23], see Fig.6. If the MERGEs are sequential in time, then the TN is in fact a special case of TTN called Matrix Product State (MPS), see Fig.7 [24]. These two types of structures appear quite often in the study of strongly correlated classical and quantum lattice systems in one spatial dimension [18, 23, 24] as well as in tensor calculus [25], and their properties are very well known by physicists and mathematicians. Moreover, if the syntax tree has some long-range dependency (e.g., movement, agree, c-command...), then this introduces some extra index in the network, correlating variables at different positions, and therefore introducing some loop in the diagram. To be precise, such extra index correlates the (perhaps distant) probability distributions for such variables, and can normally be casted into redefined tensors in order to keep the overall tree structure, as shown in the figure. As an example, this is in fact the case of the so-called CHAINS, which we mentioned in the introduction (Sec.I), and where a lexical object is intrinsically interpreted in different contexts of a sentence but only externalized in one of them, see Fig.8 for an example. More intricate cases, such as those involving a concatenation of chains (the so-called successive cyclicity), can also be accounted for similarly, see Fig.9 for an example. At any rate, though, the number of loops in the TN is always quite small, as long as the syntax tree is based on a Phrase-Structure (Constituency) Grammar [26], such as PCFGs. For the sake of clarity we restrict our explanation to these grammars. Other plausible situations, such as those arising in Dependency Grammars [27], will be briefly discussed in Sec.VII.

The syntactic structure of a sentence implies, therefore, that correlations in its probability distribution are orchestrated according to a (mostly loop-free) TN of MERGE tensors, which organize the degrees of freedom according to different time scales.
FIG. 8: (Color online) Syntactic TN for the sentence “Should Einstein play violin?” as an example of syntactic movement. The element “Should” is created at the position of $t_k$ but externalized at the position of $T_k$ (hence it “moved”). At the level of the TN, this can easily be accounted for by an extra correlation between these two positions, i.e., an extra link between them (and perhaps two new tensors, as shown in the figure). This introduces a loop in the TN. However, as shown in (b), it is possible to redefine the overall structure as a loop-free TN with tensors as those shown in the dotted red boxes, and reshaped (or fused) tensor indices (i.e., whenever there are two indices together, fuse them into a single big index).

FIG. 9: (Color online) Syntactic TN for the sentence “Ángel seems to be likely to come today”, as an example of concatenation of chains, or successive cyclicity. The syntactic information of the element “Ángel” is at different places leaving traces $t_k, t'_k,...$, but externalized at only one position $T_k$. At the level of the TN this can be easily accounted for by an extra index correlating different sites, similarly as in Fig.8.

C. Properties

Let us now enumerate some important properties of the probability structures that we just found, which come out naturally from their TN description. Some of them were already mentioned briefly, but we revisit them again for clarity:

1. Locally-built syntactic correlations at every scale

Correlations in the probability distribution are built locally at every renormalization time scale by MERGE. Distant parts of the sentence become correlated at long time scales (i.e., up in the syntax tree), whereas those that are close become correlated at short time scales (i.e., down in the syntax tree). This locality implies a nice feature of loop-free syntax trees: for a sentence with $n$ words, there are always exactly $n - 1$ merged objects. Translated into syntactic TNs, this means that if the TN has $n$ indices at the first renormalization scale $z_1$ (i.e., those corresponding to the words in the sentence), then there will be exactly $n - 1$ indices on the whole at higher renormalization scales $z_m, m > 1$. This can be easily checked by inspection in all the figures with loop-free syntax trees and syntactic TNs of this paper. The consequence is that to specify the full syntax of a typical sentence of $n$ words, one requires on the whole $2n - 1$ units of syntactic information. In the case of having TNs with loops, as in the case of long-range movement in Fig.8 and Fig.9, the index creating the loop establishes a correlation between distant positions in the sentence, some of them with only syntactic information and no word present (the so called traces). In such cases it is clear that the number of required syntactic information units is larger than $2n - 1$, though not much larger.

2. Diagonal tensors and correlated factorization

The TN, when contracted from up towards down, reproduces the different probability distributions of the linguistic variables at every renormalization time scale. In other words, the TN encodes the probabilities of the relevant degrees of freedom at all possible time scales. Moreover, it is possible to obtain the residual probability of any of the variables just by contracting all the rest. Quite importantly, in syntactic TNs one does not even need to perform any tensor contraction since, once the sentence is fixed or partially fixed, there is a correlated factorization of the whole TN because of the way human language turns out to be, which we explain in what follows. A well-known fact in grammar is that the output of a MERGE operation is always uniquely determined by its input. This is, given two objects being merged, there is only one possible output, no matter the context. This is a simple observation about how human language seems to work: the human brain does not merge an adjective $A$ and a noun $N$ into an object that sometimes behaves like a noun phrase $NP$, and sometimes like an adjectival phrase $AP$. Instead the combined object behaves always
like a noun phrase \( NP \). So, given the input of MERGE, its output becomes fixed uniquely \(^5\).

This turns out to have an important consequence for us: MERGE tensors are diagonal. As a consequence, once the sentence is given, or partially given, then the TN factorizes in a correlated way. To see why this is so, notice that if the output of MERGE is always uniquely determined by its input, then all the indices in the syntactic TN become fixed once the indices at the shortest time scale are fixed, i.e., once a specific sentence is given. Because of this, the probability of a specific sentence actually factors out in terms of correlated probabilities and no TN contraction is needed at all. The overall correct syntactic structure of the sentence is the global, non-local property that correlates all the probabilities amongst themselves. Moreover, the residual probability of, say, finding a specific word in a sentence that is partially given, can be easily computed using one MERGE tensor only, which contains information about both the immediate neighborhood of the word, as well as about the overall syntactic neighborhood, see Fig.10. This is a very remarkable property that has its roots in the peculiarities of human language. In particular, it implies that the calculation of probabilities is extremely efficient, and that if the correct syntactic structure of a sentence is fully or partially known, then the statistical perplexities of reduced probability distributions are remarkably low, as we shall discuss in more detail in the forthcoming sections.

For a given sentence, therefore, the formalism produces a correlated structure of 3-index tensors linking all possible renormalization scales, see Fig.10. For example, the overall probability of, e.g., the 4-word sentence “Román plays his guitar” (an actual possibility in Fig.10) reads

\[
p_{w_1^*, w_2^*, w_5^*} = M_{w_1^*, VP, S}^3 M_{w_2^*, NP, VP}^2 M_{w_3^*, w_4^*}^1 N P,
\]

where \( w_1^* , \ldots, w_4^* \) are the fixed words of the sentence, and no tensor contraction is needed at all. The above equation is a correlated product of coefficients from 3-index probability distributions, which encode all the syntactic information of the sentence at all time scales. The effect of this is more dramatic when it comes to residual probabilities: consider for instance predicting the word “drank” in the sentence “The man John met yesterday drank Japanese whisky”. A 3-gram model \(^{28}\) (a rather common option in speech recognition) would give a probability distribution such as

\[
p_{w_1^*, w_5^*, w_6} = 3 \text{ - gram model},
\]

i.e., correlating the word \( w_6 \) only to “met” and “yesterday”. The predictive power of this distribution is thus not very good, because there is no use whatsoever of the syntactic information from the rest of the sentence. However, in our TN description, the residual probability distribution, as shown in Fig.11, is given by

\[
M_{w_6, NP, VP}^{[6]} \quad \text{Syntactic TN model},
\]

which includes all the relevant syntactic information of the environment needed to predict \( w_6 \) in the sentence.

In other words, having \([NP [A] \text{Japanese}] [N \text{whisky}]\), the rest of the sentence imposes that whatever goes in \( w_6 \) needs to combine together with this \( NP \) necessarily into a verb phrase \( VP \). To put it simply, the marginal probability distribution is governed by this question: with whom do I merge to form what, as constrained by the rest of the sentence? In hindsight, this description includes all the relevant syntactic information required to predict the word exactly at that point.

From the above derivations, it is clear that all probabilities can be computed very efficiently, and exactly, from the TN. To be more specific, the fact that the structures are mostly loop-free implies that the calculation of probabilities, which amounts to the contraction of the tensors in the TN, can be done in polynomial time in the number of words, i.e., \( O(\text{poly}(n)) \) \(^{18,23,24}\). From the perspective of complexity theory, this is a consequence of the fact that the contraction of a loop-free TN is a problem in the complexity class \( \text{P} \) \(^{29}\) \(^6\). But in the case of syntactic TNs like the ones described here, the situation is even

\(^5\) Notice that the converse is not true.

\(^6\) Unlike some TNs with loops, which are \( \text{2P-hard} \) and therefore need an exponential time to be contracted \(^{31}\).
The two-point correlation function does not equal to the actual separation distance in the sentence or tree (i.e., their separation in the network), very by MERGE implies a separation distance in the syntax relation between two words in the sentence decays exponentially of loop-free TNs, and in our case it means that the correlation between two words is the actual separation distance in the syntax tree, i.e., somewhere up in the tree. It is possible to prove mathematically [18, 23, 24] that this correlation function decays asymptotically as

$$C(i, j) \approx e^{-d(i, j)/\tau} \quad \text{for} \quad d(i, j) \gg \tau, \quad (13)$$

with $d(i, j)$ the size of the path between $w_i$ and $w_j$ in the syntax tree, and $\tau$ a sentence-dependent (finite) correlation time, see Fig.12 and Fig.13. As is well known from the theory of TNs, parameter $\tau$ does not depend on the choice of functions $f(w_i)$ and $f'(w_j)$, so it depends only on the type of sentence and the MERGE probabilities. This conclusion also holds if the TN has a small number of loops. Importantly, the quantity $d(i, j)$ can depend a lot on the type of syntax tree that one has. Consider for instance the two examples “Noam drives the car”, and “The man from Boston drives well the car”, with syntax trees as in Fig.12 and Fig.13. In the first case, Fig.12, the syntax tree is purely sequential, and therefore the TN for the probability distribution is a Matrix Product State [24]. In such a case it is clear that the distance $d(i, j)$ between two words is the actual separation distance in the sentence, i.e., $d(i, j) = |j - i|$. However, in the second case, Fig.13, the syntax tree is a binary tree, and therefore the corresponding TN is a Tree Tensor Network [23]. In such a case, the path along the tree between two words in the sentence necessarily goes also along the vertical axis, and one can prove that it is given by $d(i, j) \approx \log_2 |j - i|$, again with $|j - i|$ the separation in the sentence, and where $\approx$ means that it is correct up to some possible additive constant term $7$. Therefore, in cases such as the one in Fig.12 (“Noam drives the car”), the correlation function between two words will behave like

$$C(i, j) \approx e^{-|j-i|/\tau} \quad \text{for} \quad |j - i| \gg \tau, \quad (14)$$

whereas in cases such as Fig.13 (“The man from Boston

$^7$ To be precise, this is correct in average, since depending on the tree it is possible to choose specific pairs of points with longer separation [23].
drives well the car”) it will behave like

\[
C(i, j) \approx e^{-\log_2 |j-i|/\tau} \approx \frac{1}{|j-i|^{1/\tau}} \quad \text{for} \quad |j-i| \gg \tau.
\]

(15)

In both cases the correlation falls down towards zero with the separation distance $|j-i|$ in the sentence, but in the first case it decreases exponentially fast, whereas in the second it is polynomially fast, and therefore much slower than in the first case. Notice, however, that at the level of renormalized objects up in the syntax tree, the two situations are completely equivalent, see Fig.13. This means that the correlation functions for the second sentence (Fig.13), but at some longer time scales (i.e., at the level of renormalized syntagmas up in the tree), decay exactly in the same way as those in the first sentence (see Fig.12).

Three remarks are in order. First, in intermediate situations between those described by the two examples above we expect also an intermediate regime between the two limiting cases from Eq.(14) and Eq.(15), but always obeying Eq.(13) asymptotically. Second, notice that the correlation time $\tau$ measures roughly how fast these correlations decay: the shorter $\tau$ is, the faster they decay towards zero. And third, notice that Eqs.(13), (14) and (15) essentially imply that language, at least within this description, has always very short-distance correlations within the syntax tree, which does not necessarily imply short-distance correlations in the separation distance within a sentence, since these two distances are different as shown in Eq.(15). In fact, we will show in Sec.IV that for actual separation distances in the sentence one finds that, on average, language has long-range correlations, i.e., they decay polynomially, in turn matching the experimental observations [30]. Similar conclusions apply as well in the case of having a small number of loops in the network, e.g. in linguistic chains, or in situations such as the German language, where a word correlated with the beginning of the sentence is actually sent to the

4. Positivity

By construction, the syntactic TNs presented here are such that all the tensors are non-negative, i.e., they are made entirely of non-negative coefficients. This is because of the stochastic nature of the MERGE tensor, which has been defined in terms of probabilities. These are sometimes called stochastic tensor networks. It is well known that such positivity restriction on the coefficients of the tensors is in fact very stringent [32] and usually implies a very large dimension for the vector spaces of the coarse-grained variables in the TN. We may therefore expect that a TN description of the overall probability distribution in terms of non-negative tensors lowers down this dimension, thus making the representation computationally more efficient. The price to pay, however, is that we loose the interpretation of the MERGE tensor as a tensor of probabilities. Still, a non-positive TN may be computationally more convenient in some situations.

D. Refinement levels

The TN structure of MERGE tensors that we just described admits different levels of refinement, when it comes to determining the actual probability of a sentence in a given language model. A practical evaluation of such
merging probabilities, once a parsed corpus (a Penn TreeBank) is given, proceeds as follows:

(i) First, one does a frequency count of all the words, and computes the probability of being some lexical category \((N, V, \text{etc})\) conditioned to being a certain word. This probability distribution corresponds, formally, to a set of words and a set of lexical categories, as mentioned briefly in the caption of Fig. 3. In practice, though, it can be accounted for by a 2-index probability matrix \(M_{a,b}\), with the first index referring to a particular word, and the second to its lexical category.

(ii) Second, one considers every sentence in the corpus and the respective syntax tree, and computes the probabilities corresponding to the coefficients of the MERGE tensors. This is done by counting the frequency of how many times two given lexical elements merge into a given object. Quite importantly, there are (at least) four different levels of refinement of the computed tensors, depending on their position in the \((z, t)\) plane and the structure of the syntax tree. In increasing order of refinement, these are:

1. One single MERGE tensor \(M\) for all possible positions in the \((z, t)\) plane.
2. One MERGE tensor \(M^{[z]}\) for each possible renormalization scale, each one for all possible positions in \(t\) at the corresponding scale.
3. One MERGE tensor \(M^{[z:t]}\) for each possible position in the \((z, t)\) plane.
4. One MERGE tensor \(M^{[T,z,t]}\) for each possible position in the \((z, t)\) plane, and for each possible syntax tree \(T\).

The more refined the information included in the computed MERGE tensors, the more accurate is the probability distribution, and therefore the better is the language model. The first of the refinement levels described above corresponds to the probabilistic language models provided by PCFGs [22]. These models are known to work reasonably in some circumstances, although on average not as good as, say, \(N\)-gram models [28]. But this is understandable, because one does not expect a priori the same MERGE tensor at all the possible positions in the \((z, t)\) plane. Importantly there are still three more levels of refinement, which should account for better models. The second level drops the assumption of “ancestor-free” (akin “scale invariance” in physical jargon), so that the tensors may depend on the scale \(z\). The third level drops, additionally, also the assumption of “place invariance” (akin “translation invariance” in physical jargon), so that the tensors may also depend on the variable \(t\). Finally, the fourth level of refinement drops the assumption of the MERGE tensors being tree-independent. In principle, the four refinement levels are computable from a TreeBank, implying increasing level of precision for the language model. As for the computational cost of retrieving the MERGE tensors, in the first three levels it should be \(O(M\bar{n})\) both for time and memory, with \(\bar{n}\) the average number of words per sentence in a corpus containing \(M\) sentences. In the fourth case, however, the time cost is also the same but the memory cost may be larger since, for a large text, we expect to find roughly all possible syntax trees for every sentence length, which for \(n\) words is in turn given by the \((n-1)\)th Catalan number,

\[
C_{n-1} = \frac{(2(n-1))!}{n!(n-1)!} \approx \frac{4^n}{\sqrt{\pi n^{3/2}}} \left(1 + O\left(\frac{1}{n}\right)\right), \tag{16}
\]

where the approximation is in the limit \(n \gg 1\), and therefore scales exponentially. However, typical sentences in human language do not usually imply a dramatically-large number of words (we elaborate more on this in Sec. VIII), and therefore the number of different syntax trees to be stored in memory may not be as large in practice as the above number.

Once the MERGE tensors have been computed from the TreeBank, the numerical probability for a sentence of \(n\) given words can be obtained in a two-step process:

(i) First, compute the possible syntax trees of the sentence (there may more than one valid tree in ambiguous cases).

(ii) Second, evaluate the probability for each tree following the correlated factorization procedure explained in the previous section, according to the four refinement levels mentioned above. The overall probability is the sum of probabilities for each valid syntax tree.

As the probabilities are computed, it is possible to calculate standard benchmark measures of language models, such as the so-called perplexity \(P\),

\[
P = 2^{H(p)} = 2^{-\sum_{w_1, \ldots, w_n} \log_2 p_{w_1, \ldots, w_n}}, \tag{17}
\]

with \(H(p)\) the Shannon entropy of the probability distribution. The lower the perplexity, the more peaked is the distribution and thus the better it predicts the sample. So, the better the language model, the lower its perplexity, at least a priori. In our case we also expect the perplexity to decrease substantially as the refinement of the coefficients of the MERGE tensors increases, according to the four refinement levels mentioned above. Moreover, the perplexity also goes down with the precision of the probabilities in our MERGE tensors. We prove these points in Sec. V., using at some steps a novel reformulation of language models in terms of quantum states.

IV. LONG-RANGE CORRELATIONS IN LANGUAGE

How are correlations, on average, for language? It turns out that we can provide an answer for this question using the results that we presented so far. We saw
in Eqs.\((13, 14, 15)\) that correlation functions \(C(i, j)\) between two points \(i\) and \(j\) in a sentence decay exponentially with the syntactic distance \(d(i, j)\), i.e., the distance between positions \(i\) and \(j\) in the network. And this distance may be the actual distance in the sentence, as in Fig.12, or not, as in Fig.13.

In practice, it is clear that all languages tend to have more structures like the one in Fig.13 than those like in Fig.12. The simple reason for this is that MERGE favours the formation of tree-like structures. So, if we were to count the number of possible tree-like sentences in a language (i.e., like those like Fig.13), they would certainly outnumber the “linear” structures (i.e., like those in Fig.12). The conclusion from this is that the average syntactic distance \(d(i, j)\) over all the sentences of a given language, obeys

\[
\overline{d(i, j)} \approx \log_2 |j - i|, \tag{18}
\]

where the overline means “average”. This must be the case, simply because most of the syntactic structures are trees as in Fig.13. In turn, this implies that the average correlation function \(C(i, j)\) behaves like

\[
\overline{C(i, j)} \approx \frac{1}{|j - i|^{1/\tau}}, \tag{19}
\]

with \(\tau\) correlation time that depends only on the average over all sentences of the specific language being analyzed. Such a behaviour is also inherited by other quantities like the mutual information \(I(i, j)\), since it satisfies the inequality [33]

\[
I(i, j) \geq a \times C(i, j)^2, \tag{20}
\]

with a some constant prefactor. Thus, on average and for the mutual information one finds

\[
\overline{I(i, j)} \geq a \times \frac{1}{|j - i|^{2/\tau}}, \tag{21}
\]

meaning that it also decays at least polynomially with the separation distance \(|j - i|\) in the sentence, and with a characteristic correlation time \(\tau/2\).

The result we just found in Eqs.\((19, 21)\) deserves some discussion. First, notice that we obtained these expressions exactly. They therefore imply that languages, on average, have a polynomial decay of correlation functions and mutual information between two points. Importantly, this has been observed in experiments and numerical analysis of specific texts and languages, and is what people often call “long-range correlations” in language [30]. Our results thus provide a first intuition on the origin of long-range correlations in language. As we can see here, they are a natural consequence from syntactic trees, since tree-like structures display this type of correlations. In long pieces of texts, it is expected also that the information in sentences organizes also in some sort of hierarchical higher-level structures (sections, chapters, etc) due to semantic/pragmatic constraints. We argue, therefore, that may actually be the reason for long-range correlations observed in long pieces of text including many sentences.

Second, notice that some properties of Eqs.\((19, 21)\) are language-independent, and therefore universal, whereas other are language-dependent. In particular, the average polynomial decay is universal. However, the characteristic time scale for this decay, namely exponent \(\tau\), is language-dependent. The number \(\tau\) therefore provides a quantitative way to classify different texts and different languages. Those texts and languages with similar syntactic structures will have similar values of \(\tau\), and is therefore natural to think that they belong somehow to the same family, i.e., probably coming from the same ancestor (e.g., Spanish, Italian and French all coming from Latin). And this is important, because such an analysis may be a way to provide hints about the genealogical (or “genetic”) root of languages for which we do not know their origin [34], such as Basque, Sumerian, and others.

V. LANGUAGE MODEL QUANTUM STATES

Let us now define the following quantum state:

\[
|\Psi(T_n)\rangle = \frac{1}{Z(T_n)^{\frac{n}{2}}} \sum_{w_1,\ldots,w_n} (p_{w_1,\ldots,w_n})^{\frac{1}{2}} |w_1,\ldots,w_n\rangle, \tag{22}
\]

with \(p_{w_1,\ldots,w_n}\) the probability of a sentence with words \(w_1,\ldots,w_n\) and syntax tree \(T_n\), and \(\{|w_1,\ldots,w_n\}\) an orthonormal (tensor product) basis of some Hilbert space for \(n\) parties, each party corresponding to the position of a word in the sentence. The dividing normalization factor \(Z(T_n)\) is actually the partition function of the probability distribution, i.e.,

\[
\langle \Psi(T_n)|\Psi(T_n)\rangle = \frac{1}{Z(T_n)} \sum_{w_1,\ldots,w_n} p_{w_1,\ldots,w_n} = 1. \tag{23}
\]

We call the state in Eq.\((22)\) a language model quantum state.

Because of the correlated factorization of syntactic TNs explained in previous sections, one can see easily that these language model quantum states admit a TN representation of their coefficients, i.e., they are really TN states in the strict quantum-mechanical sense. The TN structure of the coefficient \((p_{w_1,\ldots,w_n})^{\frac{1}{2}}\) is simply given by the same one as for the probability distribution \(p_{w_1,\ldots,w_n}\) (the syntactic TN), but replacing every coefficient of a MERGE tensor by its square root. More specifically, it is the same TN but with 3-index tensors \(A^{[i]}\) of coefficients

\[
A^{[i]}_{\alpha\beta\gamma} = \left(M^{[i]}_{\alpha\beta\gamma}\right)^{\frac{1}{2}}, \tag{24}
\]

again with \(i\) simply label for the different tensors. This simple prescription is a direct consequence of tensors being diagonal in the syntactic TN. Notice also that these
tensors obey the condition

\[ \sum_{\alpha,\beta} A^{[i]}_{\alpha\beta\gamma} (A^{[i]}_{\alpha\beta\gamma})^* = \left( \sum_{\alpha,\beta} M^{[i]}_{\alpha\beta\gamma} \right) \delta_{\gamma\gamma'} = p^{[i]}_{\gamma} \delta_{\gamma\gamma'}, \tag{25} \]

with \( p^{[i]}_{\gamma} \) the probability of merging at position \( i \) any two given lexical objects into \( \gamma \), and \( \delta_{\gamma\gamma'} \) the Kronecker delta, see Fig.14.

A. Properties

The language TN quantum state that we just defined is interesting for a number of reasons. These are described in what follows.

1. Truly random sampling

First, notice that if this quantum state becomes (somehow) experimentally available in an actual quantum system, then it can be used to do truly random sampling of the probability distribution of sentences with that particular syntax tree. For comparison, all classical samplings are based on pseudo-random number generators, which are known to induce errors in the long run for, e.g., Monte Carlo methods. The state could also be useful, for instance, to find the most-likely sentences in a language model, and things alike. Notice that, here, we understand by “perfect sampling” the fact that the sampling does not introduce spurious correlations caused by a pseudo-random number generator. Still, near-term quantum devices are known to be noisy, which could also affect this property if implemented in practice.

2. Language model quantum circuit

Second, the state can, in fact, be created by a quantum circuit with as many two-body gates as \( A \)-tensors. The procedure is sketched in Fig.15: starting from the shortest renormalization scale \( z_1 \), one reshapes the indices of the \( A \)-tensors as a matrix and performs a QR decomposition [35], as shown in the figure. Since the \( A \)-tensors are real and positive, the matrix \( Q \) is orthogonal, i.e., \( Q^T Q = I \). Reshaping back \( Q \) as a 3-index tensor provides an isometric tensor, which we keep at the particular sites of the network at that renormalization scale. Matrices \( R \), however, are contracted with the \( A \)-tensors at the next renormalization scale \( z_2 \), see Fig.15. The resulting tensors, call them \( B \), are then also QR-decomposed, where the \( Q \)s define again isometries, which we keep in the network, and the \( R \)s are contracted with the \( A \)-tensors at the next renormalization scale. By iterating this process up to the top level, one gets a TN of isometric 3-index tensors \( Q^{[i]} \), and a quantum state \( |\Omega\rangle \) at the very top carrying non-local information about the probability of the whole sentence. In particular, since tensors \( Q^{[i]} \) are isometries, one has that

\[ \langle \Psi(T_n)|\Psi(T_n)\rangle = \frac{1}{Z(T_n)} \langle \Omega|\Omega \rangle = 1, \tag{26} \]

(where the last equality follows from the normalization of the state), and therefore

\[ \langle \Omega|\Omega \rangle = Z(T_n) = \sum_{w_1,\ldots,w_n} p_{w_1,\ldots,w_n}, \tag{27} \]

which means that the norm of the quantum state \( |\Omega\rangle \) is the overall probability of having an \( n \)-word sentence (whichever) with syntax tree \( T_n \) in the language model. This global information just moved up to the top level of the TN and, importantly, we constructed it locally at
every renormalization scale by a sequence of $QR$ decompositions, therefore very efficiently (notice that we never needed to compute each one of the terms $p_{w_1 \cdots w_n}$ individually!) 8. Connecting to the usual developments in quantum-mechanical TN states, this is an example of an isometric TTN state [23]. Finally, in order to promote this structure to a quantum circuit, we simply notice that an isometric tensor can be understood as a two-body unitary gate, where one of the indices is fixed to some ancillary state $|0\rangle$ [16], see Fig.16. The resulting diagram is nothing but the picture of the quantum circuit producing the desired quantum state. The conclusion is that if the MERGE tensors are given, then one could in principle produce these quantum states efficiently in a quantum computer or a quantum simulator. Last but not least: the description above has been for TNs without loops, but it can be generalized to other situations. In case of having a small number of loops in the network (e.g. in CHAINS), there is also a similar procedure as the one indicated here by playing with several tensor decompositions ($QR$, Singular Value Decomposition, etc), always sending the non-unitary parts upwards in the syntactic network. We envisage that this may trigger applications of near-term quantum processors for computational tasks related to language.

VI. PERPLEXITY FROM ENTANGLEMENT

An interesting application of the language model quantum states defined in the previous section concerns the perplexity $P$ of a language model, which was defined in Eq.(17). For a given sentence, it turns out that we can give lower bounds on the perplexity of a given subset of words, using tools from quantum information theory, as we show next.

Let us start by considering a sentence with $n$ words, and a subset of $n' < n$ contiguous words within the sentence. These are a block of $n'$ words. The question we want to answer now is: how much is the entanglement of this block of $n'$ words in a given quantum state $|\Psi(T_n)\rangle$ for a syntax tree $T_n$? Following the usual procedure for bipartite entanglement, we get first the reduced density matrix of the block,

$$\rho(n') = \text{tr}_{n-n'} |\Psi(T_n)\rangle \langle \Psi(T_n)|,$$

with $\text{tr}_{n-n'}(\cdot)$ the partial trace over the rest of the system (the environment). As shown in the diagrams of Fig.17, this can be achieved by “cutting” out the relevant sub-tree linking the $n'$ words from the rest of the sentence. After the appropriate contractions, this reduced density matrix can always be written as

$$\rho(n') = \frac{1}{Z(T_n)} W X W^\dagger$$

with $W$ some rectangular matrix amounting for the contraction of the sub-tree for the block, and $X$ a square matrix whose rank is the number of lexical categories $N_l$ in our grammar, being this also the rank of $\rho(n')$, see Fig.18. It is easy to see, moreover, that in fact matrix $X$ is diagonal,

$$X_{\alpha \alpha'} \propto p(n-n')\alpha \delta_{\alpha \alpha'},$$

with $p(n-n')\alpha$ the overall probability of the string of $n-n'$ words merging into lexical category $\alpha$, no matter the words in the string. One can also see that the (unnormalized) eigenvectors of $\rho(n')$ are given by

$$(v_\alpha)_\omega = (W^\dagger)_{\omega \alpha},$$

with $(v_\alpha)_\omega$ the $\omega$-coefficient of the $\alpha$ eigenvector, and eigenvalues $\lambda_\alpha$ given by

$$\lambda_\alpha = p(n')\alpha \cdot p(n-n')\alpha.$$
with \( p(n - n')_\alpha \) as described above, and similarly for
\( p(n')_\alpha \) but for the set of \( n' \) words, see Fig.19. Using
Eq.(32), one can get the entanglement entropy \( S(\rho(n')) \) and the
single-copy entanglement \( E_1(\rho(n')) \) of the block of
\( n' \) words [36], which are given respectively by

\[
S(\rho(n')) = - \sum_\alpha \lambda_\alpha \log_2 \lambda_\alpha
\]

\[
E_1(\rho(n')) = - \log_2 \left( \max_\alpha \lambda_\alpha \right) .
\]  

(33)

The above entanglement measures obey the chain of inequalities

\[
E_1(\rho(n')) \leq S(\rho(n')) \leq \log_2 N_l,
\]  

(34)

which implies that the entanglement of the block can
never be too large, since the number of lexical categories
\( N_l \) in a typical grammar for human language is usually
quite small.

Next, we notice that the probability distribution \( p_\alpha \) for
the \( n' \) words in the block is actually given by the diagonal
elements of \( \rho(n') \) in the basis of Eq.(22) restricted to the
block, i.e.,

\[
p_\alpha = \rho(n')_{\alpha\alpha} .
\]  

(35)

One can check from the derivations above that this prob-
bability distribution and the one of the eigenvalues \( \lambda_\alpha \)
obey the majorization relation [37]

\[
\vec{p} < \vec{\lambda},
\]  

(36)

which implies

\[
H(p_\omega) \geq S(\rho(n')),
\]  

(37)

i.e., the Shannon entropy of the reduced probability distri-
bution for the block of \( n' \) words is larger than the entan-
glement entropy of the block. This relation, combined
with Eq.(34), implies directly that

\[
\mathcal{P} = 2^{H(p_\omega)} \geq 2^{S(\rho(n'))} \geq 2^{E_1(\rho(n'))},
\]  

(38)

with \( \mathcal{P} \) the perplexity of the distribution of the \( n' \) words
as defined in Eq.(17). Combining this with Eq.(32) and
Eq.(33), in the end we arrive to the result

\[
\mathcal{P} \geq \min_\alpha \left( \frac{1}{p(n')_\alpha p(n - n')_\alpha} \right),
\]  

(39)

which is our main lower-bound for the perplexity of the
probability distribution of the \( n' \) words.

Some remarks are in order. First, notice that Eq.(39) is
a fully classical result, even though we used the machin-
ery of quantum information theory to find it. Second,
the inequality is giving us a fundamental lower bound on
how well our language model can predict sentences, just
because of its statistical nature. Third, we can roughly
estimate the scaling of this lower bound: if \( p_{\text{max}} \) is the
maximum merging probability over all MERGE tensors
in the network, it is easy to see that

\[
\mathcal{P} \approx \left( \frac{1}{p_{\text{max}}} \right)^{n-1}
\]  

(40)

which implies, also roughly, that the perplexity gets
worse (increases) exponentially fast with the number of
words \( n \) in the sentence, but also that it improves (de-
creases) exponentially fast if the MERGE probabilities of
the language model get more refined and accurate. This
inequality shows clearly the route required in order
to improve the performance of syntax-based probabilistic
language models: short sentences, and accurate probabil-
bilities.

VI. ARBITRARY GRAMMARS AND
LANGUAGE MODELS

We would like to say a couple of words about other
types of grammars, not necessarily context-free, as well
as other language models. Importantly, the tensor net-
work picture of language is not necessarily restricted to
the cases that we presented above, and in fact can be used
to describe the correlation structure of, essentially, any
type of grammar and/or language model. For instance,
the trees of dependency grammars [27], though not based
on the MERGE operation, also admit a TN representation of their correlations when put as a probabilistic language model. We could even add long-range dependencies between the probability distributions in constituency grammars, as was shown for the case of chains in Fig. 8, but which can in fact be generalized over the whole (z, t) plane, obtaining what is known in physics as a MERA-like tensor network [16], see Fig. 20. As a matter of fact, it would be possible to model with TNs any grammatical correlation structure, even if not directly linked to human language. An example would be a syntactic structure based on an hypothetical MERGE operation with multiple outputs for a given input. Such structures would not have the property of “correlated factorization” discussed above, but most of the key properties that we mentioned would still hold, including those related to computational efficiency and short-range syntactic correlations.

From a practical perspective, the so-called N-gram models [28], where the probability of observing a word is assumed to depend only on the history of the preceding N − 1 words, also admit a similar description. For instance the case of 1-grams corresponds to the product probability distribution

\[ p_{w_1, \ldots, w_n} = p_{w_1}^{[1]} \cdots p_{w_n}^{[n]}, \]

which can be represented by the TN diagram of Fig. 21. Such a 1-gram TN does not include any correlation between the words. For comparison, similar separable TNs are also the ones used in the so-called mean-field approximation to strongly correlated systems, where correlations between different sites are discarded [38], and which is known to fail whenever correlations are important. For the case of more complicated N-grams, one can actually define an appropriate language model quantum state, i.e.,

\[ |\Psi(N \text{- gram})\rangle = \frac{1}{Z^\frac{N}{2}} \sum_{\alpha \in N\text{-gram}} (p_{\alpha})^{\frac{1}{2}} |\alpha\rangle, \]

with \( \alpha \) an index running over all possible N-grams, \( p_{\alpha} \) their probabilities, \( |\alpha\rangle \) a set of orthonormal states, one for every N-gram (which is rather easy to construct), and \( Z \) the partition function of the distribution. Once such a state is available, one can do similar things as for the TN language models discussed previously, such as truly random sampling, and so forth.

VIII. IMPLICATIONS

Our “renormalization picture” of syntax and the results presented above demand for a necessary and detailed discussion about its implications, which extend into different ambiits. In what follows we elaborate on some of them, taking a somehow more phyllosophical perspective than in the previous sections, though well-grounded in our rigorous observations so far.

A. Good and bad models for language processing

The first practical implication, as we have already hinted in the previous sections, is that “good” language models (of any kind) should be compatible with the coarse-graining picture that we presented. From a generic perspective, one should expect a language model to reproduce the way humans seem to organize correlations in sentences, and from our perspective, this is given by the organization of coarse-grained information at different time scales. Concerning the field of artificial intelligence, we thus believe that a good starting point to obtain better language-processing algorithms, is to include also this organization of linguistic information according to time scales. This is in fact partially achieved already by the so-called “syntactic language models” [39]. The same applies to theoretical models of language in theoretical linguistics. Notice, importantly, that in this work we never hypothesized about what is the fundamental theory of grammar behind the known properties of MERGE. Questions such as “why a noun and an adjective merge into a noun phrase?” or “why is the output of

\[ \text{Ref.} [40]. \]

9 A recent attempt in this direction, also related to quantum physics and linear algebra, is the Matrix Syntax model in Ref. [40].
MERGE uniquely determined by its input?”, are beyond the scope of this work. In other words: we observed how correlations in human language get organized, explained this organization using the tools of physics, and exploited the consequences. And that is everything we did. We never discussed where these correlations could come from, or why they are as they are. In any case, and this is the point that we wish to make here, models attempting to explain this, either computational or theoretical, should encompass the picture presented here to be legitimate, since our observations are general.

We are now in position to answer the following question: why are some neural networks good at processing language, whereas some others are bad? The reason for this is now clear: those neural networks that are good, such as deep convolutional networks, are nothing but Tree Tensor Networks (TTNs) [45], and therefore codify the correct renormalization structure. As we explained before, TTNs can encompass the long-range correlations observed in language (i.e., the polynomial decay of correlation functions and of mutual information for two words in a sentence, see Sec.IV). Additionally, those neural networks that are bad at language processing, such as recurrent neural networks and hidden Markov models, also have a TN structure [45] but it corresponds to an MPS, i.e., a structure such as the one in Fig.12. As we explained also before, MPS have exponentially-decaying correlation functions and thus cannot account for the average long-range correlations of language. The message is then clear: some neural networks work well because they have the correct renormalization structure (e.g., deep convolutional networks), whereas other work badly because they do not have such a structure (e.g., recurrent neural networks and hidden Markov models). In fact, the connection between neural networks and renormalization had already been pointed out, see for instance Ref.[46]. Notice that, as a byproduct, our observation implies that one could have discovered the structure of biological neural networks not by looking at how the actual neurons are organized in our brains, but rather by analyzing the correlation structure of the language output of these brains. From such an analysis we could have already concluded, as the only logical option, that something like neurons must be somehow interconnected in a way compatible with the renormalization of information. As we know, this is in fact the case, and it is nothing but what tries to be mimicked by artificial neural networks. The point is, that one could have hinted this biological structure without having to open any brain!

B. Universal and non-universal properties

Given the renormalization structure and the properties of TN language models, one can predict universal features, i.e., properties that should be the same, no matter the language, and which only depend on the correlation structure of syntax\textsuperscript{10}. In this paper we found already some of such universal properties, as discussed in Sec.IV: the polynomial decay of correlation functions and mutual information is universal. However, the characteristic time-scale\textsuperscript{7} for this decay is language-dependent and should depend on external factors, such as cultural heritage.

We wish to remark that universal properties of language had already been observed by analyzing linguistic information with the tools of complex networks [41]. This is the field of physics and mathematics that analyzes complex systems and their structure from the network perspective (examples are ubiquitous: the internet, the power grid of a country, financial networks, the synaptic network in the brain...). In this setting, the so-called linguistic networks allow for a study of the properties of syntax from a pure network-theory perspective. In order to avoid confusion, we stress that our approach here is radically different, since we start from a very different physical perspective: renormalization, and how this orchestrates correlations. This led to a TN picture of language models which is different but complements the one obtained using complex-network theory.

C. Optimality of language

Several of the properties from the previous section seem to be related to the conjectured “perfection and economy” of human language in the Minimalist Program, as well as to the conjectured efficient processing of linguistic information in the brain [7, 21]. Let us take for concreteness the language models that we analyzed before. The fact that the TN structures are mostly loop-free automatically implies that the retrieval of information can be done efficiently in all computational resources (a problem in the complexity class P). Such computational efficiency strongly depends on the quasi-loop-free renormalization structure of syntax trees, and is therefore generically valid, i.e., not just for the case of language models. In fact, loop-free structures are well-known to be the cheapest non-trivial class of correlation structures in terms of the manipulation of their information [23]. The surprising fact is that human language is even more efficient than this, because of the properties of MERGE. In particular, we saw that the uniqueness of the output of MERGE once the input is specified, implied diagonal tensors and thus a correlated factorization in the TN, which leads to a dramatic efficiency in the calculation of probabilities for TN language models. It looks, therefore, that the human language chose the cheapest possible option able to keep non-trivial correlations between information

\textsuperscript{10} In words of N. Chomsky, “there is only one human language” (private communication).
units. Our brain could have evolved to use a MERGE where the output is non-unique for a given input and still maintain a big part of the computational efficiency in the manipulation of information, but this just didn’t happen. This observation makes precise the common-lore statement that language is, indeed, the cheapest non-trivial computational system. This may be one of the reasons why our brains choose to work with such correlation structures, instead of a different one. And we manage to externalize it through a physiological interface pretty well: we communicate (on average and most of the time) via sequential sounds in time produced with one mouth, instead of producing correlated sounds with, say, each one of our fingers, which would amount to 20 mutually correlated outputs, and thus a syntax full of correlation loops in turn implying computational inefficiency in the processing of its information.

D. Non-Markovian memory environment

A coarse-graining is a process that finds effective degrees of freedom to describe an emergent object, and inherently involves an information loss when moving from one scale to the next. It is well known in physics that renormalization is, usually, irreversible (the so-called “irreversibility of RG flows”) [42]. In language, however, it is clear that even if syntax manipulates coarse-grained objects at some long-time scale, we still know about the information content of the short-time scales. This is, our brain seems to organize the information according to different time scales, but does not seem to fully erase the information when going from one scale to the next, at least for some period of time. For instance, when we say a sequence of the type \([XYP] [A X] [Y Y]]\) (an adjective \(X\) followed by a noun \(Y\)), we remember for a while what it actually refers to: “happy cat”, “hot meal”, “interesting paper”, and so on. This seems to indicate that the “discarded” information seems not to be immediately erased, but just put apart for a while in some memory degree of freedom. To put it in physical jargon, one would say that the “memory environment” is non-Markovian, in the sense that there seems to be access for some period of time to the discarded information, shall this be needed. Understanding how and why this happens is indeed a relevant but different question to the one that we addressed in this paper.

E. Context-free grammars in other ambits

An interesting observation is that probabilistic context-free grammars (PCFG), though originally developed in linguistics, have proven recently very powerful in the probabilistic modelling of RNA and protein structures. In particular, PCFGs offer a way of determining the secondary structure of RNA, with a comparable accuracy to that obtained by energy minimization methods [43]. Concerning proteins, the situation is more complex but several achievements have already been reported using PCFG methods [44]. Many of the things that we mentioned previously in this work for the case of language, therefore, apply as well to the study of RNA and protein sequences. Even if being a very different scenario, the relevant correlation structures that appear in these biological problems happen to be similar to the ones that we described in this work, and therefore the same derivations could be applied to study those. The same is also true for the correlation structures present in programming languages, such as C++, Java, and so on. From a theoretical perspective, programming languages actually apply the rules of some grammar, i.e., rules by which words in a computer code are interpreted into meaningful machine instructions.

Intriguingly, one can also make a turnaround in the derivation that we presented here, and consider some TN structures as the natural correlation output of grammars. To be more precise, one could argue that TTNs and MPS can, in general, always be regarded as the output of some set of “generalized” context-free grammar rules where one allows for several possible outputs of a MERGE operation for a given input, being the outputs associated to complex “weights”. As such, this then implies that ground states of gapped 1d local quantum many-body Hamiltonians, which are known to have an MPS structure [18], are (roughly speaking) nothing but generalized grammatical structures. Whether this simple observation has consequences in the (analytical and numerical) study of quantum many-body systems remains as a provocative open question.

F. On typical human abilities

Intriguingly, similar structures to the ones presented here for the case of language and grammar have also been found in different but somehow related scenarios. For instance, as we said before, the correlation structure of neural network algorithms (which mimic in part the behavior of neurons in the brain) is, in fact, that of a Tree Tensor Network [45]. Renormalization-like algorithms are also common in the study of image compression, such as those based on wavelets [47], and even on Matrix Product States [48], where information of a picture gets organized according to different 2d length scales. Matrix Product States have also been used in the context of machine learning [49]. Moreover, it has been argued that the harmonic structure of tonal music may be, in fact, also a result of the MERGE syntactic operation [50]. As a matter of fact, it is believed that the faculty of language appeared in evolution almost simultaneously to the faculties of mathematics and music, with some people arguing in favour of the three faculties being actually three different manifestations of the same basic ability, which became available to our ancestors due to some genetic mutation throughout evolution [51]. A
very subtle, somehow missed point, but key in this regard, is that the mathematical faculty looks itself also as a coarse-graining of (mathematical) information. This is in fact a consequence of MERGE being the successor function in mathematics [52]. In order to make this point more explicit, let us directly cite a rather popular paragraph (at least in the linguistics’ community) in one of the recent works of N. Chomsky [53]:

“Suppose that a language has the simplest possible lexicon: just one lexical item, call it “one”. Application of MERGE to the lexical item yields {one}, call it “two”. Application of MERGE to {one} yields {one, {one}}, call it “three”. And so on. In effect, MERGE applied in this manner yields the successor function. It is straightforward to define addition in terms of MERGE(X,Y), and in familiar ways, the rest of arithmetic. The emergence of the arithmetical capacity has been puzzling ever since Alfred Russell Wallace, the co-founder of modern evolutionary theory, observed that the “gigantic development of the mathematical capacity is wholly unexplained by the theory of natural selection, and must be due to some altogether distinct cause”, if only because it remained unused. It may, then, have been a side product of some other evolved capacity (not Wallace’s conclusion), and it has often been speculated that it may be abstracted from the faculty of language by reducing the latter to its bare minimum. Reduction to a single-membered lexicon is a simple way to yield this consequence.”

Moreover, and at an experimental level, neuroscientists have recently discovered what could be the signature of the MERGE operation in neural activity, by analyzing the neural activation of epileptic patients performing several language tasks [54].

Given all this, we take the liberty to take off and hypothesize, somehow phylosophically and because everything seems to point in this direction, that the human abilities of language, mathematics, and probably others, may actually be different manifestations of a fundamental single ability of the human brain, namely, the ability to organize and process information according to different physical scales. To put it simple: one could say that the human brain is, among other things, a biological information-renormalization machine. When it comes to human language, this allows the brain to build a language system of discrete infinity, i.e., a discrete and recursive system able to produce infinitely-many outputs.

IX. CONCLUSIONS AND PERSPECTIVES

The observations and results in this paper are highly interdisciplinary. Let us briefly summarize here the main points. We have argued that the linguistic MERGE operation entails renormalization in physics: the information content in, e.g., sequences of words (short time scale) gets renormalized by MERGEs up to sentences (long time scale). We have made this observation exact and formal for language models, and have found that probabilities of meaningful sentences are naturally given by quasi-loop-free TNs, which in turn organize correlations according to different renormalization time scales. Such language models are naturally related to probabilistic context-free grammars, though not restricted only to them. We have discussed some of the properties of these TN language models: locally-built syntactic correlations at every scale, very high efficiency of information processing because of correlated factorization of the TN, syntactic correlations, and practical refinement levels. We also proved that long-range correlations in language follow naturally from our approach. Moreover, we proposed how to promote probabilistic language models to probability distributions of quantum states, argued that such quantum states may be useful when it comes to sampling the distribution, showed how they can be built efficiently in a quantum computer, and used their entanglement properties to provide a classical lower bound on the statistical perplexity of finding a set of words in a sentence. We discussed also how this useful formalism may be generalized to other types of grammars, and discussed a number of implications of our observations in several ambits. These concern the legitimacy of language models and neural networks for language processing, universality and optimality of language, some required properties of the memory environment, the potential application of our formalism for RNA and protein sequencing as well as programming languages and quantum many-body systems via context-free grammars, and the overall picture of several human faculties all somehow boiling down to MERGE. In the end, we have taken the liberty to hypothesize that the human brain seems to have a natural fundamental ability to organize information according to different physical scales, from which other faculties may materialize.

Our work opens the possibility to use all the mathematical and physical knowledge about TN states, both classical and quantum, in the theoretical and computational study of language and grammar. This includes a wide variety of applications not just in linguistics, but also in RNA and protein sequencing [43, 44] and the design of computer languages, just to name some well-known examples. In particular, the different ways to quantify correlations and the information content in the network, as well as associated numerical algorithms [18], should find useful applications in these scenarios. Moreover, the efficient descriptions of probability distributions of relevant grammars by means of quantum states, opens the exciting possibility to use possible quantum comput-
ers and quantum simulators to deal with problems in all these ambit. A prominent example is AI, where our results show that quantum information tools can be used to validate, simulate, assess, and improve state-of-the-art language models, as well as that quantum computers can be used to implement perfect random sampling of language, which is simply impossible with classical technology. This is particularly relevant given the recent big advances in the development of experimental quantum processors. Additionally, it would also be interesting to study the connection between the observed structure of probability distributions for language, and those obtained from the renormalization method for reaction-diffusion models in physics \cite{55}, which seem to share some similarities.

By digging deeper into linguistic concepts it is indeed possible to take our analogies further. We do this in Appendix A. All in all, our conjecture that MERGE in linguistics is connected to RG in physics turns our to be extremely fruitful, since many of the key linguistic ideas from the last century fit perfectly with known physical concepts linked to renormalization. We have also seen that, as a consequence, many concepts in computational linguistics also match perfectly with well-known physical conceptions. The main analogies discussed in this paper, including those in the appendix, are summarized in Table I.

| Linguistics | Physics |
|-------------|---------|
| MERGE       | Coarse-graining |
| Relabelling | Rescaling |
| Derivation  | RG flow |
| Phase       | RG scale |
| Phase impenetrability | RG irreversibility |
| Optimality and efficiency | Loop-free structures |

| Prob. language model | 1d tensor network |
|----------------------|------------------|
| N-gram models        | Mean-field theory |
| Prob. context-free grammar | 3-index tensor & MPS/TTN |
| Dependency grammar   | (k > 3)-index tensor & 1d MERA |
| \sqrt{Prob. language model} | Quantum circuit |
| Perplexity            | Quantum entanglement |

TABLE I: Main analogies and connections between linguistics and physics proposed in this paper. The upper part corresponds to concepts usually discussed in theoretical linguistics, and the lower part to concepts in computational linguistics. 1d means that the “physical” degrees of freedom span along one dimension, which in the case of language is time. The “square-root” symbol in the lower-left panel is a way of saying that the corresponding quantum circuit produces probability amplitudes that are the square root of the actual probabilities given by the language model.

Only good things can happen by studying language from the perspective of physics. Our work here is one more addition to the long-standing program of using physics tools to understand the linguistic capacity. Other works have focused on descriptions based on quantum field theory, complex network theory, and quantum mechanics \cite{56}, and the results in this paper add a fresh new perspective to this overall framework. The fields of physics and linguistics have been traditionally very far away from each other. But indeed, linguistics focuses on the study of the laws of language, and physics on the study of the laws of Nature. For a linguist, the human language is the universe, and it has deep connections with how our brain processes and manipulates information, as well as other situations whose correlations are orchestrated by grammar-like rules. From the perspective of physics, it feels just natural to think that classical and quantum information theories should be somehow useful for this purpose. Being able to formalize mathematically some of the most relevant aspects of language and grammar in terms of physical ideas is already an important achievement. We strongly believe that the cross-fertilization of physics and linguistics will become increasingly relevant in the future. Phyllosophical questions, such as those encountered sometimes in linguistics, usually lead to deep, profound scientific problems, and our work here is no exception to this rule.

Acknowledgments

We acknowledge the Universitat Autònoma de Barcelona, the Max Planck Institute of Quantum Optics, and the University of Mainz, where the ideas in this paper materialized from interdisciplinary discussions over several years. Discussions with Oniz Aizpuru, Gustavo Ariel Schwartz, Gemma de las Cuevas, Laura García-Álvarez, Geza Giezdke, José I. Latorre, Enrique Solano, Miles Stoudenmire and Juan Uriagereka are acknowledged. A special acknowledgement to Sergi Quintana, for explaining us the basics of Chomsky’s generative grammar 28 years ago: qui sembra, recall.

Appendix A: More analogies by digging deeper

Our paper is written having in mind a reader with background on physics and mathematics. However, the topic itself is strongly interdisciplinary. Because of this, in this appendix we would like to add some extra information useful for the reader with knowledge about theoretical linguistics. In particular, we would like to define a few concepts more precisely in linguistic jargon. Thanks to this, we will see that by digging deeper into the linguistic jargon, more analogies with physics will show up, in turn strengthening our thesis that MERGE in linguistics and RG in physics are deeply linked to each other.

To begin with, the term Universal Grammar (UG) is nothing but a label for the striking difference in cognitive capacity between “us and them”, i.e., humans versus the
rest of animal species. UG is thus the research topic of generative grammar in its attempt to understand what it is and how it evolved in our species. Finding a satisfying answer to the latter question may be impossible with the tools we have right now, but any theory of UG seeking to address the former must meet a criterion of evolvability: any properties, mechanisms, etc. attributed to UG should have emerged in what appears to have been a unique and relatively sudden event on the evolutionary timescale [57]. This line of thought presupposes that UG (the genetic encoding of the the human linguistic capacity) manifests bona fide traits of perfect design, in the sense that contains operations and mechanisms that follow from conceptual necessities, efficiency principles or interface demands. In this respect, linguistic expressions (sentences, phrases, words, etc.) are built up by adhering to these principles, therefore in an optimal fashion. While these notions are intuitively clear, their precise formulations remain vague and controversial.

One of the most important mathematical achievements of generative grammar is the so-called “Chomsky Hierarchy” [58], a classification of formal grammars according to their complexity. As Chomsky showed sixty years ago, human languages manifests both context-free and context-sensitive properties, needed to construct PHRASES and CHAINS respectively, shown in the Sentences A1(a,b):

\[
\begin{align*}
a. \ & \text{John killed John.} \\
b. \ & \text{John was killed} < \text{John }> \\
\end{align*}
\]

In Sen.A1(a) (a PHRASE) we have two tokens of the lexical item “John” that participate in phrasal dependencies to yield a compositional interpretation whereby the first John is the agent of a killing event, and the second John is the patient of such event. What we have in Sen.A1(b) (a CHAIN) is more complex. This time, we don’t have two tokens of “John”, but two occurrences of the same lexical item – as if they were one and the same object in two positions at the same time, where the notation < John > means that the word itself is not pronounced at that position, but it is also interpreted there from the logical point of view. This is what is called CHAIN in linguistics. In languages of the English type, the first (leftmost) occurrence is spelled-out, whereas the second (rightmost) is necessary to keep a syntax-semantics homomorphism (that is, to capture the desideratum that a specific interpretation is tied to a specific position). Notice that the same type of object (a CHAIN) is necessary in Sen.A2, where “John” is pronounced to the left of seem, although it is interpreted as the patient of killed.

John seems to have been killed < John > \hspace{1cm} (A2)

In order to account for these properties, generative grammar has resorted to phrase structure rules (PSR) and transformations. The most articulated version of PSR is known as X-bar Theory, which resorted to different devices that have been subject to a revision within minimalism. In particular, Chomsky [10] argued that the basic properties of PSR could be understood by means of a computational operation, dubbed MERGE [10], which captures two empirical properties of human language that are non-negotiable: discrete infinity and displacement. To be able to account for those properties, one must assume an operation that constructs hierarchically structured expressions with displacement. And that is what MERGE does. MERGE applies to two objects X and Y (be these words or bigger units), yielding a new one, K, which is the set containing X and Y, i.e., \{X, Y\}. If X, Y are distinct (taken directly from the lexicon or independently assembled), K is constructed by what is called EXTERNAL MERGE (EM); if Y is part of X (if Y is contained in X), then we have what is called INTERNAL MERGE (IM). The latter scenario is that of Sentences A1(b) and A2 above, where MERGE turns “John” into a discontinuous object (a CHAIN). For completeness, if the operation is at the beginning of a derivation (e.g., with bare lexical items from a lexicon), it is called FIRST MERGE, and if it operates with partially-derived items (phrases), it is called ELSEWHERE MERGE.

Chomsky [10] takes MERGE to be strictly binary, as it is what is minimally necessary to create hierarchical structure. Generation by MERGE thus entails a restrictive class of recursively defined, binary-branching and discrete-hierarchical structures.

It is also worth mentioning that in X-bar Theory, the label identifies the properties of the entire phrase, at the cost of this being a theory-internal symbol that departs from inclusiveness demands. An alternative to this is a label-free representation (see Fig.1), where endocentricity (the assumption that all phrases must be headed) is not preserved. This entails that syntactic objects can be exocentric, as seems to be necessary for objects formed by the combination of two phrases, \{XP, YP\}. Syntactic objects are “endocentric” if they contain an element that can be determined by Minimal Search – typically, a head. Given this logic, \{XP, YP\} is endocentric and \{XP, YP\} exocentric. Consequently, such a system freely generates objects of different kinds, without stipulating their endocentric nature.

Moreover, MERGE is subject to efficiency and economy conditions. One such condition is inclusiveness, which precludes the introduction of extraneous objects, like the ones that X-bar Theory deployed: traces, bar-levels, projections, etc. Inclusiveness also bars introduction of features that are not present in lexical items.

To further clarify MERGE, we stress that the combination of two objects, X and Y, yields a new one, K, which is the set \{X, Y\}. Once we have \{X, Y\}, we may want to merge K and some object W, which can be either internal to K or external to it (see above). In any event, the merger of W cannot change or tamper with \{X, Y\}, which behaves as a unit. More precisely, subsequent applications of MERGE must yield Eq.A3(a), not
A3(b):

\[ a. \text{MERGE}(K, W) = \{(X, Y), W\} \quad \text{(A3)} \\
\]

\[ b. \text{MERGE}(K, W) = \{(X, W), Y\} \]

The driving force of this work is the fact that MERGE and renormalization seem to play a similar role on various respects. As noted above, MERGE takes two objects, \(X\) and \(Y\), to yield a new one, \(K\), thus removing \(X\) and \(Y\) from the computational workspace (WS). In the simplest scenario, MERGE maps WS = \(\{X, Y\}\) onto WS' = \(\{X, Y\}\), reducing the complexity of WS. Notice that MERGE never extends the WS, at least in terms of cardinality; thus WS = \(\{X, Y\}\) and WS' = \(\{W, \{X, Y\}\}\) are equally bigger, since they only contain one set. A new element can be added to WS (or WS') in only one way: by taking two items \(W, Z\) from the lexicon and introducing \(\{W, Z\}\) into WS as a new element, yielding WS'' = \(\{W, Z, \{X, Y\}\}\). Of course, cardinality can be reduced if we apply EM (EXTERNAL MERGE) and neither of the elements are taken from the lexicon, as if we map WS'' = \(\{W, Z, \{X, Y\}\}\) onto WS''' = \(\{(W, Z), \{X, Y\}\}\). This idea is indeed very similar to that of a coarse-graining in physics, in the sense made precise throughout the paper.

Additionally, the possibility that computational load is reduced by MERGE is perhaps somewhat new, as this typically follows from a principle in linguistics that is called STRICT CYCLICITY. The notion of cycle (and thus cyclicity) goes back to the fifties, where work in phonology [60] showed that the application of stress-assigning rules apply from innermost to outermost units of a word, putting aside linear order information. More generally, an object is build under cyclic principles if it is COMPOSITIONAL, which means that its interpretation is fixed by the elements it contains and the way in which they are combined. Consider this with Sentences A4, where the interpretation is crucially different (Brutus is an agent in (a), and a patient in (b)), although both examples contain the same three words:

\[ a. \text{Brutus stabbed Caesar.} \quad \text{(A4)} \\
\]

\[ b. \text{Caesar stabbed Brutus.} \]

The concept of STRICT CYCLICITY is a stronger version of cyclicity. The key intuition behind it is that for certain linguistic object constructed in a derivation (say, a \(VP\)), further computation should not modify it. Let us see this with the example in Eq.A5, where the verb “leave” is merged with the \(NP\) “the room” to yield the complex \(VP\) “leave the room”, which we can call \(K\) for ease of reference.

\[ \text{MERGE(leave, \{the, room\}) = \{leave, \{the, room\}\}} \quad \text{(A5)} \]

What is of interest here is that the interpretation of \(K\) (that is, of “leave the room”) is determined at that stage of the derivation (at that “cycle”), and cannot be changed at subsequent stages (“cycles”). Therefore, if we add “Mary” to obtain “Mary leaves the room” (call it \(K'\)), as in Eq.A6, the interpretation of \(K\) will be the same in Eq.A5 and in Eq.A6.

\[ \text{MERGE(Mary, } K) = \{\text{Mary, \{leaves, \{the, room\}\}}\} \quad \text{(A6)} \]

In a nutshell, the interpretation of complex objects is constructed stepwise, in a step-by-step fashion, and whatever has been done at a stage \(s\) cannot be undone at stage \(s + 1\) (Eqs.A5 and A6 above). This, in turn, is quite analogue to the idea of irreversibility of RG flows in physics, which matches perfectly with our interpretation of MERGE as a coarse-graining of information.

Such stages at a derivation, where a “computation” is done and cannot be altered afterwards, correspond with the so-called linguistic PHASES, and the device responsible for ensuring that the interior of a PHASE is no longer accessible is the PHASE IMPENETRABILITY CONDITION (PIC for short). What has been called “phase” roughly corresponds with the notion of “cycle” described above. Using the physical interpretation that we introduce in this paper, one would say that a PHASE in linguistics is the analogous of an RG scale in physics.

To be more precise, a phase is defined in linguistics as a domain \(D\) where uninterpretable features (number and person features of verbs) are valued. When a phase is closed off, the complement domain \(\Omega\) (which can itself be complex, in the sense of having some inner structure) of the phase head \(P\) cannot be modified from the outside; this means, for instance, that the case of an \(NP\) within \(\Omega\) (e.g., “the book” in the \(VP\) “read the book”) cannot be changed once the phase headed by \(P\) is complete [61]. Among other things, this entails that “the book”, which is the Direct Object of “read” Sentence A7 (it receives accusative case from “read”), cannot also be the Direct Object of the matrix verb “believe”:

\[ \text{I believe that John read the book.} \quad \text{(A7)} \]

That “the book” is the Direct Object of “read” and not of “believe” is shown in Sentences A8, where we see that this \(NP\) can be passiviced in the embedded clause, but not in the matrix clause (* signals ungrammaticality):

\[ a. \text{I believe that the book was read.} \quad \text{(A8)} \\
\]

\[ b. \star \text{The book was believed that John read.} \]

This “shielding” effect that makes the \(VP\) impenetrable is captured by the phase impenetrability condition mentioned above. Physically, this is the irreversibility of the RG flow when moving from one RG scale to the next. There are various approaches to Phase Theory [59], but all of them share the key intuition that PHASES are domains where complexity is reduced by somehow allowing the system to “forget” about an amount of structure that has been created and which will be no longer accessible. This process of “forgetting” is, in fact, analogous to the process of “discarding irrelevant degrees of freedom” in an RG-step in physics.
Moreover, the “rescaling” step in RG has not been discussed in this paper, but also appears naturally when particularizing to specific models of language. For instance, in the Matrix Syntax model \[40\] this rescaling appears naturally in order to recover the correct linguistic labels after a MERGE operation (see Ref.\[40\] and the discussions therein for more information). We believe that this is a general feature: the “rescaling” in physics is nothing but the “mathematical relabelling” that one needs in order to recover the correct labels (NP, VP, etc) after a MERGE operation when dealing in practice with models of language.

**Funding:** This project was not funded by any grant.

**Conflict of interest:** The authors declare that they have no conflict of interest.
der, J. Zittartz, Equivalence and solution of anisotropic spin-1 models and generalized t-J fermion models in one dimension, J. Phys. A 24, L955 (1991); A. Klümper, A. Schadschneider, J. Zittartz, Matrix-product-groundstates for one-dimensional spin-1 quantum antiferromagnets, Europhys. Lett. 24, 293 (1993); U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. 326, 96 (2011).

[25] I. V. Oseledets, Tensor-Train Decomposition, SIAM J. Sci. Comput. 33(5), 2295-2317 (2011).

[26] N. Chomsky, Syntactic structures. The Hague/Paris: Mouton (1957).

[27] See, e.g., H. Liu, Dependency Grammar: from Theory to Practice. Beijing: Science Press (2009).

[28] See https://en.wikipedia.org/wiki/N-gram cite_note-1

[29] See https://en.wikipedia.org/wiki/QR ♯

[30] C. H. Papadimitrou, Computational Complexity, Addison Wesley, 1994.

[31] E. Alvarez-Lacalle, B. Dorow, J.-P. Eckmann, E. Moses, C. Holzhey, F. Larsen, and F. Wilczek, Geometric and

[32] N. Schuch

[33] I. V. Oseledets, Tensor-Train Decomposition, SIAM J. Sci. Comput. 33(5), 2295-2317 (2011).

[34] N. Chomsky, Syntactic structures. The Hague/Paris: Mouton (1957).

[35] See, e.g., H. Liu, Dependency Grammar: from Theory to Practice. Beijing: Science Press (2009).

[36] See https://en.wikipedia.org/wiki/N-gram cite_note-1

[37] See https://en.wikipedia.org/wiki/QR ♯

[38] L. P. Kadanoff, Dimension

[39] Spin-1 models and generalized t-J fermion models in one

[40] K. Temme, F. Verstraete, et al

[41] N. Chomsky, E. Alvarez-Lacalle, B. Dorow, J.-P. Eckmann, E. Moses, C. Holzhey, F. Larsen, and F. Wilczek, Geometric and

[42] N. Schuch
Piattelli-Palmarini, G. Vitiello, *Linguistics and some aspects of its underlying dynamics*, arXiv:1506.08663; M. Piattelli-Palmarini, G. Vitiello, *Quantum field theory and the linguistic Minimalist Program: A remarkable isomorphism*, Journal of Physics: Conf. Series **880** 012016 (2017); R. Solé, *Synthetic transitions: towards a new synthesis*, Phil. Trans. R. Soc. B **371**: 20150438 (2016).

[57] J. Bolhuis, I. Tattersall, N. Chomsky, and R.C. Berwick, *How Could Language Have Evolved?* PLoS Biology **12**: e1001934 (2014); R. C. Berwick and N. Chomsky *Why Only Us*, Cambridge, MA: MIT Press (2016).

[58] N. Chomsky, *Three models for the description of language*, -IRE Transactions on Information Theory 2: 113-124 (1956).

[59] A. J. Gallego (ed.), *Phases. Developing the Framework*. Berlin: De Gruyter (2012).

[60] N. Chomsky, M. Halle, F. Lukoff, *On Accent and Juncture in English*. In *For Roman Jakobson: Essays on the occasion of his sixtieth birthday*, M. Halle et al. (eds.), 65-80. The Hague: Mouton and Co. (1956); N. Chomsky, M. Halle, *The Sound Pattern of English*. New York: Harper Row (1968).

[61] N. Chomsky, *Problems of Projection*. Lingua 130: 33-49 (2013); N. Chomsky, *Problems of Projection. Extensions*. In E. di Domenico et al. (eds.), *In Structures, Strategies and Beyond*, 1-16. Amsterdam: John Benjamins. (2015).