# Processing and Representation of Linguistic Properties in Large Language Models

Elisabetta Rocchetti[1,*,†], Alfio Ferrara[1,†]

[1]Università degli Studi di Milano, Department of Computer Science

## Abstract

This paper reviews studies evaluating the linguistic performance of large language models (LLMs), focusing on how they process and represent linguistic properties. We explore methods like probing classifiers and Iterative Nullspace Projection (INLP) to assess whether LLMs actively use encoded knowledge during inference, and how shifting representations can help evaluate model performance. We report findings showing that LLMs can encode formal properties, such as syntax and morphology, but are less proficient with functional phenomena like semantics, with monolingual models outperforming multilingual ones. We also highlight gaps in current evaluations, such as the limited testing of recent large models on a small set of tasks [1]. We suggest that future research could integrate different perspectives of what linguistic competence is.

## Keywords

Explainable AI, Large Language Models, Probing, Linguistic Properties

## 1. Introduction

State-of-the-art Large Language Models (LLMs) have demonstrated an extraordinary ability to generate human-like text. Their fluency suggests that they do more than just predict the next word based on surface-level patterns observed during training. Instead, these models appear to capture deeper, more abstract linguistic properties, enabling them to construct grammatically correct and contextually appropriate sentences. However, the precise nature of these learned representations remains an open question. Researchers in the field of Explainable AI (XAI) have been actively investigating which linguistic properties LLMs encode, how they use them, and whether these properties are merely incidental patterns or core components of their reasoning process.

One fundamental observation is that LLMs go beyond simple word association. If a model merely memorized sequences of words, it would struggle to generalize grammatical rules to new contexts. However, research suggests that LLMs implicitly learn various linguistic structures that guide their predictions. For example, models can often correctly conjugate verbs according to the subject of the sentence, a phenomenon known as subject-verb agreement. Consider the sentence:

(1) The cat that chased the dog is sleeping.

Even though the noun “dog” is closer to the verb “is sleeping”, the model correctly associates “is sleeping” with “cat”, demonstrating an understanding of syntactic dependencies rather than relying solely on proximity.

XAI research has explored which linguistic properties LLMs encode and how this information is structured within their internal components. In particular, researchers have investigated whether specific neurons or layers store linguistic knowledge and how this knowledge contributes to the model’s behavior. Identifying where particular information is stored is valuable for various applications, such as controlling the model’s output [2] or reducing model size by retaining only its most important components [3, 4]. By analyzing internal representations—either through embeddings or neuron

SEBD 2025: 33rd Symposium on Advanced Database Systems, June 16-19, 2025, Ischia, Italy

[*]Corresponding author.

[†]These authors contributed equally.

elisabetta.rocchetti@unimi.it (E. Rocchetti); alfio.ferrara@unimi.it (A. Ferrara)

0009-0000-5617-7612 (E. Rocchetti); 0000-0002-4991-4984 (A. Ferrara)

activations—researchers have found evidence that LLMs capture linguistic phenomena in structured ways.

In this work, we present a comprehensive literature review of state-of-the-art methods for investigating whether large language models (LLMs) encode knowledge of various linguistic properties. We begin by formalizing the types of language models under consideration, as well as the linguistic phenomena that can be analyzed. We then examine existing research on how LLMs represent linguistic information within their internal layers, with a focus on studies that employ probing classifiers to uncover specific linguistic features. Additionally, we explore approaches for evaluating whether the linguistic information present in a model is actually utilized during inference. Finally, we synthesize empirical findings from the literature, highlighting key results and validating the discussed methodologies.

The paper is structured as follows: Section 2 introduces the notation that will be used to explain key concepts; Section 3 presents examples of the linguistic properties that can be analyzed in LLMs; Section 4 describes probing classifiers as a methodology for interpreting and analyzing the internal representations of LLMs; Section 5 discusses methods for evaluating whether a LLM encodes specific linguistic phenomena in its hidden representations. It also introduces Iterative Nullspace Projection (INLP) as a method for selectively identifying and removing specific features from hidden representations; Section 6 summarizes key insights regarding the linguistic competence of language models. It also explores how language models' ability to learn and encode linguistic properties varies when considering different languages, particularly in the context of multilingual versus monolingual models; Section 7 concludes the paper.

## 2. Language models

In this section, we introduce the notation that will be used to refer to LLMs throughout the rest of the paper.

Let $\mathscr{M}$ be a LLM with $L$ hidden layers, each containing $N$ neurons. Each neuron in layer $l \in \{1, \ldots, L\}$ is connected to neurons in layer $l + 1$ through weighted connections with biases. $\mathscr{M}$ refers to a LLM with any architecture, including LSTM, RNN, or Transformer. Let $\mathbf{W}^{(l)} \in \mathbb{R}^{N \times N}$ be the weight matrix storing the connection weights between neurons in layer $l_i$ and those in layer $l_{i+1}$. Similarly, let $\mathbf{b}^{(l)} \in \mathbb{R}^{N}$ be the bias vector associated with layer $l$. The model $\mathscr{M}$ processes an input sequence represented as a matrix $\mathbf{X} \in \mathbb{R}^{S \times V}$, where $S$ is the sequence length and $V$ is the vocabulary size. Each row of $\mathbf{X}$ corresponds to a one-hot encoded representation of a token from the input string.

As the input propagates through the model, it is transformed into a series of latent representations. Specifically, at each layer $l \in \{1, \ldots, L\}$, the model computes a hidden representation $\mathbf{H}^{(l)} \in \mathbb{R}^{S \times N}$, where each row corresponds to a transformed token embedding in a latent space of dimension $N$. We denote by $h_{ij}^{(l)}$ the activation of the $j$-th neuron at the $l$-th layer for the $i$-th token in the sequence.

Finally, the model generates an output matrix $\mathbf{Y} \in \mathbb{R}^{S \times V}$, where each row $\mathbf{y}_s$ represents a probability distribution over the vocabulary, indicating the likelihood of each token being the next in the sequence at position $s$. The final row, $\mathbf{y}_S$, corresponds to the probability distribution for the next token following the last input token. This output distribution is obtained by applying a final linear transformation to the last hidden representation, followed by a softmax function.

During decoding, the next token is typically selected by choosing the most probable token from $\mathbf{y}_S$, or by sampling from the distribution, depending on the inference strategy used.

Table 1 summarizes the notation discussed in this section.

## 3. Linguistic properties

In this section, we present examples of the linguistic properties that can be analyzed in LLMs. Prior work, such as [1, 5], categorizes these properties into two broad types: *formal* and *functional* linguistic phenomena.

**Table 1**
Summary of notation used in the paper.

| Notation | Description |
|---|---|
| $\mathscr{M}$ | LLM (e.g. LSTM, Transformer) |
| $L$ | Number of hidden layers in $\mathscr{M}$ |
| $N$ | Number of neurons per layer |
| $S$ | Sequence length (number of tokens in input) |
| $V$ | Vocabulary size |
| $\mathbf{X} \in \mathbb{R}^{S \times V}$ | Input sequence as a matrix (one-hot encoded tokens) |
| $\mathbf{W}^{(l)} \in \mathbb{R}^{N \times N}$ | Weight matrix connecting layer $l$ to layer $l+1$ |
| $\mathbf{b}^{(l)} \in \mathbb{R}^{N}$ | Bias vector for layer $l$ |
| $\mathbf{H}^{(l)} \in \mathbb{R}^{S \times N}$ | Hidden representation at layer $l$ |
| $h_{ij}^{(l)}$ | Activation of neuron $j$ at layer $l$ for token $i$ |
| $\mathbf{Y} \in \mathbb{R}^{S \times V}$ | Output matrix (probability distributions over vocabulary) |
| $\mathbf{y}_s$ | Probability distribution over the vocabulary for position $s$ |
| $\mathbf{y}_S$ | Probability distribution for the next token after the last input token |

**Table 2**
Table from [1]. Examples of different linguistic phenomena and their corresponding labels. The relevant part of the example for the specific label is underlined.

| Type | Phenomena | Example | Label |
|---|---|---|---|
| Morphology | Subject-Verb Agreement | And then, the cucumber was hurled into the air. | Correct |
| | | And then, the cucumber were hurled into the air. | Wrong |
| Syntax | Part-of-Speech | And then, the cucumber was hurled into the air. | NN (Noun Singular) |
| Semantic | Semantic Roles | And then, the cucumber was hurled into the air. | Direction |
| Reasoning | Negation | And then, the cucumber was hurled into the air. | No Negation |
| Discourse | Node Type in Rhetorical Tree | And then, the cucumber was hurled into the air. | Satellite |

**Formal linguistic phenomena** refer to the structural aspects of language. *Morphology* deals with the internal structure of words, including rules governing word formation, inflection (e.g., pluralization, verb conjugation), and derivation. *Syntax* studies the arrangement of words and phrases to form grammatically correct sentences, including subject-verb agreement, word order, and syntactic dependencies.

**Functional linguistic phenomena** relate to how language conveys meaning and is used in communication. *Semantics* analyzes the meaning of words and sentences, including word sense disambiguation and compositional meaning. *Reasoning* explores the ability to perform logical inferences, including handling negation, speculation, and deductive reasoning. *Discourse* focuses on the structure and coherence of text beyond the sentence level, such as understanding rhetorical relations, reference resolution, and text cohesion.

Formally, we define the set of all possible linguistic properties as $\mathscr{T} = \{t_1, \dots, t_T\}$, where each property $t_i \in \mathscr{T}$ is associated with a set of possible values or labels $\mathbf{y}^{(t_i)}$. For instance, consider the linguistic property of subject-verb agreement: this phenomena can take two possible values: $\{\texttt{correct}, \texttt{incorrect}\}$, indicating whether the verb is correctly conjugated to match its corresponding subject. Additional examples of linguistic properties and their respective labels can be found in Table 2.

## 4. Probing classifiers

Probing classifiers represent a powerful methodology for detecting linguistic knowledge in internal representations of LLMs. The core idea behind probing is to train an external classifier to predict a specific linguistic property based on the representations extracted from the model [6, 7, 8, 9, 10].

Formally, a probing classifier can be defined as a function $p : \mathbf{H}^{(l)} \rightarrow \mathbf{y}^{(t_i)}$, which maps a hidden representation $\mathbf{H}^{(l)}$ from layer $l$ to a label $\mathbf{y}^{(t_i)}$ associated with a linguistic property $t_i$. The classifier's weights $\boldsymbol{\theta}$ are trained on an annotated dataset, where the inputs consist of hidden representations extracted from the LLM, and the outputs are labels corresponding to a specific linguistic phenomena. The performance of $p$ provides insight into how well the internal representations encode the target property. If a probe $p$ successfully predicts the property (achieving high accuracy), it suggests that the linguistic feature in question is encoded within the model's hidden states. Conversely, if the probe fails, this may indicate that the information is either absent or not accessible in the given representation.

To illustrate, consider the task of subject-verb agreement—a fundamental grammatical rule in many languages. A probing classifier can be trained to predict whether a verb correctly agrees in number with its subject based on the hidden states of an LLM. Suppose we extract the representation $\mathbf{H}^{(l)}$ from a given layer of the model while processing sentence 1: "*The cat that chased the dogs is sleeping.*" The correct verb form *is* must agree with the singular subject *cat*, despite the presence of the plural noun *dogs* intervening between them. A well-trained probe should be able to infer that the subject is singular based on $\mathbf{H}^{(l)}$ and correctly predict this agreement pattern. If the probe achieves high accuracy across various test cases, this suggests that the LLM's internal representations contain information about subject-verb agreement. However, if the probe struggles to distinguish correct from incorrect verb forms, it may imply that this grammatical rule is not explicitly encoded in the model's representations or is only accessible in a more complex, non-linear manner.

Probing classifiers have been used extensively to investigate a variety of linguistic properties, including syntactic structures and part-of-speech information. For instance, previous studies have shown that linear transformations of LLM representations can encode syntax tree structures [9, 7] and part-of-speech tags [8]. Specifically, in cases like those mentioned, where the probing classifier is trained as a linear classifier, it can be inferred that the information it learns is also represented linearly in the model. By systematically applying probes to different layers and architectures, researchers can gain deeper insights into how linguistic information is distributed within LLMs and how different levels of abstraction are captured across layers.

In the following sections (4.1 and 4.2), we illustrate how LLMs' internal representations can be used to train a probing classifier for detecting and localizing linguistic properties.

### 4.1. Probing linguistic properties via embeddings

One way to analyze the linguistic knowledge stored in an LLM is by evaluating its internal representations, specifically the embeddings produced at different layers. This approach considers all neurons in a layer collectively, working with the hidden representation $\mathbf{h}_i^{(l)}$ rather than isolating the contribution of individual neurons. This analysis does not attempt to pinpoint where linguistic information is stored but instead assesses whether the model encodes specific linguistic properties at all.

A common method involves extracting the latent representations of tokens from the last layer of the network, $\mathbf{H}^{(L)}$, and using them as input to probing classifiers. These classifiers are trained to predict linguistic features from the embeddings, providing insight into the extent to which the model $\mathcal{M}$ captures syntactic or semantic information [6, 7, 8, 9, 10].

### 4.2. Neuron ranking

To go beyond embedding-level analysis, some studies focus on ranking individual neurons based on their relevance to a specific linguistic property. Instead of treating the entire hidden representation as a single feature vector, this method evaluates the contribution of individual neurons by analyzing their activations and how they influence external classifiers.

This approach involves training a probing classifier to predict a linguistic feature while using neuron activations as input features. By examining how much each neuron contributes to the classifier's prediction, researchers can rank neurons according to their importance. If this ranking is meaningful, we should be able to retain only the top-$k$ neurons and still achieve high accuracy with the probe. If the probe maintains strong performance using only a small subset of neurons, this indicates that the ranking effectively identifies the most informative neurons [11, 12, 13].

Focusing on linear probes, Dalvi et al. [11] propose training a linear classifier to classify the values $\mathbf{y}^{(t_i)}$ of a linguistic property $t_i$ (e.g., subject-verb agreement, part-of-speech tagging, morphological analysis, or semantic tagging). The importance of a neuron for classifying $t_i$ can be inferred from the magnitude of its associated classifier weight. Specifically, to rank neurons by their relevance, the authors extract the weight vector $\boldsymbol{\theta} \in \mathbb{R}^N$ from the trained probe and sort its elements by absolute value in descending order.

By ranking neurons, [11, 12, 13] have demonstrated that linguistic knowledge is distributed across neurons, though not uniformly. Instead, it is often concentrated in a subset of highly ranked neurons, rather than being evenly spread across all neurons in the model. This finding supports the idea that some neurons specialize in encoding specific linguistic properties, making it possible to extract and interpret this information with targeted analysis.

## 5. Assessing and controlling the usage of linguistic features in LLMs

So far, we have discussed methods for evaluating whether a LLM encodes specific linguistic phenomena in its hidden representations. However, probing classifiers and similar analysis techniques primarily reveal whether linguistic properties are present in the model's internal representations—they do not indicate whether and how these properties are actually used during inference [14].

To determine whether an LLM actively uses a certain linguistic property during inference and to modify the model's behavior accordingly, *Iterative Nullspace Projection* (INLP) [15] provides a method for selectively identifying and removing specific features from hidden representations. In the original paper, the authors successfully identified gender information encoded in the model's contextual representations. By removing this information, they observed that the model's predictions were no longer biased, indicating that the original model had indeed relied on gender-related features in its hidden representations.

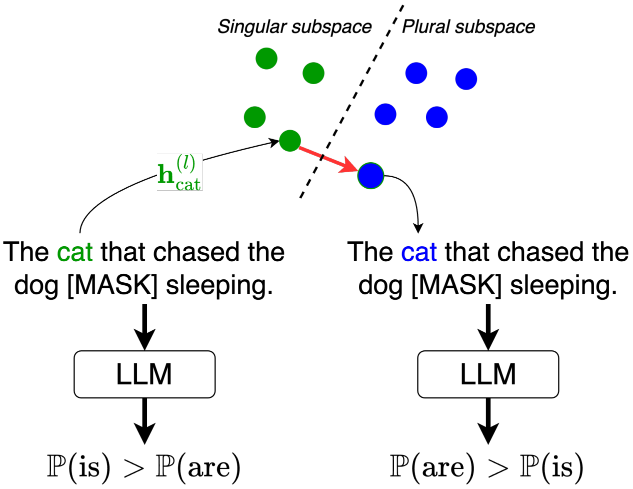
Formally, let $\mathbf{h}_i^{(l)} \in \mathbb{R}^N$ be the hidden representation of the $i$-th token in the input sequence at layer $l$, and let $\mathbf{y}^{(t_i)}$ represent the corresponding value of a linguistic property $t_i$. Following [15], consider gender as an example of such a linguistic property, where $\mathbf{y}^{(t_i)}$ can take values from $\{\texttt{male}, \texttt{female}\}$. The objective is to find a transformation $g : \mathbb{R}^N \to \mathbb{R}^N$ such that the transformed representation $g(\mathbf{h}_i^{(l)})$ no longer encodes information about $t_i$. In other words, after applying $g(\cdot)$, it should not be possible to predict $t_i$ from the transformed representation.

At the same time, $g(\mathbf{h}_i^{(l)})$ should preserve as much other information as possible to avoid degrading the overall performance of the language model. For instance, in the case of gender bias removal, the transformed representation $g(\mathbf{h}_i^{(l)})$ should still be useful for the original task (e.g., text generation or classification) but should no longer encode gender-related information.

To construct such a transformation $g(\cdot)$, the authors first train a linear probing classifier $p$ to predict the linguistic property $t_i$ from the hidden representations. This classifier has an associated weight vector $\boldsymbol{\theta} \in \mathbb{R}^N$, which captures the most predictive direction for $t_i$ in the representation space. This weight vector is used to construct a projection matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$ such that: $\boldsymbol{\theta}^\top(\mathbf{P}\mathbf{h}_i^{(l)}) = 0, \quad \forall i$. This ensures that the transformed representation $\mathbf{P}\mathbf{h}_i^{(l)}$ lies in the null space of $\boldsymbol{\theta}$, effectively removing the information related to $t_i$.

The projection matrix $\mathbf{P}$ is obtained as follows:

**Figure 1:** Illustration of the procedure to test if the LLM uses grammatical number for subject-verb agreement. In the first execution (left), the LLM predicts “is” for the masked position, recognizing the singular subject. After shifting the hidden representation of “cat” to the plural subspace using INLP, the model is asked to predict again (right) and now predicts “are”, aligning with the plural subject. This figure is inspired by [17].


Singular subspace
Plural subspace
$\mathbf{h}_{\text{cat}}^{(l)}$
The cat that chased the dog [MASK] sleeping.
The cat that chased the dog [MASK] sleeping.
LLM
LLM
$\mathbb{P}(\text{is}) > \mathbb{P}(\text{are})$
$\mathbb{P}(\text{are}) > \mathbb{P}(\text{is})$

1. Compute the null space of $\boldsymbol{\theta}$, defined as: $N(\boldsymbol{\theta}) = \{\mathbf{h}_i^{(l)} \mid \boldsymbol{\theta}^\top \mathbf{h}_i^{(l)} = 0\}$.
2. Construct the projection matrix $\mathbf{P}_{N(\boldsymbol{\theta})}$ using the basis vectors of $N(\boldsymbol{\theta})$.
3. Obtain the final transformation by projecting onto the orthogonal complement:$\mathbf{P}_{N(\boldsymbol{\theta})}\mathbf{h}_i^{(l)}$.

To ensure that all traces of $t_i$ are removed, this procedure is applied iteratively. Specifically, additional probing classifiers $p'$ are trained to detect any remaining information about $t_i$ in the transformed representations. If a new classifier $p'$ still predicts $t_i$ with above-random accuracy, a new weight vector $\boldsymbol{\theta}'$ is obtained and used in the next iteration of the null-space projection process. This iterative procedure continues until no further linear information about $t_i$ can be extracted from the representations.

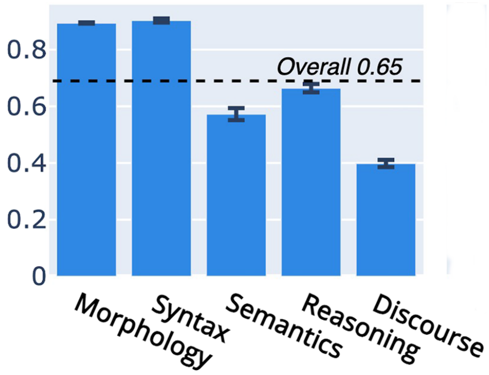
INLP not only enables the complete removal of specific information from hidden representations but also provides a principled framework for controlling how linguistic features are encoded and utilized by LLMs. Crucially, by applying INLP and observing changes in model predictions, one can empirically verify that a given piece of information was indeed being used by the model during inference.

Beyond its application to mitigating bias, INLP has been employed to analyze other linguistic phenomena, such as subject-verb agreement in relative clauses [16, 17] and a broader range of syntactic structures [18]. In these studies, rather than simply removing number information from hidden representations, the authors used INLP to generate counterfactual examples. For example, consider an LLM’s performance in predicting the correct verb agreement with the subject in the sentence “The cat (SUBJECT) that chased the dog [MASK] (VERB) sleeping” as shown in Figure 1. The model is considered to perform well if it predicts $\mathbb{P}(\text{is}) > \mathbb{P}(\text{are})$ for the masked position, recognizing that the subject is singular. To test this, the projected hidden representation of *cat* ($\mathbf{h}_{\text{cat}}^{(l)}$) is shifted within the number subspace identified by INLP, specifically towards the plural subspace (illustrated by the red line in Figure 1). This shift generates a counterfactual example, represented by the blue dot in the plural subspace. If the probabilities change upon re-evaluation using the counterfactual representation instead of the original, it suggests that the model was indeed using number information to predict the correct agreement. This demonstrates that LLMs encode and utilize grammatical number in a structured manner.

## 6. Results on the linguistic competence of language models

Waldis et al. [1] conduct an evaluation using the classifier-based probing approach described in Section 4, analyzing classification performance across various linguistic tasks by treating the model’s final-layer

**Figure 2:** Average task metric for each linguistic phenomena type. Dashed lines are average measure over all phenomena. This figure is taken from Waldis et al. [1].



embeddings as input to the classifiers. The authors measured classification performance by using different metrics according to the analyzed task (e.g. in subject-verb agreement, the task metric is the accuracy of predicting the correct verb given the subject). Building on their findings, here we summarize key insights regarding the linguistic competence of language models; Figure 2 is reported here from Waldis et al. [1] to depict the linguistic task performance.

While language models excel at capturing local syntactic properties, such as part-of-speech tagging, they face greater challenges when tasked with understanding more complex linguistic phenomena like semantic roles and reasoning. For example, consider the sentence *The cat that chased the dog is sleeping*. In this case, the part-of-speech tag of the word *cat* is highly dependent on its surrounding context. The presence of the definite article *the* before *cat* helps the model predict that *cat* is a noun. This is a relatively simple syntactic phenomenon where word dependencies within nearby contexts can be easily captured.

However, when moving to more complex semantic tasks, the challenges become more pronounced. Consider again the sentence *The cat that chased the dog is sleeping*. Now, if we are to assign a semantic role to the words *cat* and *dog*, the task becomes significantly more complicated. The appropriate semantic role for *cat* would be `Agent` (the entity performing the action) , while *dog* would be the `Theme` (the entity affected by the action). While models can easily handle word co-occurrences within the immediate context, they struggle with more distant and intricate relations, such as rhetorical connections or understanding semantic roles. In this regard, language models show a tendency to approximate simpler word dependencies in nearby contexts well, but their performance diminishes when these dependencies extend over larger portions of text or involve more abstract relationships.

An additional consideration is that the linguistic performance of language models can be significantly improved by augmenting them with an encoder module. This is especially true when models are equipped with additional parameters, which allow them to better approximate simpler word co-occurrences. However, when dealing with more complex co-occurrences, such as those found in rhetorical relations or higher-order semantic structures, language models still encounter difficulties.

## 6.1. Linguistic performance in multilingual vs monolingual models

While the studies discussed earlier primarily focus on monolingual (English) language models, a growing body of literature explores how language models' ability to learn and encode linguistic properties varies when considering different languages, particularly in the context of multilingual versus monolingual models.

In their work, [19] propose a framework for assessing syntactic properties in both monolingual and multilingual models, with a particular focus on the subject-verb agreement task. Their findings suggest that monolingual models typically achieve high accuracy on syntactic dependencies without attractors, while showing poorer performance on agreement in object relative clauses. Furthermore, they observe that languages with richer morphology tend to have higher agreement accuracy across various syntactic constructions. In contrast, multilingual models, particularly those in their current form, do not exhibit evidence of positive grammar transfer across languages. Instead, these models often experience harmful interference, which negatively affects their ability to learn linguistic properties effectively across languages.

Regarding the potential for grammar transfer in multilingual models, Mueller et al. [20] explore whether syntactic neurons are shared across a set of high-resource, grammatically similar languages. Their findings reveal significant overlap in neurons responsible for syntactic agreement across languages in autoregressive multilingual models, but not in masked language models. Notably, two distinct patterns of layerwise effects and sets of neurons were found, corresponding to syntactic agreement, depending on whether the subject and verb were separated by other tokens. These results suggest that multilingual models may be capable of sharing some syntactic knowledge across grammatically similar languages, but this is not necessarily the case for all model architectures or language pairs.

While multilingual models exhibit some shared knowledge across languages, these results raise important questions about the impact of multilingual training on grammar proficiency. The evidence suggests that while multilingual models may benefit from shared syntactic structures in some cases, they often struggle to maintain the same level of proficiency in grammar and syntax as their monolingual counterparts. This is especially true when languages with vastly different syntactic structures or morphological richness are included in the training data, as the model may struggle to reconcile conflicting grammatical patterns across languages. Therefore, it appears that multilingual training may not always enhance grammar and syntax capabilities but can instead lead to interference that hinders the model's ability to learn and generalize linguistic properties effectively.

## 7. Conclusion

This paper provides an overview of key studies on evaluating the linguistic performance of language models, covering essential concepts in this area. We discussed how language models process text, the linguistic properties that can be studied, and the tools available for assessing the presence of these properties in model representations, with a particular focus on probing classifiers. Furthermore, we explored methods to evaluate whether the linguistic knowledge encoded in a model is actively used during inference. Additionally, we highlighted techniques like INLP for guiding the model to disregard certain learned information (e.g., gender bias) and examined how shifting representations of linguistic properties (e.g., number) in the opposite direction within their respective subspaces can help evaluate linguistic performance.

Language models are generally better at encoding formal linguistic phenomena, such as morphology and syntax, compared to functional linguistic phenomena like semantics. This trend is especially true for monolingual language models, while multilingual models appear to struggle more with transferring and applying grammatical knowledge across languages.

An important observation made by [1] is that the current evaluations of linguistic performance in language models are limited in scope. Notably, only three language models were probed on more than 20% of the tasks, and only one task (part-of-speech tagging) was evaluated for more than 20% of the models. Furthermore, recent large language models are significantly underrepresented in these evaluations, suggesting a gap in how we assess the linguistic capabilities of state-of-the-art models.

Looking forward, future work can expand the understanding of linguistic competence in language models by considering alternative perspectives. For instance, Chomsky [21] and De Saussure [22] offer distinct views on linguistic competence. Chomsky defines linguistic competence as the unconscious knowledge of language and linguistic performance as the application of this knowledge in actual utterances. In contrast, de Saussure distinguishes between the structured rules and lexicon of language (langue) and its dynamic, social usage (parole), emphasizing the ongoing negotiation and evolution of language within society. The majority of studies covered in this paper follow Chomsky's interpretation, treating language models as static representations of a particular moment in time. However, de Saussure's framework invites a more dynamic perspective on linguistic competence, where the evolution of language knowledge and societal shifts in communication practices are considered. Future research could apply the methods outlined in this paper, but with an eye toward Saussure's view of language. This would help us model the relationship between the language model's interaction with the language it is trained on and whether this interaction aligns with the evolving nature of language as a societal process. Such an approach could offer valuable insights into how language models might better capture the dynamics of language use and evolution over time.

## Acknowledgments

This work was supported in part by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor the Italian MUR can be held responsible for them.

## Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 and NotebookLM in order to: Drafting content, Paraphrase and reword, Improve writing style, Abstract drafting, Grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

[1] A. Waldis, Y. Perlitz, L. Choshen, Y. Hou, I. Gurevych, Holmes A Benchmark to Assess the Linguistic Competence of Language Models, Transactions of the Association for Computational Linguistics 12 (2024) 1616–1647. doi:10.1162/tacl_a_00718.

[2] A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, J. Glass, Identifying and controlling important neurons in neural machine translation, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=H1z-PsR5KX.

[3] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5797–5808. URL: https://aclanthology.org/P19-1580/. doi:10.18653/v1/P19-1580.

[4] H. Sajjad, F. Dalvi, N. Durrani, P. Nakov, On the effect of dropping layers of pre-trained transformer models, Comput. Speech Lang. 77 (2023). URL: https://doi.org/10.1016/j.csl.2022.101429. doi:10.1016/j.csl.2022.101429.

[5] K. Mahowald, A. A. Ivanova, I. A. Blank, N. Kanwisher, J. B. Tenenbaum, E. Fedorenko, Dissociating language and thought in large language models: a cognitive perspective, CoRR abs/2301.06627 (2023). URL: https://doi.org/10.48550/arXiv.2301.06627. doi:10.48550/ARXIV.2301.06627. arXiv:2301.06627.

[6] Y. Adi, E. Kermany, Y. Belinkov, O. Lavi, Y. Goldberg, Fine-grained analysis of sentence embeddings using auxiliary prediction tasks, in: International Conference on Learning Representations, 2017. URL: https://openreview.net/forum?id=BJh6Ztuxl.

[7] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, M. Baroni, What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties, in: I. Gurevych, Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2126–2136. URL: https://aclanthology.org/P18-1198/. doi:10.18653/v1/P18-1198.

[8] I. Tenney, D. Das, E. Pavlick, BERT rediscovers the classical NLP pipeline, in: A. Korhonen, D. R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational

Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, 2019, pp. 4593–4601. URL: https://doi.org/10.18653/v1/p19-1452. doi:10.18653/V1/P19-1452.

[9] J. Hewitt, C. D. Manning, A structural probe for finding syntax in word representations, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4129–4138. URL: https://aclanthology.org/N19-1419/. doi:10.18653/v1/N19-1419.

[10] Y. Belinkov, Probing classifiers: Promises, shortcomings, and advances, Computational Linguistics 48 (2022) 207–219. URL: https://aclanthology.org/2022.cl-1.7/. doi:10.1162/coli_a_00422.

[11] F. Dalvi, N. Durrani, H. Sajjad, Y. Belinkov, A. Bau, J. Glass, What Is One Grain of Sand in the Desert? Analyzing Individual Neurons in Deep NLP Models, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 6309–6317. doi:10.1609/aaai.v33i01.33016309.

[12] L. Torroba Hennigen, A. Williams, R. Cotterell, Intrinsic probing through dimension selection, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 197–216. URL: https://aclanthology.org/2020.emnlp-main.15/. doi:10.18653/v1/2020.emnlp-main.15.

[13] N. Durrani, H. Sajjad, F. Dalvi, Y. Belinkov, Analyzing individual neurons in pre-trained language models, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 4865–4880. URL: https://aclanthology.org/2020.emnlp-main.395/. doi:10.18653/v1/2020.emnlp-main.395.

[14] O. Antverg, Y. Belinkov, On the Pitfalls of Analyzing Individual Neurons in Language Models, https://arxiv.org/abs/2110.07483v3, 2021.

[15] S. Ravfogel, Y. Elazar, H. Gonen, M. Twiton, Y. Goldberg, Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection, 2020. doi:10.48550/arXiv.2004.07667. arXiv:2004.07667.

[16] S. Ravfogel, G. Prasad, T. Linzen, Y. Goldberg, Counterfactual Interventions Reveal the Causal Effect of Relative Clause Representations on Agreement Prediction, in: A. Bisazza, O. Abend (Eds.), Proceedings of the 25th Conference on Computational Natural Language Learning, Association for Computational Linguistics, Online, 2021, pp. 194–209. doi:10.18653/v1/2021.conll-1.15.

[17] S. Hao, T. Linzen, Verb Conjugation in Transformers Is Determined by Linear Encodings of Subject Number, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 4531–4539. doi:10.18653/v1/2023.findings-emnlp.300.

[18] K. Lasri, T. Pimentel, A. Lenci, T. Poibeau, R. Cotterell, Probing for the Usage of Grammatical Number, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 8818–8831. doi:10.18653/v1/2022.acl-long.603.

[19] A. Mueller, G. Nicolai, P. Petrou-Zeniou, N. Talmina, T. Linzen, Cross-linguistic syntactic evaluation of word prediction models, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5523–5539. URL: https://aclanthology.org/2020.acl-main.490/. doi:10.18653/v1/2020.acl-main.490.

[20] A. Mueller, Y. Xia, T. Linzen, Causal Analysis of Syntactic Agreement Neurons in Multilingual Language Models, 2022. doi:10.48550/arXiv.2210.14328. arXiv:2210.14328.

[21] N. Chomsky, Aspects of the Theory of Syntax, 50 ed., The MIT Press, 1965. URL: http://www.jstor.org/stable/j.ctt17kk81z.

[22] F. De Saussure, Cours de linguistique générale, volume 1, Payot, Paris, 1916.