Linguistic Generalization in Transformer-Based Neural Language Models

Karim Lasri

PSL UNIVERSITÉ PARIS

UNIVERSITÀ DI PISA

THÈSE DE DOCTORAT

DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Linguistic Generalization in Transformer-Based Neural Language Models

Defended by

Karim Lasri

Le 01/02/2023

Doctoral school n°540

Sciences et Lettres - ED

Specialty

Sciences du Langage

Prepared at

LATTICE - ENS/PSL, CNRS, Univ. Sorbonne Nouvelle

CoLing Lab - University of Pisa

Jury Composition :

Justine CASSELL Inria	*President of the jury*
Marco BARONI ICREA, Pompeu Fabra University	*Rapporteur*
Shalom LAPPIN University of Gothenburg	*Rapporteur*
Alysson ETTINGER University of Chicago	*Examiner*
Alessandro LENCI University of Pisa	*Examiner*
Thierry POIBEAU Ecole Normale Supérieure	*Examiner*

ENS | PSL

Abstract

Neural language models are commonly deployed to perform diverse natural language processing tasks, as they produce contextual vector representations of words and sentences which can be used in any supervised learning setting. In recent years, transformer-based neural architectures have been widely adopted towards this end. After being pre-trained with a generic language modeling objective, they achieve spectacular performance on a wide array of downstream tasks, several of which should in principle require knowledge of sentence structure. As these models are not explicitly supervised with any grammatical instruction, this suggests that linguistic knowledge emerges during the pre-training stage. The nature of the linguistic abilities acquired during training is still scarcely understood, as these models are generally used as black boxes. Their decisions are hard to interpret, as they generally possess a great number of parameters (up to 10^{12} for the most recent architectures) and learn very complex functions. These observations led to the emergence of a growing body of research aimed at uncovering the linguistic abilities of such models. While this literature is very abundant, the epistemic grounds of the different methodologies are not translatable into each other, which underlines the need to formulate more clearly the questions addressing the capture of linguistic knowledge. To this end, we identify the different stances on the greater problem: in addition to downstream performance, evidence for a trained model's linguistic abilities can be sought in its components, representations and surface behavior. Throughout the thesis, we attempt bridging the epistemic gap between these facets by formulating explicitly the relations which lie between these different subproblems. In particular, we adopt three levels of analysis to understand neural language models as information processing systems, from the highest to the deepest: the behavioral level, the algorithmic level, and the implementational level. In our framework, our departing point to investigate linguistic abilities is surface linguistic generalization. The empirical portion of this thesis first presents behavioral tests targeting a syntactic ability, to investigate the nature of the information processed at the algorithmic level - in particular we provide evidence for the entanglement between syntactic and semantic processes. We then show that behavioral tests can be limited to inform us on the nature of the information processed by

the model. In particular, we provide evidence that surface behavior on a structure-sensitive task can be approximated to a good extent without relying on word order. Faced with this observation, we make the case for targeted causal interventions, and investigate a neural language model's reliance on position information on the masked language modeling task. In doing so, we show that the model increasingly relies on word order as the number of masked tokens increases during training. We also demonstrate the power of causal interventions to assess the usage of targeted information. We then discuss how causal interventions, coupled with targeted behavior tests, can inform us on the model's linguistic abilities at the three levels mentioned previously. Indeed, the introduced framework allows us to (i) find the neural substrate responsible for representing or transferring linguistic information, (ii) assess the presence of certain representations or operations at the algorithmic level and (iii) determine the causal influence of these representations and operations over the model's behavior. We discuss how this analysis can be performed and apply the methodology introduced to shed light on the encoding and usage of grammatical number information on the subject-verb agreement task. In doing so, we bridge the gap between representation-oriented and behavior-oriented analyses of linguistic knowledge.

Keywords : Deep Learning, Language Model, Linguistic Knowledge, Natural Language Processing, Generalization

Résumé

Les modèles de langage neuronaux basés sur des transformeurs sont couramment déployés pour effectuer diverses tâches de traitement automatique des langues, car ils produisent des représentations vectorielles de textes qui peuvent être utilisées dans le cadre d'un apprentissage supervisé. Après avoir été pré-entraînés comme modèles de langage génériques, ils atteignent des performances spectaculaires sur un large éventail de tâches, dont plusieurs nécessitent en principe des connaissances sur la structure des phrases. Ces modèles ne sont pas explicitement supervisés avec la moindre instruction grammaticale, ce qui suggère que ces connaissances émergent pendant la phase de pré-entraînement. La nature des capacités acquises est peu comprise, car ces modèles sont généralement utilisés comme des boîtes noires. Leurs décisions sont en outre difficiles à interpréter, en raison de leur grand nombre de paramètres (jusqu'à 10^{12} pour les architectures les plus récentes) et de la complexité des fonctions apprises. De ce constat a émergé un nombre important de travaux de recherche visant à mieux comprendre les capacités linguistiques de ces modèles. Bien que cette littérature soit abondante, les paradigmes épistémologiques sous-tendant les différentes méthodologies ne sont pas compatibles entre eux, ce qui souligne la nécessité de formuler plus clairement les questions portant sur l'acquisition des connaissances linguistiques. Tout au long de la thèse, nous tentons de combler le fossé épistémique entre ces facettes en formulant explicitement les relations qui lient les approches existantes. En particulier, nous adoptons trois niveaux d'analyse pour comprendre les modèles de langage neuronaux en tant que systèmes de traitement de l'information, du plus haut au plus profond : le niveau comportemental, le niveau algorithmique et le niveau de l'implémentation. La partie expérimentale de cette thèse introduit d'abord des tests comportementaux évaluant le niveau d'abstraction syntaxique, afin d'étudier la nature des informations traitées au niveau algorithmique - en particulier nous mettons en évidence l'intrication entre les processus syntaxiques et sémantiques. Nous montrons ensuite que les tests comportementaux sont limités pour nous renseigner sur la nature de l'information traitée par le modèle. En particulier, nous montrons que les prédictions sur une tâche dépendant de la structure des phrases peuvent être approximées sans faire usage d'information sur l'ordre

des mots. Face à ce problème, nous soulignons la nécessité de réaliser des interventions causales et nous étudions l'utilisation de l'information positionnelle par un modèle de langage neuronal. Nous prouvons que le modèle s'appuie graduellement sur l'ordre des mots à mesure que le nombre de mots masqués au cours de l'entraînement augmente. Nous démontrons également le pouvoir explicatif des interventions causales pour évaluer l'utilisation d'information de manière ciblée et non équivoque. Nous montrons ensuite comment les interventions causales, couplées à des tests de comportement, peuvent nous renseigner sur les représentations linguistiques du modèle aux trois niveaux mentionnés précédemment. Le cadre introduit permet (i) de mettre en évidence le substrat neuronal responsable de la représentation et du transfert d'information linguistique, (ii) d'évaluer la présence de représentations et d'opérations au niveau algorithmique et (iii) de déterminer l'influence causale de ces dernières sur le comportement du modèle. Nous appliquons ensuite la méthodologie introduite pour mettre en lumière l'encodage et l'utilisation de l'information sur le nombre grammatical dans le cadre d'une tâche d'accord sujet-verbe. Ainsi, nous comblons le fossé entre les études sur les capacités d'abstraction linguistique focalisées sur les représentations du modèle et celles axées sur son comportement de surface.

Mots clés : Apprentissage Profond, Modèle de Langage, Connaissance Linguistique, Traitement Automatique des Langues, Généralisation

Acknowledgements

First, I would like to express my gratitude to my PhD advisors, Alessandro Lenci and Thierry Poibeau without whom this thesis would not have been possible. I am grateful for that they gave me a unique opportunity to engage in this three-year long challenge. I am additionally thankful that they made it possible to change my work atmosphere during these years, when I would be welcomed with warmth and hospitality by Alessandro in Italy.

Further, I would like to thank all those with whom I had passionate conversations before starting the PhD. My deepest gratitude goes to Jean-Pierre Nadal, and Sabine Ploux who offered me their kindness, help and support when I was considering starting a PhD as an intern at the CAMS, EHESS. I can't thank enough Sabine, along with Raffaella Bernardi, who kept on being supportive and warm throughout the PhD in my "comité de suivi". I would also like to warmly thank Salvador Mascarenhas, and Emmanuel Dupoux who guided me before my PhD started, and who gave me several relevant reading suggestions which undoubtedly shaped my later research.

I would like to thank Marco Baroni, and Shalom Lappin, for accepting to review this manuscript, and also Justine Cassell and Alysson Ettinger for being part of my defense committee. Additionally, I am grateful for their contributions to the questions addressed in this PhD, which gave me matter for my thinking.

I would like to further thank all the collaborators who inspired me during my thesis with their collaborative mindset, specially Dieuwke Hupkes, Mario Giulianelli, Verna Dankers and Koustuv Sinha from the GenBench project, but also Olga Seminck who was very supportive and helped me collect crowd-sourced data, and Paolo Pedinotti for his kindness and for helping me set up a behavioral test experiment. More broadly, I would like to thank all my collaborators for all that I learned with them, but also all those with whom I shared memorable experiences these years, my colleagues at Lattice, the CoLing lab, the WorldBank, the ENSAE teaching staff, and also people from the ALMAnaCH study group.

I would like to thank all my friends, especially les *froeurs*[1], my musical buddies, Valentin, Ludo, and all the others with whom I had so many good times these years

[1] Portmanteau word obtained by blending *frères* and *soeurs*.

full of discoveries, intense conversations, and nonserious matters, thank you for all the laughters and fun.

Last but not least, I am deeply thankful to my amazing family, parents, sisters, but also aunts, uncles, and cousins, for their continuous presence and unconditional support, especially Sophia, who made everything when she could for me to work on my thesis in the best conditions.

Funding

This work was funded by the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

Contents

List of figures

List of Tables

Chapter 1

General Introduction

1.1 Motivation

Large pre-trained neural language models have become ubiquitous in recent NLP pipelines [Devlin, 2019a; Liu, 2019b; Raffel, 2020]. While they require large computing resources to be trained, their weights are often made accessible publicly [Wolf, 2020], such that their knowledge can be easily transferred to a wide array of tasks by practitioners. Using pre-trained transformer language models is appealing as fine-tuned models seem to be robust on a number of dowsnstream NLP applications, even with relatively small dataset sizes. Pre-trained language models are becoming increasingly large, as seen in fig. 1.1, as increasing the number of parameters so far resulted in better perplexity of the trained model. Yet, we still lack a precise understanding of how such models generalize.

As evaluation of pre-trained language models is traditionally performed by taking a look at opaque measures for approximating the data's distribution, such as perplexity, it is hard to really know what this performance really means.

Facing the inability to directly access the causes underlying a neural model's decisions, a lot of ink has been spilled on the alleged capacities of pre-trained transformer language models. Some authors proposed to benchmark neural language models by transferring their knowledge to downstream tasks [Wang, 2018; Wang, 2019] , but performance on these tasks is also opaque.

Such tasks often involve high-level linguistic abilities that in principle require mobilizing a large toolbox of lower-level linguistic abstractions to be solved, which makes it hard to pinpoint exact reasons for failures, or ensuring that success on these tasks really reflects that models are truly learning the task – i.e. that it has acquired and combines lower-level linguistic abilities instead of making use of heuristics [McCoy, 2019].

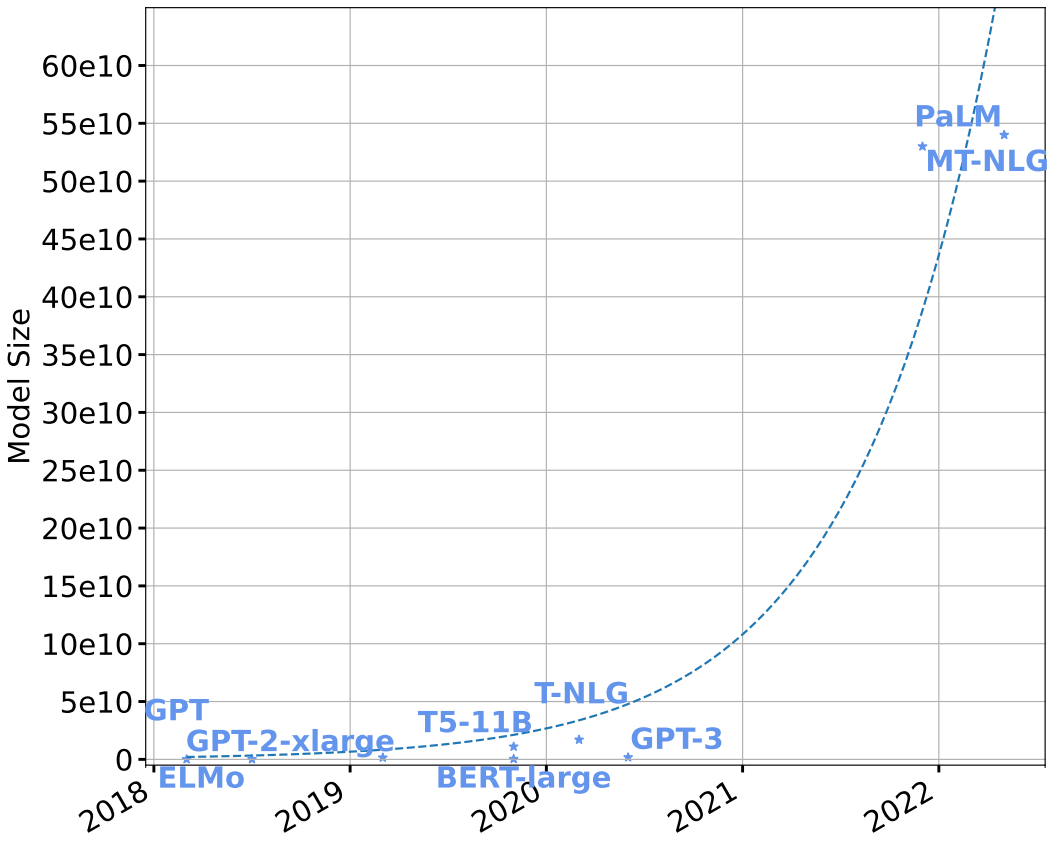


Figure 1.1: Evolution of neural language model sizes over the past years

The resulting discussions divide the research community regarding whether those models abstract away from data by capturing linguistic generalizations, as pointed out by former research studies. Indeed, more recent work showed them to make use of shallow heuristics to solve some tasks, underlining limitations of previous analyses carried to evaluate neural language models. If we understood better how pre-trained language models generalize, we could potentially diagnose an architecture's ability to acquire linguistic knowledge and perform better architecture search, reducing the computing resources required to train models. Further, if the purpose of pre-trained language models is to be deployed on downstream applications, we can hardly be confident in using them if their performance on the surface does not reflect that the model abstracted away from training examples – that is if they just approximate the task on the provided data. Still, a deeper understanding of the inner workings of such models is still largely lacking and they remain black boxes in most the vast majority of their applications. By their nature, neural models acquire knowledge that is not easily accessible after they have been trained with a specific objective, which raised the need to evaluate their robustness when they seemingly perform well on their training task [Alain, 2016; Adi, 2017; Elazar, 2021].

Designing proper analyses to diagnose the decisions of neural networks is challenging due to the complexity of the neural architectures under investigation, but also the variety in the data that lies in the task space. As a tentative remedy to this problem, a plethora of smaller-grained analysis techniques have been proposed—these techniques are usually referred to as probes. Regardless of this diversity of methods, there is great room for gaining a better understanding regarding the nature of the linguistic knowledge transformer-based neural language models acquired after training, as well as where and how these models encode such linguistic knowledge.

This PhD's goal is to gain better understanding of the linguistic knowledge captured by transformer-based neural language models. This is achieved by providing more systematic formal and methodological tools aimed at characterizing the capture of linguistic knowledge that Neural Language Models (NLMs) acquire during their training, with a focus on the BERT architecture [Devlin, 2019a], a transformer-based model which encountered tremendous success since its release.[1]

[1]The model's release paper reached more than 47K citations when this thesis was written.

1.2 Research Questions

As seen in the previous section, there is great room in understanding the linguistic abilities captured by neural language models after pre-training, in particular transformer-based architectures which support state-of-the-art models. We frame the research question as follows: **What aspects of Linguistic Generalization do Transformer-based Neural Language Models acquire?** In the following paragraphs, we briefly introduce the various concepts mentioned here.

1.2.1 Transformer-based Neural Language Models

Pre-trained language models are neural models trained on a language modeling task, with the objective of predicting a word w given its context c. The goal of pre-training language models is to obtain high-quality sets of representations for input sentences, such that the knowledge acquired in such representations during training can be transferred to a wide array of downstream tasks. Such tasks are often higher-level linguistic tasks ranging from language inference to question answering, requiring the acquisition of deep linguistic knowledge. During pre-training, language models approximate the true language distribution p by estimating:

$$\forall w \in \mathcal{V}, q(w \mid c) \approx p(w \mid c)$$

Transformer language models are built by stacking transformer layers [Vaswani, 2017], the distinctive feature of which is the attention mechanism. While language models have existed for longer than the transformer itself, first supported by other architectures such as recurrent neural networks, transformers have become ubiquitous in pre-training based approaches and in other areas of the NLP spectrum because of their ability to outperform previous models on a wide array of tasks, hence the choice to analyze this family of models in this thesis. A more detailed introduction to transformer-based neural language models in chapter 2.

1.2.2 Generalization in Statistical Learning

In order to test whether a neural model generalizes, one typically evaluates the model based on its performance on an unseen, held-out test set after training. Recently, this widely accepted procedure has been put under question. Good performance on a test set which is sampled from the same distribution as the training set has been shown to not necessarily reflect generalization.

Memorization [Zhang, 2017; Zhang, 2021] discuss that neural networks with enough capacity can just memorize labels during training, even when they are assigned randomly to each observation. This raises a serious concern regarding the true abilities of neural language models when their performance on the test set indicates that they generalize. A pure memorizer in principle shouldn't be able to guess correct answers on test time as it is in principle exposed to novel, unseen data. Yet recent studies showed that large pre-trained language models did memorize a significant amount of their training data [Carlini, 2020; Anonymous, 2023]. While this work raised concerns about memorization as an undesired property of neural language models, other authors instead tried build on memorization, coupled with interpolation, to enhance pre-trained NLMs [Khandelwal, 2020]. Language models could partly rely on memorizing lexical patterns without capturing sentence structure if such associations are repeated in both its train and test sets. A complete picture on the role of memorization in NLMs is yet to be explored, especially its relationship to generalization in the context of language modeling.

In-domain vs. out-of-domain generalization Another line of inquiry in the machine learning literature investigates the robustness of models to data shifts. These shifts correspond to changes in the distribution of the data seen at test time, when compared to the training distribution. Shifts can be roughly defined as cases where:

$$p(X_{train}, Y_{train}) \neq p(X_{test}, Y_{test}) \tag{1.1}$$

However, this framework assumes that models are trained and evaluated against a single task. On the other hand, language models are usually pre-trained on (masked) language modelling and tested, or even fine-tuned on different tasks, sometimes requiring a change in the architecture, output space, and therefore the random variables underlying the different tasks. While this evaluation method can be used to evaluate robustness of a pre-trained model for each task at fine-tuning time, it remains blind to deep confounds that arise from the pre-train/transfer paradigm. These will be discussed in the following sections.

Under these circumstances, we should be extremely cautious when claiming that a model is generalizing given the only observation that it reached good performance on a given downstream task. As such observation seemingly have low epistemic value regarding whether the model is truly generalizing, we discuss hereafter how linguistic generalization can be otherwise assessed.

1.2.3 Linguistic Generalization

As previously seen, generalization on a given downstream task, even when it is designed to target a specific high-level linguistic task (e.g. natural language inference, or paraphrase) is hard to assess precisely because it is not easy to ascertain that the model has learned the task instead of just memorizing answers or shallow heuristics. While neural models are generally trained to perform a specific single task, pre-trained language models are trained with the goal of capturing all sorts of abstractions from language in their representing spaces. It follows that the evaluation of linguistic generalization in a pre-trained model should thus differ from traditional evaluation of a neural model's performance on a learning task, thus linguistic knowledge should **not** be sought using traditional assessments of generalization performance in neural models. In turn, we propose to evaluate the linguistic capacities of pre-trained models using fine-grained analysis methods, at a lower level than general-purpose generalization assessment in the machine learning literature, raising two separate questions, making this thesis two-fold:

1. What aspects of linguistic generalization are captured by a given transformer-based neural language model? This is a vast question in itself as it requires targeting specific linguistic phenomena and designing proper analyses to assess their capture by a neural language model.

2. How should we evaluate the linguistic abilities of neural language models? The literature is overabundant in methodologies, which calls for a better characterization of their epistemic value – that is how precisely they inform us on our neural language model's capacities. Designing proper analyses aimed at understanding how such neural networks work in practice is tedious due to their complexity. Modern Neural Language models are supported by very complex functions with huge numbers of parameters, which makes their decisions opaque. One needs targeted analyses to pinpoint granular abilities, in order to understand what these networks capture.

Note that these two questions are not orthogonal, as we require a better understanding of evaluation methods to make claims regarding the emergent linguistic abilities possessed by neural language models.

This second question in turn can be broken down as follows:

1. Does my pre-trained model's behavior show evidence supporting aspects of linguistic generalization on targeted prediction settings (in its training setting – i.e. language modeling, or token prediction tasks)?

2. Do my model's intermediate representations capture abstractions of linguistic properties?

It is worthy to note that while the first question still treats the model as a black box, it allows for more targeted evaluation than downstream tasks, and avoids confounds discussed above. The second question in-turn goes beyond a black-box usage of neural models, and invites us to take a look inside the neural architecture. We further propose to bring answers to the first question by evaluating the model's behavior on tests that are designed to fit bi-directional transformers' training regime, masked language modeling. As for the second question, we propose to examine the extent to which intermediate representations capture linguistic abstractions mobilized by the model to produce a given behavior.

1.3 Structure of this Thesis

In chapter 2, we first equip the reader with the basic concepts needed throughout this thesis. We first present the objects under investigation, their specificity and their purpose, and then review the literature that attempted bringing answers to our research question.

In chapter 3, we build on our synthesis of the previous literature and formulate a comprehensive analysis of the question under investigation in this thesis. We draw inspiration from both linguistic theory and principles of neuroscience, and discuss how our quest and linguistic theories can contribute to each other. We further adopt a systemic view on the capture of linguistic knowledge, in which we define various levels of analysis on that question, each with a delimited scope, but all connected to the NLM's purpose as an information processing system optimized on a given objective. This leads us to naturally describe the relationship that these levels share with each other in that broader view. In doing so, our first main contribution consists in proposing a formulation of our research question which bridges the epistemic gaps between the different approaches identified in the previous literature. We further taxonomize behavioral constraints useful to our future investigations, as well as linguistic units which can be represented in layers – those are the building blocks to understand the strategies employed by our NLM to produce certain behavior.

We then present first results derived from behavioral tests to investigate the nature of a NLM's acquired linguistic abilities in chapter 4 and chapter 5. Specifically, we first provide evidence for the entanglement between semantic and syntactic processes on a syntactic task, subject-verb agreement. We then show that on other

tasks requiring sentence structure, the model's behavior is approximated to a good extent by a co-occurrence based model deprived from word order information.

This leads us to discuss the limitations of behavioral tests in informing us on the algorithmic processes supporting a model's decisions. To circumvent this issue, we introduce causal tests which allows us to target the usage of specific information and make more precise algorithmic claims. We illustrate this with a case study on word-order usage in chapter 6, and show that NLMs do rely on position information, but only under a certain amount of masking. This shows them to be equally capable of robust memorization, and to abstract away from word order information, a feature necessary to capture structure-sensitive aspects of languages.

In chapter 7, we build on the introduced causal account and couple it with targeted behavioral tests to formulate a functionalist view on a NLM's components. In this view, we wish to uncover representations and transforms which play a causal role on targeted linguistic tasks. The purpose of this formulation is to gain a better understanding of the algorithmic processes supporting the model's decisions when they hint at the capture of linguistic abilities. Specifically, we introduce a methodology aimed at assessing the capture of linguistic representations in the model's layers, mobilized to support a given decision in the context of a linguistic task. In chapter 8, we finally apply such methodology on a case study targeting the encoding of grammatical number. The causal relations found between representations in the model's layers and its surface behavior constitute novel, strong proof for the acquisition of linguistic properties during training, rather than reliance on shallow memorization.

Figure 1.2: "AI with a maze on its head by Georgia O'Keefe", K.L. x DALL·E 2 [Ramesh, 2022]

Chapter 2

Background: Language Modeling, Transformer-based Architectures, and Probing Methods

Goals

The goal of this chapter is to give the reader some background before we articulate the questions asked in this thesis. We first introduce language modeling, as it is the task learned by the neural models which we will analyze. In particular, we present neural language models, along with their usual usage and purpose. We then introduce transformer-based neural language models, a subtype of neural language models as well as their distinctive characteristics. Finally, we present a taxonomy of the different methodologies used in previous research studies aimed at examining the linguistic knowledge captured by such architectures.

Contents

2.1 An Introduction to Language Modeling

2.1.1 Notations

In natural language processing, modeling a language is equivalent to estimating a probability distribution over sentences in that language. As probability distributions are mathematical functions, we first need to formalize natural languages as mathematical objects. Generally speaking, natural languages can be seen as communication systems which emerged naturally in humans.

We consider the vocabulary $\mathcal{W}$ of all words occurring in a given language $\mathcal{L}$. Denoting $\mathcal{W}^*$ the set of all possible sequences $\mathcal{W}^* = \cup_{n>0}\{(w_1, \ldots, w_n) \in \mathcal{W}^n\}$, it follows that $\mathcal{L}$ verifies:

$$\mathcal{L} \subset \mathcal{W}^*$$

A *language* $\mathcal{L}$ can be formalized as a set of word sequences. Under this set-based formal definition, there exists a binary distinction between sentences from $cal L$ and other word sequences. An intuition on what this binary distinction could be, one could see it as the quality of sentences which could be written by a speaker of that language, and understood by another in a given context. Another view on this binary distinction is associated to grammatical well-formedness, a property of sentences which respect grammatical constraints. Note that this latter view is slightly different from the former as it does not necessarily imply meaningfulness.[1]

Under this binary view, it is usual to call the property defining word sequence in $\mathcal{W}^*$ which belong to $\mathcal{L}$ *well-formedness*. Well-formedness in this case refers to a property, such that it is true for any sentence belonging to a considered language.

Well-formedness is hence the property $T^{\mathcal{L}}$ such that:[2]

$$\begin{cases} \forall s \in \mathcal{L} \subset \mathcal{W}^*,\ T^{\mathcal{L}}(s) = \text{True} \text{ and} \\ \forall s \in \mathcal{W}^* \setminus \mathcal{L},\ T^{\mathcal{L}}(s) = \text{False} \end{cases}$$

It is widely assumed by linguists who subscribe to such view that *well-formedness*, as a property of sequences $s \in \mathcal{W}^*$ can be seen as resulting from a set of constraints, verified by sentences in $\mathcal{L}$ – those can be referred to as *linguistic constraints*. Grammars aim to describe such constraints for a given language, such that they capture sentences in $\mathcal{L}$. Examples of such constraints will be given in section 3.4.3. In practice, grammars do not perfectly capture those as it is very difficult

[1] While looking like a definition, this is rather a formal view on languages, that there exists a binary property distinguishing two sets, sentences in $\mathcal{L}$ and ill-formed word sequences in $\mathcal{W}^{*\setminus\mathcal{L}}$. We discuss this in the next chapter.

[2] There are also continuous views on well-formedness, which attempt accounting for evidence for gradience in acceptability judgements.

to enumerate precise, explicit constraints that define $\mathcal{L}$, whether they are syntactic or semantic constraints. The probability distribution of sentences $s \in \mathcal{L}$ estimated over large corpora comes handy, due to its ease to deploy in practice, and as this suppresses the necessity to feed a model explicit rules about the language that is modeled, especially knowing that the description of constraints found in grammars are imperfect models of language.

2.1.2 Language modeling

Language modeling adopted a probabilistic view to determine which sentences are part of a given language, considering an unknown true probability distribution over sequences p^{true}.

Property 2.1.1. The true probability distribution p^{true} over word sequences is such that:

$$\begin{cases} \forall s \in \mathcal{L} \subset \mathcal{W}^*,\ p^{\text{true}}(s) > 0, \text{ and} \\ \forall s \in \mathcal{W}^* \setminus \mathcal{L},\ p^{\text{true}}(s) = 0 \end{cases}$$

As this distribution is unknown, the goal of language modeling is to approximate such unknown probability distribution by learning an estimate over a large corpus of text. This property has certain limitations however when dealing with natural language corpora. Ideally, in an arbitrarily large set of sentences assumed to be well-formed, there should be no ungrammatical sentence. However, ill-formed word sequences are unlikely to be absent from real large corpora used to model languages. Additionally, language models often learn from limited, sparse data, and the resulting probability distribution needs to be smoothed to estimate probabilities over unseen word sequence. As a consequence to these two facts, language models typically assign non-zero probabilities to ill-formed sentences. Additionally, it is not reasonable either to expect that a language model should assign an arbitrarily low probability to ill-formed sentences by defining a threshold, as this would imply a finite set of well-formed sentences with a probability higher than this threshold.[3]

Language modeling consists in approximating p^{true} using an estimate $p^{\text{model}} \approx p^{\text{true}}$, defined over any sequence of $\mathcal{W}^*$. Given a sequence $s = (w_1, \ldots, w_{|s|})$, its probability can be decomposed as follows, using the **chain rule of probability**:

$$p(w_1, \ldots, w_n) = p(w_1)p(w_2|w_1)p(w_3|w1w_2) \ldots p(w_n|w_1w_2 \ldots w_{n-1}) \tag{2.1}$$

One could argue that such estimates can theoretically not approximate the underlying distribution of sentences. For instance, the number of sentences in a given

[3] For more details on this argument, see [Lau, 2016].

language can be infinite even though the lexicon is not. In practice, for probabilities to be estimated one often has no choice but to constrain a model to estimate the probability of sequences under a certain maximum length. This assumption is acceptable as sentences above a certain length are almost never used in any context.

Noam Chomsky himself criticized statistical approximations of language. Quoting him, from a conversation with Steven Pinker at the Brains, Minds, and Machines symposium held during MIT's 150th birthday party:[4]

"It's true there's been a lot of work on trying to apply statistical models to various linguistic problems. I think there have been some successes, but a lot of failures. There is a notion of success ... which I think is novel in the history of science. It interprets success as approximating unanalyzed data."

2.1.3 *n-gram* modeling

In the early era of natural language processing, *n-grams* have been used as a Markov model to approximate the true distribution of a given language. Using an estimate of the probability of n-uplets, one can use the Markov approximation of the formula in eq. (2.1). The n-gram probability is simply computed as follows:

$$p(w_n|w_1,\ldots,w_{n-1}) = \frac{count(w_1,\ldots,w_n)}{countw_1,\ldots,w_{n-1}} \tag{2.2}$$

For example, if order to approximate the true probability using 2-grams (usually referred to as bigrams), eq. (2.1) can be approximated as follows:

$$\begin{aligned} p(w_1,\ldots,w_n) &= p(w_1)\ p(w_2|w_1)\ p(w_3|w1w_2)\ \ldots\ p(w_n|w_1w_2\ldots w_{n-1}) \\ &\approx p(w_1)\ p(w_2|w_1)\ p(w_3|w_2)\ \ldots\ p(w_n|w_{n-1}) \end{aligned} \tag{2.3}$$

As an example, given the sentence s :="I saw a deer yesterday", its probability can be approximated as:

$$p(s) \approx p(\text{yesterday}|\text{deer})p(\text{deer}|\text{a})p(\text{a}|\text{saw})p(\text{saw}|\text{I})p(I)$$

This is obtained by approximating probabilities of completions given left-contexts by the probability of completions given their immediate left neighbor, which is estimated in a bigram model. For example, given the left context c :="I saw a", the probability $p(\text{deer}|\text{I saw a})$ is approximated as $p(\text{deer}|\text{a})$.

[4]The transcript is available at http://languagelog.ldc.upenn.edu/myl/PinkerChomskyMIT.html

While this approximation has been widely used for language modeling before the appearance of neural models, it fails to capture long range dependencies and leaves no choice but to use heuristics to find a maximum probability sequence, as the combinations of n-grams grow exponentially with the sequence length. Indeed, one has no choice but to constrain the search space to output a maximum probability completion given a context, as it is not possible to hold in memory the probability of all intermediate sequences.[5] This is one of the many difficulties alleviated by training neural architectures to learn a language model, in addition to the flexibility that these models offer by producing vector representations, which we review next.

2.1.4 Neural language models

Artificial Neural Networks Schematically speaking, an artificial neural network (ANN) is an interconnected set of neurons, initially inspired by a simplification of neurotransmission in the brain.[6] The simplest neural architecture, the perceptron, has first been proposed by [Rosenblatt, 1958]. In the general case, connections between neurons can form cycles[7], but in the simplest forms of ANNs, information moves in only one direction and is passed to each layer only once, as displayed in fig. 2.1. Here, each circle is a node that represents an artificial neuron and each arrow represents a connection from the output of one artificial neuron to the input of another.

Thus, neural networks can be seen as a parametric function $g_{\theta \in \Theta}$ resulting from the composition of layers, which themselves are simple functions $(f_i)_{1 \leq i \leq n_{layers}}$. Each of these layers computes a transformation of the previous layer's output, and at the last layer, the last representations are converted into the neural network's output, the format of which depends on the task under consideration. ANNs are

[5] The number of combinations using n-grams to construct a completion of length l, given a vocabulary size m is $(m^n)^{l-n} = m^{n(l-n)}$. For example, if $m = 10000$, $n = 3$ and $l = 5$, this number is about $10000^{3*(5-3)} = 10^{24} \approx 2^{80}$.

[6] ANNs differ in many aspects from real neural networks, which are far more complex and involve a far greater number of parameters at lower scales. In the most common type of synapses, chemical synapses, neurotransmission is a complex process involving several ionic concentrations which affect post-synaptic signal. Most ANNs do not even model pre-synaptic and post-synaptic signal using a time dimension but simply as a real-valued number for each neuron, which might have profound implications on the encoding of information. The synaptic strength, also modeled by one single real-valued number in ANNs, is updated using optimization algorithms while it is subject to different biological mechanisms involving short-term and long-term changes in real neural networks. While ANNs are probably not as expressive as real neural networks, they comprise a variety of architecture and draw inspiration from that a large set of neurons transmitting signal – as simple as this principle can be – can be optimized with the goal of producing desirable responses to given stimuli – as complex as these responses can be.

[7] This is the case for recurrent neural networks

often trained to learn a task, that is to map certain inputs to outputs. One way to achieve this is to use a labeled dataset $\mathcal{D} = (\mathbf{x}_i, y_i^{true})_{1 \leq i \leq n}$ where each input is assigned a desired output by annotating the data, a setting called supervised learning. The main optimization procedure, known as Empirical Risk Minimization (ERM), consists in minimizing a given loss function L across examples in $\mathcal{D}$, generally the cross-entropy between the model's outputs and the expected labels:

$$min_{\theta \in \Theta}[\sum_{(x,y) \in \mathcal{D}} L(g_\theta(x), y)]$$

Supervised learning tasks can further be distinguished based on whether the outputs of the network are discrete or continuous. In the former case, the network learns a classification task. In the latter case, it learns a regression task. There also exist other paradigms for an ANN to learn from data. In unsupervised settings, ANNs learn to extract certain patterns from input data (e.g. clusters, or anomalies), without any annotations.

Neural Language Models In language modeling, the task learned by a neural model is the following: given a sentence, the model is asked to predict a word or a set of words in that sentence, given its context. As manually annotating inputs can be a long and costly process yielding limited data, pairs of input contexts and output predictions can be extracted automatically from large corpora of sentences assumed to be well-formed.[8] In such paradigm, sometimes referred to as self-supervised learning, no human-annotated labels are required. The first language models have been trained to estimate the conditional probability in eq. (2.1). They were usually fed with the left-context, obtained by cutting a given sentence $s = (w_1, \ldots, w_n)$, and asked to predict sequentially the next words. The Recurrent Neural Network (RNN) architecture has been first widely used as a neural language model trained to iteratively produce the next token given a left-context [Medsker, 2001; Mikolov, 2010]. The most widely used variant of such architecture is the Long Short-Term Memory (LSTM), due to its ability to avoid vanishing gradients during backpropagation [Hochreiter, 1997; Pascanu, 2012]. Other variants of the LSTM architecture have been proposed since [Chung, 2014; Dey, 2017]. The interesting properties of LSTMs made them a candidate of choice for language modeling [Sundermeyer, 2012; Soutner, 2013], and for other linguistic tasks [Wang, 2016b; Chen, 2017]. While language models traditionally learn to predict a word from the left context, LSTMs have been further enhanced to operate

[8] For example, online resources such as Wikipedia have been largely used to train language models as large corpora of well-formed sentences for a variety languages

bidirectionally, in order to produce contextual representations of words [Melamud, 2016; Peters, 2018a]. Concurrently, the Convolutional Neural Networks (CNN) architecture [Lecun, 1995; OShea, 2015] has also been used for language modeling [Dauphin, 2016] and downstream applications [Kalchbrenner, 2014]. As CNNs were mostly successful in image processing applications [Krizhevsky, 2012; Dong, 2014; Simonyan, 2015], multimodal applications have also been proposed to integrate information from both images and text using hybrid models, where CNNs were used to extract visual semantic features from images [Kiros, 2014; Lazaridou, 2015]. In more recent years, the Transformer architecture became popular in NLP due to Vaswani's famous paper[9] [Vaswani, 2017], and gave rise to transformer-based language models. Bidirectional versions of such architecture are trained to predict masked tokens given the entire (left and right) context [Devlin, 2018; Liu, 2019c], yielding contextual word representations which integrate information about the full seentence. However, such architectures cannot directly produce sentence probabilities due to their bidirectional nature.[10] Another equally famous family of transformer neural language models is autoregressive, and learns the probability of a given word given its left context, as in traditional language models [Radford, 2018]. Transformer language models have become ubiquitous in NLP pipelines, due to their robustness and versatility in a wide array of tasks, which we discuss in the next section.

2.1.5 Wait... Can you remind me why we model languages?

Arguably, language modeling has two main purposes. The first is practical, as such models can be deployed on a wide range of applications which involve processing texts. The second is theoretical, as such models are information processing systems capable of handling aspects of language.

Downstream application Language models are typically deployed in downstream application, which makes them very practical as they can be used in a wide array of contexts. Even a simple *n-gram* language model can be used in applications such as text categorization [Cavnar, 1994] and machine translation [Doddington, 2002; Mariño, 2006]. The emergence of pre-trained neural language models broadened this spectrum of applications. These models learn vector representations for the

[9] This paper reached more than 51K citations at the time this thesis was written.

[10] While sentence probabilities cannot be estimated from a bidirectional model, [Lau, 2020] proposed a bidirectional formulation which cannot be directly compared to true sentence probabilities, but is assumed to reflect the model's confidence over a given sentence's likelihood. Such estimate can be used as an input to prediction tasks or correlation measurements, see [Lappin, 2021] for a discussion.

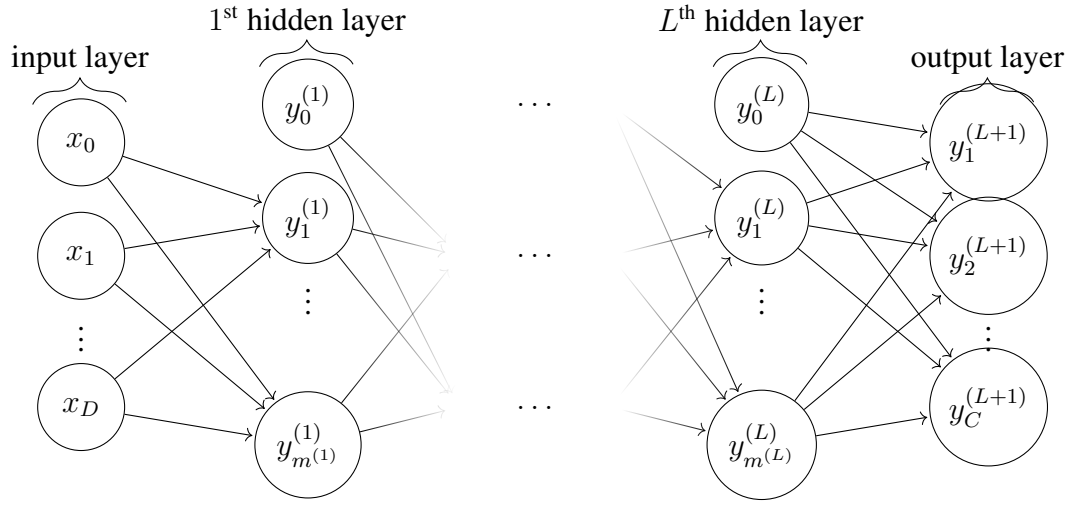


Figure 2.1: Network graph of a $(L+1)$-layer perceptron with D input units and C output units. The l^{th} hidden layer contains $m^{(l)}$ hidden units.

input sentences they process, such that they can be used as general-purpose sentence representations which can be used in any task that involves processing texts: sentiment analysis, sentence entailment, question answering, paraphrase, named entity recognition, and machine translation are examples. This has led to the appearance of several benchmarks aimed at assessing the robustness and versatility of pre-trained language models on a variety of tasks [Wang, 2018; Wang, 2019; Rajpurkar, 2016; Conneau, 2018a].

Investigating theoretical questions The second main purpose of language modeling is to help computational linguists understand how language can be processed, or it can disprove hypotheses about necessary properties of the cognitive structures supporting linguistic knowledge. Let us assume that an artificial system is capable of modeling accurately certain aspects of language. As we have perfect knowledge of the architecture supporting the computations which produce such capacities, we can investigate questions that are central to the cognitive modeling of language processing. For example, we can understand better whether certain inductive biases are necessary to process hierarchical structure [Yao, 2021; McCoy, 2020], or whether these models are capable to handle unseen combinations after being exposed to their structure and primitives separately [Loula, 2018; Lake, 2018]. We will discuss this more extensively in the next chapter.

2.2 Transformer-Based Neural Language Models

2.2.1 Network's Structure

As previously seen, neural architectures result from stacking layers – each computing a transformation of intermediate representations – on top of each other. Transformer-based architectures are one subtype of such models, which typically result from stacking transformer layers, characterized by the self-attention mechanism [Vaswani, 2017]. Contrarily to their predecessors, such architectures do not contain any recurrent connection – each layer transmits information to the next. Each transformer block itself can be decomposed into several layers, and is characterized by the presence of an attention layer. In bi-directional transformer models, a sequence of n tokens $(t_1, \ldots, t_n)$ is transformed into a sequence of intermediate representations $(\mathbf{r}_1^{(l)}, \ldots, \mathbf{r}_n^{(l)}) \in E^l = \mathbb{R}^{d_{E^{(l)}}}$ at each layer $1 \leq l \leq n_{layers}$. This can be written in matrix form $R^{(l)}$, where $R^{(l)}_{i,:} = \mathbf{r}_i^{(l)}$.

At the input level, tokens are transformed into input vector representations using a token-type embedding matrix (to each token corresponds one unique type embedding). This results in a sequence of type embeddings $(\mathbf{e}_{t_1}, \ldots, \mathbf{e}_{t_n})$.

As position information also needs to be injected into the neural network, some authors [Devlin, 2019a] chose to also use absolute position embeddings (APE) at the input level (to each absolute position corresponds one unique position embedding), which is summed to each token's type embedding.[11] Given the sequence of n tokens $(t_1, \ldots, t_n)$, each vector in the sequence of absolute position embeddings $(\mathbf{p}_1, ...\mathbf{p}_n)$ is just summed to the corresponding word-type embedding in $(\mathbf{e}_{t_1}, \ldots, \mathbf{e}_{t_n})$ for architectures using APE.

Input vectors can thus be denoted $(\mathbf{r}_1^{(0)}, ..., \mathbf{r}_n^{(0)})$, such that:

$$\forall 1 \leq i \leq n,\ \mathbf{r}_i^{(0)} = \mathbf{e}_{t_i} + \mathbf{p}_i$$

Denoting transformer blocks $(f^{(1)}, ..., f^{(n_{layers})})$, each set of intermediate representations $(\mathbf{r}_1^{(l)}, \ldots, \mathbf{r}_n^{(l)}) \in E^l = \mathbb{R}^{d_{E^{(l)}}}$, noted in matrix form $R^{(l)}$ is such that:

$$R^{(l)} = f^{(l)} \circ f^{(l-1)} \cdots \circ f^{(1)}(R^{(0)})$$

Autoregressive and bidirectional transformer models In the network structure introduced above, to each token corresponds an intermediate vector. As we have just seen, intermediate vectors at a given layer result from a transformation of the

[11] Note that there has been a plethora of methods to inject position information in transformer neural models [Press, 2021; He, 2020a; Su, 2021; Chang, 2021; Chen, 2021a; Chen, 2021b], which we will discuss in chapter 6.

previous layer's vectors, which integrates information about the context. There exist two families of transformer models which differ in the context which is processed to produce an output probability: autoregressive[12] and bidirectional models[13].

Simply put, these two types of models differ in the output probability distribution that they compute. In *autoregressive* models, the output probability vector at a given position depends only on its left context, $\forall s = (t_1, ...t_{|s|}) \in \mathcal{W}^*, \forall 1 \leq i \leq |s|$ the model estimates $p(t_i|(t_k)_{k<i}$. In *bidirectional* models in turn, the output probability vector at a given position depends on the whole (left and right) context, $\forall s = (t_1, ...t_{|s|}) \in \mathcal{W}^*, \forall 1 \leq i \leq |s|$, the model estimates $p(t_i|(t_k)_{k\neq i}$. This has certain implications as the chain-rule presented in section 2.1 can be applied to the output probability of autoregressive models, but in principle **cannot** be applied bidirectional models.[14]

In both model types, there are as many intermediate vectors as there are input tokens. The difference above however, has implications on the context from which information is accessible at each position. Autoregressive models process their input from left to right as they are unidirectional, and as shown in fig. 2.2. $\mathbf{r}_i^{(l)}$, the i-th token's representation at layer l only depends on vector representations at preceding positions in the previous layer:

$$\mathbf{r}_i^{(l)} = f^{(l)}(\mathbf{r}_1^{(l-1)}, \ldots, \mathbf{r}_{i-1}^{(l-1)})$$

In bidirectional models however, as shown in fig. 2.3, $\mathbf{r}_i^{(l)}$ depends on vector representations at all positions in the previous layer:

$$\mathbf{r}_i^{(l)} = f^{(l)}(\mathbf{r}_1^{(l-1)}, \ldots, \mathbf{r}_n^{(l-1)})$$

A natural question emerges: why is it appealing to train bidirectional models, if we lose the chain rule, essential to computing sentence probabilities? The answer simply lies in the purpose that such models have. It matters little whether they can compute the whole sentence probability as their main usage is to be general-purpose language models, which are fine-tuned on downstream tasks pre-training on the masked language modeling objective. Hence, all that matters is *capturing as*

[12]These models are also called *causal* language models. Throughout this thesis, we will refer to them as autoregressive.

[13]Bidirectional models in this thesis refer to *masked* language models. In these models, some tokens are masked in the sentence, while the rest of tokens is visible in each layer.

[14]Note however, that [Salazar, 2020] proposed to treat probabilities conditioned on bidirectional context as if they were conditioned on the left context for bidirectional models, and introduce sentence pseudo-perplexities. However, and to our knowledge, this quantity is not guaranteed to capture properties of sentence probability estimates.

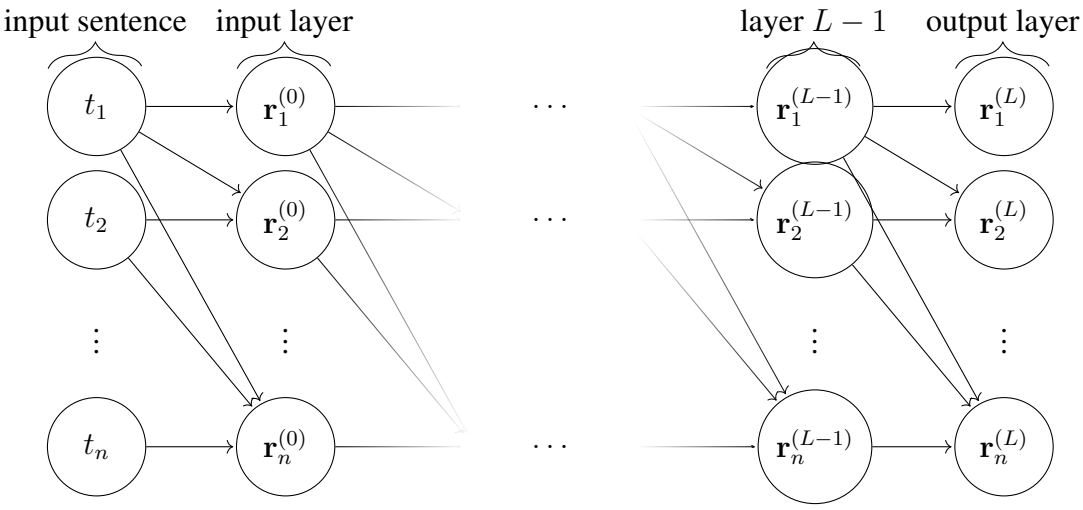


Figure 2.2: Network graph of a (L)-layer autoregressive model with n input tokens. Each nodes represents a the intermediate representation at a given position, which is a vector instead of a single neuron. Arrows represent non-zero attention weights, as the self-attention is the only layer where information is passed across tokens.

much information as possible from the input. While they are not suited to compute sentence probabilities, it is reasonable to think that their predictions still require capturing a vast amount of linguistic knowledge.

A number of robust bidirectional models have been proposed since the emergence of transformer-based architectures [Devlin, 2019b; Liu, 2019c; He, 2020b], but there are also certain widely used transformer models which are autoregressive [Radford, 2018; Yang, 2019; Dai, 2019].

2.2.2 The Transformer Layer and the Attention Mechanism

As mentioned earlier, transformer-based architectures are composed of transformer layers, which are characterized by an attention mechanism. The latter became popular after being proposed in encoder-decoder architectures applied to Neural Machine Translation (NMT) [Bahdanau, 2014; Vaswani, 2017]. The general idea behind the attention mechanism in transformer is to compute vectors at a given layer as a weighted sum of previously available latent representations. Attention weights can thus intuitively be seen as reflecting the relative contribution for each of the weighted vectors. Originally, in encoder-decoder architectures used for NMT, a contextual vector was generated for each position of the input sequence by an encoder, and the attention mechanism was used to weight the relative contributions for each of these vectors, taken as input to a decoder [Bahdanau, 2014].

[Vaswani, 2017] further proposed the scaled-dot product attention mechanism which became popular in widely used in subsequent transformer-based models.

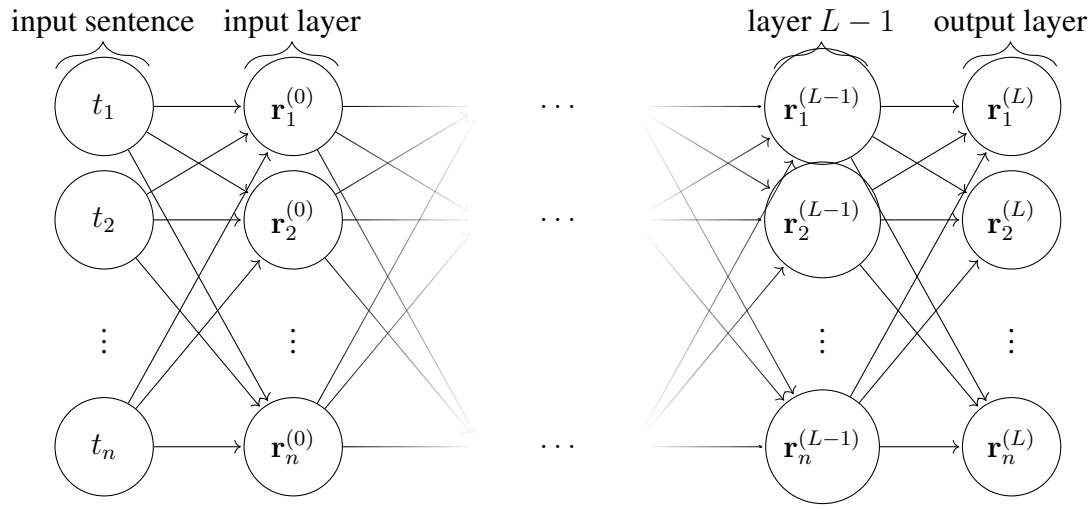


Figure 2.3: Network graph of a (L)-layer autoregressive model with n input tokens. Each nodes represents a the intermediate representation at a given position, which is a vector instead of a single neuron. Arrows represent non-zero attention weights, as the self-attention is the only layer where information is passed across tokens.

For each vector $\mathbf{r}_i$ representing the i-th position token, three different affine transformation are applied:

$$\mathbf{q}_i^{(l)} = W_Q^{(l)}\mathbf{r}_i + \mathbf{b}_Q^{(l)}$$

$$\mathbf{k}_i^{(l)} = W_K^{(l)}\mathbf{r}_i + \mathbf{b}_K^{(l)}$$

$$\mathbf{v}_i^{(l)} = W_V^{(l)}\mathbf{r}_i + \mathbf{b}_V^{(l)}$$

This can be written in matrix form, $\mathbf{r}_i^{(l)}, \mathbf{q}_i^{(l)}, \mathbf{k}_i^{(l)}, \mathbf{v}_i^{(l)}$ respectively being the i-th rows of matrices $R^l, Q^{(l)}, K^{(l)}, V^{(l)}$ (omitting biases):

$$Q^{(l)} = R^{(l)}W_Q^{(l)\top}$$

$$K^{(l)} = R^{(l)}W_K^{(l)\top}$$

$$V^{(l)} = R^{(l)}W_V^{(l)\top}$$

An attention-weight matrix is further computed as follows:

$$A^{(l)} = Q^{(l)}K^{(l)\top}$$

Each of A's rows are normalized using layer normalization [Ba, 2016], which results in a normalized matrix $\tilde{A}^{(l)}$. Values (as rows of V) are weighted using the previously computed attention weight matrix:

$$R^{(l+1)} = \tilde{A}^{(l)}V^{(l)}$$

Attention heads' outputs are then merged by summing resulting matrices for each attention head, after applying an affine transform specific to each head's output. An affine transform is finally performed and an activation function is applied, typically Gaussian Activation Linear Unit [Hendrycks, 2016].

The difference between autoregressive and bidirectional models mentioned in the previous section is also mirrored in the computation of attention, in which attention is directed either to preceding tokens only or to all tokens, as displayed in section 2.2.2.

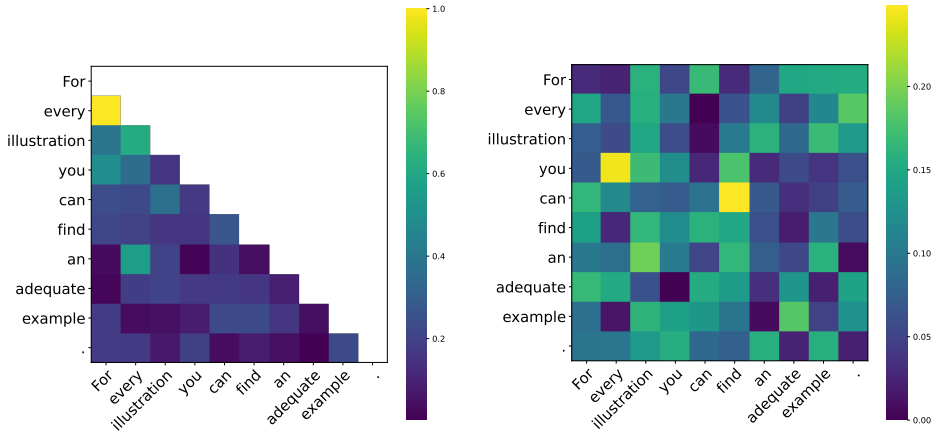


Figure 2.4: Attention matrix examples for an autoregressive model (left) and a bidirectional model (right)

2.2.3 Previous Architectures

Previously to transformer-based language models, the preferred neural language models were recurrent. The main difference between transformer-based neural networks and recurrent neural networks is that in recurrent NNs, the same series of transformations is applied to hidden states (the output loops back into the layer's input), while in transformer architectures, a series of different transformer blocks which learn different weights are stacked on top of each other and each applied sequentially to the output of the previous layer [Medsker, 2001; Mikolov, 2010]. Variants of this architecture include the Long Short-Term Memory (LSTM) [Hochreiter, 1997; Pascanu, 2012], and the Gated Recurrent Unit (GRU) [Chung, 2014; Dey, 2017].

An example of LSTM layer is displayed in fig. 2.5. Since the appearance of transformers, some authors proposed to make low-level comparisons between the computational capacities of recurrent and transformer-based neural networks [Katharopoulos, 2020; Bhattamishra, 2020], a theoretical question that had been

largely investigated for previous architectures and neural networks [Siegelmann, 1995; Sperduti, 1997; Korsky, 2019]. A number of studies also compared these architectures' ability to handle certain tasks [Lakew, 2018; Shim, 2022; Delétang, 2022]. While the cross-architecture comparison is certainly a fruitful and necessary body of research, we do not focus our efforts on such quest. We still formulate all our questions in the most general possible form, so that the reasoning and methodologies can be applied indifferently to neural language models other than those under consideration in this thesis.

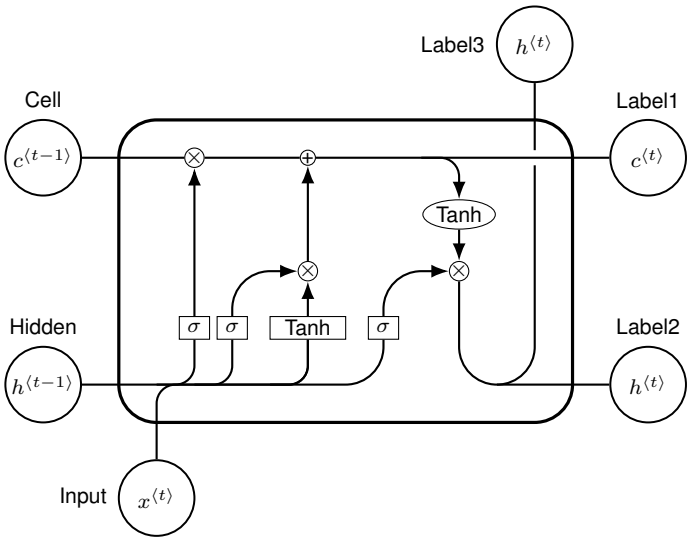


Figure 2.5: A Long-Short Term Memory (LSTM) layer.

2.2.4 The Alleged Abilities of Transformer-based NLMs

Various Transformer-based NLMs have been shown to perform well on several *downstream* tasks such as a Natural Language Inference (NLI), paraphrase, sentence entailment, or question answering [Wang, 2018; Wang, 2019; Rajpurkar, 2016; Conneau, 2018a]. A wealth of tasks which neural language models can be finetuned on have been proposed, the variety of which can be seen in table. 2.2. As discussed previously, one purpose of pre-trained models is to produce general-purpose quality representations, such that they can be deployed on downstream applications after being fine-tuned on limited data. As an example of their performance we display BERT's results on the GLUE benchmark in table. 2.1.

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
BiLSTM+ELMo+ Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	86.7/85.9	72.1	92.7	94.9	60.5	86.5	89.3	70.1	82.1

Table 2.1: GLUE Test performance for BERT and one of its most robust predecessors. The nature of these tasks is given in table. 2.2. These figures are taken from the original BERT paper [Devlin, 2019b].

Limitations Considering these abilities, one natural question to ask is whether the models truly generalize on these tasks, and if so, how they come to achieve such performance. As these tasks are high-level linguistic tasks, if models generalize on them, they should capture a vast amount of lower-level linguistic knowledge. While a number of benchmarks point towards the capture of linguistic knowledge by modern NLMs, the validity of claims like these can be problematic. Indeed, if a high-level downstream task is solved, this could be due to the task or dataset being too easy, which says little about the capacities of the model itself.

A number of factors could explain both successes or failures of a model on a downstream task. Usually, performance on a benchmark is usually interpreted in light of the following two factors:

- The difficulty of the task, and the necessity of acquiring lower-level linguistic abilities to solve it. For instances, tasks such as inferring whether a proposition is entailed by a premise, or summarization are considered to be hard task requiring deep knowledge of syntactic and semantic structure.

- The quality of the model's pre-training, which itself integrates many subfactors such as the capacities that are learnable by the model's architecture, but also the quality of the pre-training data, or the optimization algorithm, *inter alia*.

However, the difficulty of the dataset could play a substantial role as the latter could fail to represent the difficulty of the task in its more general form. This makes it possible to learn simple heuristics to perform well on specific dataset. For instance, [McCoy, 2019] show that correct answers can be largely guessed on the MNLI dataset [Williams, 2018]using simple heuristics such as lexical overlap between the premise and the hypothesis. [Lai, 2014] show in turn that a model relying only on the presence of negation on the SICK dataset [Marelli, 2014] can achieve great performance when detecting contradictions.

Under these circumstances, it is difficult to disentangle those factors when only observing the surface performance of a NLM. It is therefore hard to conclude on the model's capture of linguistic abilities. For instance, one cannot *reasonably*

Task family	Task	Dataset	Paper
Classif.	Acceptability	GLUE - CoLa	[Wang, 2018]
	Sentiment Analysis	GLUE - SST2	[Wang, 2018]
		DecaNLP - SST	[McCann, 2018]
	Paraphrase	GLUE - MRPC	[Wang, 2018]
		GLUE - QQP	[Wang, 2018]
		XTREME - PAWS-X	[Hu, 2020]
	NLI	GLUE - MNLI	[Wang, 2018]
		DecaNLP - MNLI	[McCann, 2018]
		XTREME - XNLI	[Hu, 2020]
		GLUE - QNLI	[Wang, 2018]
		GLUE / SuperGLUE - RTE	[Wang, 2018; Wang, 2019]
		GLUE - WNLI	[Wang, 2018]
		SuperGLUE - CB	[Wang, 2019]
		SICK - Entailment	[Marelli, 2014]
	Question Answering	SuperGLUE - BoolQ	[Wang, 2019]
		SuperGLUE - COPA	[Wang, 2019]
		SuperGLUE - MultiRC	[Wang, 2019]
	Coreference	SuperGLUE - WSC	[Wang, 2019]
	Goal-Oriented Dialogue	DecaNLP - WOZ	[McCann, 2018]
	Word Sense Disambiguation	SuperGLUE - WiC	[Wang, 2019]
		CoSimLex	[Armendariz, 2020]
		Usim-2	[Erk, 2013]
		DWUG	[Schlechtweg, 2021]
	NP Enrichment	TNE	[Elazar, 2022]
Seq2seq	Neural Machine Translation	DecaNLP - IWSLT	[McCann, 2018]
	Summarization	DecaNLP - CNN/DM	[McCann, 2018]
	Semantic Parsing	DecaNLP - WikiSQL	[McCann, 2018]
Text Retrieval	Parallel Sentence Extraction	XTREME - BUCC	[Hu, 2020]
	Sentence Alignment	XTREME - Tatoeba	[Hu, 2020]
Span Extraction	Coreference	SuperGLUE - WSC	[Wang, 2018]
	Pronoun Resolution	DecaNLP - MWSC	[McCann, 2018]
	Semantic Role Labeling	DecaNLP - QA/SRL	[McCann, 2018]
	Keyphrase Extraction	KP20K	[Meng, 2017]
		Inspec	[Hulth, 2003]
		SemEval 2010	[Kim, 2010]
		SemEval 2017	[Augenstein, 2017]
	Question Answering	DecaNLP - SQuAD	[McCann, 2018]
		XTREME - XQuAD	[Hu, 2020]
		XTREME - MLQA	[Hu, 2020]
		XTREME - TyDiQA-GoldP	[Hu, 2020]
		SuperGLUE - ReCoRD	[Wang, 2019]
Text Regression	Sentence Similarity	GLUE - STS-B	[Wang, 2018]
		SICK - Relatedness	[Marelli, 2014]
Sequence labelling	Dialogue Labelling	Switchboard	[Godfrey, 1992]
		MapTask	[Thompson, 1993]
	Metaphor Detection	VU Amsterdam Dataset	[Leong, 2020]
	POS-tagging	XTREME - POS	[Hu, 2020]
	Named Entity Recognition	XTREME - NER	[Hu, 2020]

Table 2.2: A taxonomy of downstream NLP applications that have been used to benchmark neural language models.

claim that model M has acquired abstractions about the compositional structure of sentences because it performs well on a dataset D that represents a natural language

inference task Y, nor can one be reasonably certain about the opposite if the model performs poorly.

This prompts the need to craft probing methodologies, designed to understand better the capture of targeted linguistic abilities by such models. If evidence is found for the absence of lower-level linguistic abilities necessary for higher-level tasks, then this should hold as additional proof that the model isn't truly generalizing on that higher-level task, regardless of its performance after fine-tuning time. In the next section, we review the literature which attempted gaining understanding of the model's capture of finer-grained linguistic knowledge.

2.3 Revealing the Linguistic Knowledge of Transformer-Based NLMs: Methodologies

In search for Linguistic Knowledge While there is no consensus in the literature on how to search for linguistic knowledge in neural language models, broadly speaking the task consists in determining whether the neural model captures linguistic abilities. Achieving this goal is a challenge, as modern neural language models are supported by increasing large architectures which consist in hundreds of millions, if not billions of parameters (as displayed in fig. 1.1). When searching for linguistic knowledge, one first needs to target a specific linguistic property or phenomenon of interest before assessing its capture by a neural language model. This first choice is not easy given the variety of formal descriptions of linguistic structure found in different grammars, and the plethora of linguistic phenomena which have been described by linguists – **what** linguistic knowledge should I look for in my neural language model? Once a linguistic property is chosen, a second challenge consists in picking a location in the neural model to seek knowledge – **where** should we look for evidence of linguistic knowledge, in the model's intermediate representations, in the layers' transformations applied to representations, or in the model's output? Finally, linguistic properties are often defined as discrete functions, while neural language models process representations and output scores which lie in continuous spaces. This raises a third difficulty in seeking linguistic knowledge, once one answered the **what** and **where** questions – **how** do I expect linguistic knowledge to be captured by my model, i.e. what holds for evidence that the model acquired linguistic knowledge?

The broad spectrum of answers which can be reasonably given to such questions inspired a variety of methodologies – referred to as **probing methods** – which are reviewed in this section. In the following, we taxonomize such methodologies as they each have their own stance in looking for linguistic knowledge, that is distinct views on some of the questions listed above.

2.3.1 Linguistic knowledge in intermediate representations

The stance of researchers seeking linguistic knowledge in intermediate representations can be phrased as follows: *A model that captures linguistic knowledge is able to represent linguistic properties in its layers.*

Structural probes: finding linguistic structure in space geometry

A first category of such methods evaluates whether the geometry of intermediate vectors mirrors the linguistic structure of their corresponding input sentences. These methods typically make a hypothesis regarding how syntactic structure can be mapped into intermediate representations, and approximate vectors of that representing space. They are thus aimed at unveiling how syntactic representations are mirrored in the structure of the probed representing space [Hewitt, 2019b; McCoy, 2018; White, 2021b].

In order to seek linguistic structure in intermediate representations, one needs to choose a linguistic structure over inputs. Indeed, there are multiple ways to represent grammatical structure of sentence in a given language. Models of syntax include Transformational Grammar [Chomsky, 1965a], Dependency Grammar [Arrivé, 1969], Lexical-Functional Grammar [Kaplan, 2004], and Combinatory Categorial Grammar [Steedman, 2000]. The aforementioned methods typically use the dependency parse tree of input sentences, and map positions in the parse tree into the set of vectors at each position in the sentence.

Once a syntactic model is chosen, one needs to choose a representation of positions for each word in the sentence, in order to map structure into representing spaces. Given a sentence $s = (t_1, ..., t_n)$ and its dependency tree T^s, [Hewitt, 2019b] tests whether the distance between any word pair (t_i, t_j) in the dependency tree, denoted $d_{T^s}(t_i, t_j)$, and the parse tree-depth of any token, denoted $||t_i||$ are represented in the intermediate space. They assume that such structural features are represented linearly in the intermediate space, i.e. they test whether these distances can be retrieved by transforming representing vectors linearly by minimizing the reconstruction distance using linear projection matrices $B \in \mathbb{R}^{d_E,k}$, defining a family of squared distances between linear transforms of intermediate representations:

$$d_B(\mathbf{r}_i, \mathbf{r}_j)^2 = (B(\mathbf{r}_i - \mathbf{r}_j))^\top (B(\mathbf{r}_i - \mathbf{r}_j)),\ B \in \mathbb{R}^{d_E,k}$$

Finding the best approximation is equivalent to minimizing the following objective:

$$min_B \sum_s \frac{1}{|s|^2} \sum_{i,j} |d_{T^s}(t_i, t_j) - d_B(\mathbf{r}_i, \mathbf{r}_j)^2|$$

[White, 2021b] instead test non-linear distance metrics in intermediate representing vectors to retrieve the same distances between word-pairs and tree-depths. To do so, they build on positive semi-definite kernels which can be used to define a non-linear distance, using a few non-linear kernels.

[McCoy, 2018] in turn assumes that intermediate vectors approximate struc-

ture using tensor products, using filler-role decompositions of syntactic structure [Smolensky, 1990]. In such, roles are generic representations of position in a structure, and fillers are the elements placed at such positions.

Limitations These methodologies are not so abundant in the literature and require strong priors regarding the structure of the input space. Besides, the reconstruction error metrics are not directly interpretable and only allow for comparison between different reconstructions, without even knowing whether such approximations make sense to the model. Additionally, it is not clear whether the obtained mappings from transformer vectors to dependency trees are unique, or whether mappings to alternative syntactic structures can be equally learned using a suitably trained structural probe. This raises the need for extrinsic evaluation, which is difficult in practice as intermediate representations are embedded using a non-bijective transformation in the case of [Hewitt, 2019b; White, 2021b].

Diagnostic classifiers - extracting information from intermediate representations

Principle This approach also seeks linguistic knowledge in intermediate representations. It assumes that such knowledge can be retrieved by training supervised models on top of fixed pre-trained representations [Adi, 2017; Conneau, 2018b; Hall Maudslay, 2020] – these models are sometimes referred to as *diagnostic classifiers* in the literature [Giulianelli, 2018]. This type of study is precisely motivated by the fact that performance on downstream tasks is hard to interpret as is, as we cannot know how neural models make correct decisions on such tasks – that is what information they rely on when correctly guessing the attribute. The general assumption behind this methodology is that, if a probing classifier trained to predict a property of interest based on a model's intermediate representations achieves high accuracy, then that property is encoded in the representations.
Given our notations introduced in section 3.1, a linguistic property can be denoted:

$$\mathcal{T} : (s, \bar{s}) \in \mathcal{W}^* \times \mathcal{P}(s) \rightarrow \mathcal{T}(s) \in \mathcal{V}^T \cup \{\text{undefined}\}$$

Recall that for a sentence $s = (t_1, \ldots, t_n)$, and given our notations in section 2.2, intermediate representations at layer l in a transformer-based model form a sequence of n vectors $(\mathbf{r}_1^{(l)}, \ldots, \mathbf{r}_n^{(l)})$. Thus, to each part of the sentence $\bar{s} \in \mathcal{P}(s)$ corresponds a subset of these intermediate representations (which in the case of single tokens is simply the representation of that vector). We denote such representations $\mathcal{R}_{\bar{s}}^{(l)} \subset (\mathbf{r}_1^{(l)}, \ldots, \mathbf{r}_n^{(l)})$.

Definition 2.3.1. Given a property $\mathcal{T}$ and a set of representations $\mathcal{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_n)$, diagnostic probing is a learning task:
$\forall s, \bar{s}$ such that $\mathcal{T}(s, \bar{s})$ is defined, predict $\mathcal{T}(s, \bar{s})$ based on $\mathcal{R}_{\bar{s}}$.

In table. 2.3, we display examples of probing tasks along with their scope among other features. For years, it has been widely accepted that if the probe reaches good performance in that prediction task, it means that the property represented by $\mathcal{T}$ is encoded the probed intermediate representations.

Scope	Type	Name	# Outputs	Description	Paper
Sentence	Surface	SentLen	6	The length of the input sentence, binned.	[Conneau, 2018b] [Adi, 2017]
		WC	1000	Whether the sentence contains a specific word.	[Conneau, 2018b] [Adi, 2017]
	Syntactic	BShift	2	Whether two words have been shifted in the sentence.	[Conneau, 2018b]
		TreeDepth	8	The depth of the constituency parse-tree.	[Conneau, 2018b] [Perone, 2018]
		TopConst	19	The sequence of constituent children immediately below the root of the constituency tree.	[Conneau, 2018b]
	Semantic	Tense	3	The tense of the verb in the main clause.	[Conneau, 2018b]
		SubjNum	2	Number for the subject of the main clause.	[Conneau, 2018b]
		ObjNum	2	Number for the direct object of the main clause.	[Conneau, 2018b]
		SOMO	2	Whether a verb or noun in the sentence has been replaced with another.	[Conneau, 2018b]
		CoordInv	2	Whether two clauses have been inverted in the sentence.	[Conneau, 2018b]
		Semantic Role	2	Whether a predicate contains a given word with a given role.	[Ettinger, 2016]
		Negation Scope	2	Whether a certain scope is negated in the sentence.	[Ettinger, 2016]
Span	Syntactic	Constit.	30	The constituency label for a given span.	[Tenney, 2019b]
	Semantic	Named Ent.	18	Predict the superset of a span-level, using a given taxonomy.	[Tenney, 2019b]
Pair	Surface	Word Order	2	The relative order of two words in the sentence.	[Adi, 2017]
	Syntactic	Depend.	49	The dependency arc label for a given word-pair.	[Tenney, 2019b] [Blevins, 2018]
		Coref	2	Whether two spans corefer.	[Tenney, 2019b]
	Semantic	SRL	66	The semantic role of a span relative to a predicate	[Tenney, 2019b]
		SPR	18	Semantic proto-roles, fine-grained semantic attributes of a span relative to a predicate	[Tenney, 2019b]
		Rel.	19	Real-world semantic relation between entities	[Tenney, 2019b]
Word	Syntactic	POS	48	The POS tag of a word.	[Blevins, 2018] [Belinkov, 2017b] [Belinkov, 2017a] [Peters, 2018a] [Peters, 2018b]
		Parent	48	The POS tag of a word's parent.	[Blevins, 2018]
	Semantic	Animacy	2	Whether a noun refers to an animate or inanimate subject.	[Klafka, 2020]
		Dyn-Stat.	2	Whether a verb is dynamic or stative.	[Klafka, 2020]
		CIA	2	The presence of a causative-inchoative alternation.	[Klafka, 2020]
		SEM	-	Semantic labels for certain words in the sentence.	[Belinkov, 2017b]
	Morpho-syntactic	Number	2	The number predicted exclusively on nouns, or verbs.	[Klafka, 2020]
		Tense	2	The tense of a verb.	[Klafka, 2020]
		Gender	2	The gender of a given noun.	[Klafka, 2020]

Table 2.3: Examples of diagnostic probing tasks found in the literature.

How informative is my probing classifier's performance? [Conneau, 2018b] have found that downstream task performance often did not correlate, or correlated negatively with probing classifiers' performance. In their analysis though, they use a multi-layer perceptron (MLP) with non-linearities to extract information. Though earlier probing papers principally made use of linear classifiers [Shi, 2016; Ettinger, 2016; Alain, 2016], this tendency decreased over years [Belinkov, 2017a; Liu, 2019a; Ettinger, 2018]. Some authors however suggested that a less complex classifier gives us more insight into the model. Others, however, called this criterion into question [Tenney, 2019b; Tenney, 2019a; Voita, 2020; Papadimitriou, 2021; Sinha, 2021c; Pimentel, 2020a; Pimentel, 2021]. Notably, [Hewitt, 2019a] proposed that complex classifiers may learn to extract a property by themselves, and may thus not reflect any true pattern in the representations. They show for instance that a probe can sometimes even learn random labels, prompting the need for better criteria to design probes, in particular, controlling for their complexity. [Zhang, 2018] also tested the ability of probes to learn random labels, and in turn show that varying the amount of data used to train a probe is a good approach to determine the quality of the encoding for a given linguistic property. In doing so, they share the intuition that the probing classifier should not overfit the probing task to detect linguistic knowledge readily available in representations.

Diagnostic probing viewed under the information-theoretic lens Further, [Pimentel, 2020b] showed that, under a weak assumption, contextual representations encode as much information as the original sentences. This shows an obvious limitation of considering probing only as extracting mutual information. Other researchers in turn take complexity of the probe as an important criterion to take into account when probing a neural language model. [Voita, 2020] in turn uses minimum description length (MDL) as an information-theoretic characterization for simplicity of a probe.

Limitations The first attempts at uncovering what abstractions are captured in a model's representations have been flawed, by their purely correlational nature. For instance, the fact that a property can be predicted from intermediate representations depends on the dataset used for the study, in which other properties could just correlate with the property of interest without the model really capturing it broadly speaking. Further, when obtaining high performance by extracting information from intermediate layers, one concludes that the property is encoded without ever knowing if the model ever makes use of this encoding, that is the encoding which the probing classifier relies on to extract information.

More recent work investigated this in causal frameworks [Elazar, 2021; Vig, 2020a; Lakretz, 2019; Tucker, 2021; Ravfogel, 2021] for which targeted interventions on a neural architecture result in predictable outcomes at the decision layer. For example, [Vig, 2020b] fix a neuron's value[15] while manipulating the model's input to evaluate this neuron's role in mediating gender bias. Relatedly, [Elazar, 2021] propose a method to erase a target property from a model's intermediate layers. They then analyze the effect of such interventions on a masked language model's outputs. Causal analyses usually provide stronger proofs that a given structure in the model's representing space is meaningful to the model and captures the targeted linguistic property of interest and are a necessity to gain better understanding at what knowledge neural language models possess.

2.3.2 Behavioral tests

The stance of researchers seeking linguistic knowledge by performing behavior tests can be phrased as follows: *A model that captures aspects of linguistic generalization is able to show preference for linguistically acceptable sentences and completions.*

Another probing paradigm analyzes the behavior of pre-trained models on carefully curated datasets. This family of studies only investigates the capture of linguistic knowledge by a model by looking at its predictions using carefully designed datasets such that certain output is expected for certain stimuli. These methods comprise cloze tasks that target the acquisition of specific phenomena such as semantic roles [Ettinger, 2020], or syntactic structure [Goldberg, 2019; Newman, 2021a] by only looking at the model's outputs. By avoiding the use of diagnostic probes, they do not fall prey to the criticism above—tasks are directly performed by the model with no further training, and thus must reflect the pre-trained models' acuity. One notable example is Linzen et al. [Linzen, 2016], who evaluate a language model's syntactic ability via a careful analysis of a number agreement task. By controlling the evaluation, Linzen et al. could disentangle the model's syntactic knowledge from a heuristic based on linear ordering. In a similar vein, a host of recent work makes use of carefully designed test sets to perform behavioral analysis [Ribeiro, 2020; Warstadt, 2020c; Warstadt, 2020a; Lovering, 2021; Newman, 2021a].

Limitations This paradigm typically treats the model itself as a blackbox, thus failing to explain how individual components of the model work, or how the model

[15] A neuron here is just one dimension of an intermediate representing vector.

represents its linguistic knowledge. Therefore, one does not know how the model produces the correct behavior, nor if it produces it for the right reasons. Still, behavioral probing often yields powerful insights, while also making it possible to test the capture of fine-grained linguistic knowledge.

2.3.3 Assessing the function of attention heads

The stance of researchers seeking linguistic knowledge by probing attention heads can be phrased as follows: *A model that captures aspects of linguistic generalization combines its representations in a way that is linguistically meaningful*

As seen in section 2.2, self-attention is a key component of transformer-based neural architectures. This mechanism outputs intermediate representations for each position in the sentence, which result from a learned weighted average of vectors representing each token at the previous layer:

$$\forall s = (t_1, \ldots, t_n) \in \mathcal{W}^*,$$

$$\forall 1 \leq i \leq n, \forall 1 \leq l \leq n_{layers},$$

$$\mathbf{r}^{l+1} = A_{i,:}(W_V r_l + b_V)$$

Thus, every token grabs information from other parts of the sentence. Attention heads can be visualized in order to gain understanding about how each token attends to other parts of the sentence. [Kovaleva, 2019] have proposed a typology of the patterns that appear in self-attention matrices by segmenting the set of 144 attention heads in $BERT_{base}$, and tried to map some of them to specific linguistic functions. They showed that the number of active attention heads in BERT's architecture could be greatly reduced while keeping performance unchanged.

A more systematic methodology to prune attention heads was proposed by [Michel, 2019], in which attention matrices are ranked using an importance metric capturing whether they have an effect on the model's decisions on a given task. As a reminder, given a sequence of n vectors $\mathbf{r}_1, ...\mathbf{r}_n$ representing tokens of the sentence, the self-attention mechanism is computed using each query vector q for the representing vector e_i of each token of the sentence, and for each attention head h using the formula:

$$A_{W_k^h, W_q^h, W_v^h, W_o^{\mathbf{r},q}} = W_o \sum_{i=1}^{n} \alpha_i W_v^h \mathbf{r}_i$$

$$\text{where } \alpha_i = softmax(\frac{q^T W_q^T W_k \mathbf{r}_i}{\sqrt{d}})$$

They proposed the following modified formula for computing self-attention

$$\tilde{A}(x,q) = \sum_{k=1}^{N_h} \varepsilon_h A_{W_k^h, W_q^h, W_v^h, W_o^h}$$

where ε_h are mask variables in $\{0,1\}$. Masking head h is equivalent to setting ε_h to 0 [Michel, 2019].
This modified formula, introducing the binary ε variable, makes it possible to compute a head importance score as follows:

$$I_h = \mathbb{E}_{x\sim X} \left| A_h(x)^T \frac{\partial L(x)}{\partial A_h(x)} \right|$$

Heads with the smallest score are iteratively masked and a performance score is computed at each iteration. When performance decrease reaches a certain threshold, the pruning process is stopped.

Limitations Attention matrices are very hard to interpret in practice, and despite a number of attempts to finding simple patterns at inference time, they have failed so far to give easy access to interpretable linguistic features.

2.3.4 Synthesis

This synthesis of the literature resulted in the following diagnosis:

1. The wealth of methodologies in the literature on probing for the capture of linguistic properties disagree in results, and their epistemic grounds are *a priori* not translatable into each other.

2. In the literature, expressions such as *linguistic abilities*, *linguistic knowledge*, or *knowledge of linguistic structure* refer to a very varied set of measurable characteristics of NLMs which are different in nature and not explicitly related to each other.

3. The epistemic value of some of the evaluation methods is highly questionable. For example, transferring the knowledge of representations learned by BERT *inter alia* makes it possible to solve NLI tasks in benchmarks such as GLUE [Devlin, 2019a; Wang, 2018], but such tasks are too easy as word order does not even matter [Sinha, 2021b]. Accounts for the capture of linguistic information that make use of probing auxiliary classifiers are also largely questionable as correlation is not causation, and we do not know whether the model ever uses the information extracted by the probe. Furthermore it is

possible for an auxiliary classifier to learn random labels [Hewitt, 2019a] as long as its capacity is sufficient.

4. Behavioral tasks show the model is not even able to capture negation, or simple semantic roles relations [Ettinger, 2020]. In this sense, they are informative as they permit us to rule out aspects of linguistic generalization. However, cloze tasks and behavioral probes alone can be limited as they say little about how the model solves the task, or why it fails, by not remaining agnostic on the mechanisms which take place inside the black box.

With this series of observations in mind, we will investigate the output behavior of language models as a direct reflection of its functioning in its training setting. We will then try to understand whether evidence for linguistic knowledge observed at the output level translates into the capture of abstract representations of linguistic properties the model's layers.

In a first series of experiments, we investigate whether the surface behavior of transformer NLMs, measured at the output level, demonstrates the capture of linguistic generalizations. This is achieved by evaluating the model's output behavior in light of linguistic theories, and by comparing it to human's behavior.

Further, we develop a methodology grounded in the causal literature to ensure that the model has captured abstract representations of linguistic properties in its layers, leading it to producing behavior suggesting the model captures linguistic generalizations. In doing so, we bridge the gap between representational and behavioral evidence for linguistic knowledge.

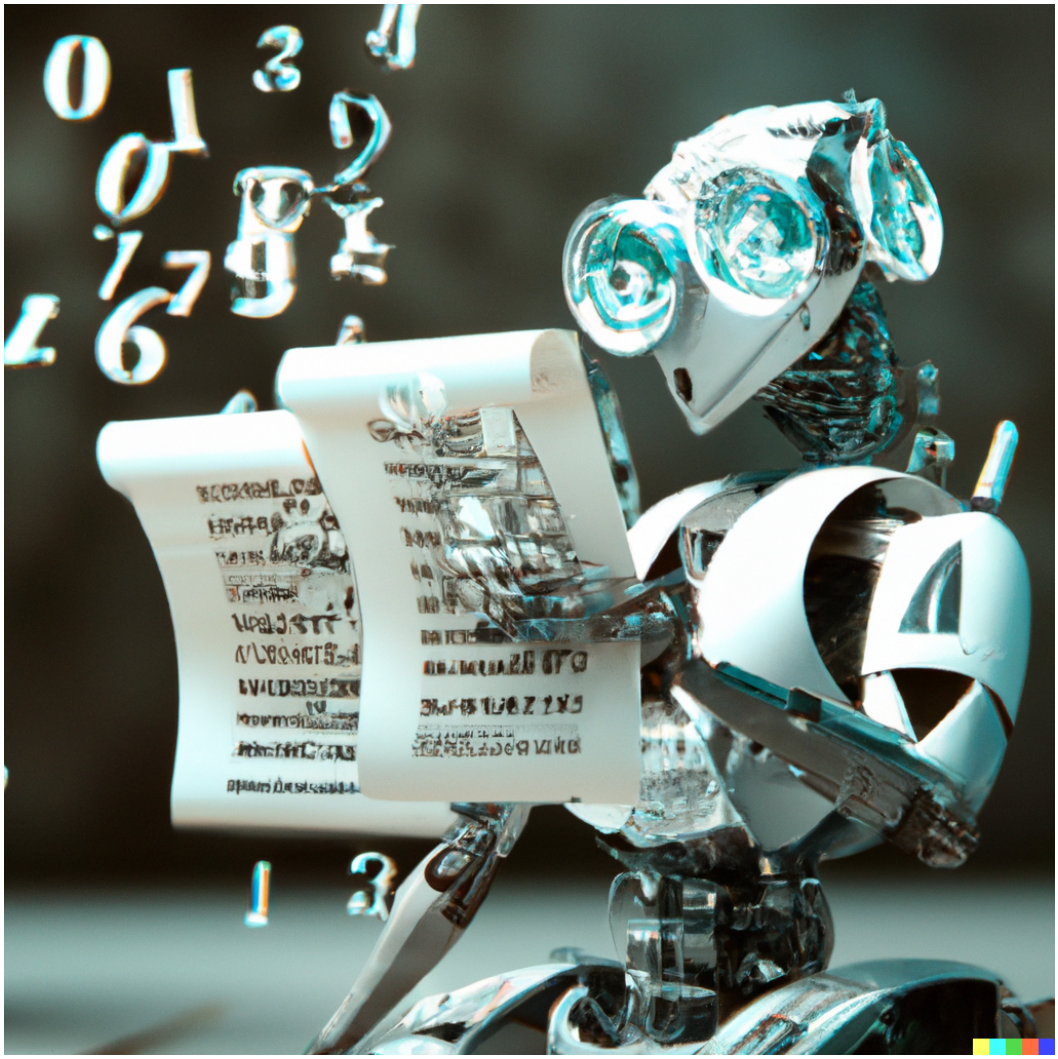


Figure 2.6: "A robot reading text carefully", K.L. x DALL·E 2

Chapter 3

Decomposing Linguistic Generalization: Linguistic Knowledge in Neural Language Models

Goals

Examining the linguistic abilities of neural language models is an ambitious task. This investigation has multiple facets which fail to benefit each other when studied in isolation. The goal of this chapter is to propose a systemic formalization of the investigations aimed at uncovering NLM's linguistic abilities, delimiting the scope of each subset of questions, and formalizing how they relate to each other. As a tentative epistemological unification of the wealth of stances and methods, we propose to map the various subproblems aimed at understanding whether and how neural language models capture linguistic abilities, placing them into different interdependent levels of analysis of the broader question, and giving explicit relations. In doing so, we attempt connecting the different paradigms in evaluating a NLM's linguistic knowledge with each other, reconciling methodologies which at first glance appear to have disjoint epistemic grounds. This formalization is crucial, as it places each future investigation into a comprehensive understanding of the abilities possessed by NLMs.

Contents

3.1 Linguistic Generalization Decomposed

In section 2.1, we have seen that language models are probability distributions over word sequences. Ideally, the learned function should approximate the "true" probability distribution of sentences. In practice, this true probability distribution is not directly accessible but rather a theoretical object. Instead, this theoretical true probability is approximated using large corpora which are considered to be representative *to some extent* of the language one aims to model.

Generally, a number of biases could arise from the choice of the training corpus, such as domain-specificity or frequency effects. Under the assumption that the corpus is representative, we can expect a neural language model to capture general aspects of language.

Definition 3.1.1. In this thesis, **linguistic generalization** refers to the degree to which a language model is able to extrapolate certain linguistic constraints to any sentence in $\mathcal{W}^*$, while only being exposed to a finite, limited training corpus.

We make two remarks on this definition:

- First, it assumes certain desirable linguistic constraints on the model's output probability distribution (of sentences or completions), which depends on a set of hypotheses shaping the output probability.
- Second, this definition is different in nature from that of statistical generalization, as the latter only considers that the model is able to approximate the distribution from which the training data has been sampled. One way to see that this is different from the whole set of sentences, is that in practice the domain from which the training data is sampled does not cover a large spectrum of sentence structures, and lexical combinations.

The coarsest metric that can be used to assess that my NLM is adequately modeling language is that used to assess statistical generalization, i.e. the model's cross-entropy loss on a held out test dataset.[1] The latter writes:

$$L_{\mathcal{D}^{\text{test}}}(p) = \sum_{(x,y)\in\mathcal{D}^{\text{test}}} -\sum_{c=1}^{|\mathcal{W}|} \log y_c p(w_c|x)$$

As the latter aggregates predictions on all sentences, this statistical generalization metric is too coarse for us to interpret targeted aspects of the model's linguistic abilities. Furthermore, the absolute value obtained on a given set is not directly interpretable as it is lower-bounded by the true probability's entropy, which cannot

[1] Or, equivalently, its perplexity.

be known as we do not have access to the language's true probability. Therefore, we cannot know how close we are to that lower bound. This metric can only inform us on whether a model is *relatively* robust, i.e. whether it is more performant than another.

Given our definition above, a natural way to perform targeted evaluation of a NLM's linguistic abilities would be in turn to test its capacity to produce *adequate* probability estimates on subsets of sentences which address specific linguistic phenomena. In this regard, linguistic generalization is assessing whether the model captured such during training. In particular, these phenomena are mirrored by constraints imposed over the probability distribution at the model's output level. Carrying such analysis thus requires having an idea on what should be the shape of the model's output probability distribution for certain input sentences, which can be driven by linguistic hypotheses on their grammaticality or well-formedness. In addition to observing *whether* the model produces output probabilities respecting certain constraints, another fundamental question is *how* it does so. Bringing answers to this questions requires making hypotheses at levels deeper than the surface behavior level, which we will discuss in section 3.3. Further, there are immediate considerations to take into account when addressing constraints at the behavioral level:

- The model outputs probabilities, which are real-valued. Therefore, they cannot capture well-formedness as a binary property of sentences.
- While grammaticality is often viewed as a binary property of sentences, it is largely accepted that acceptability judgements in turn are gradient. This has implications on how to define ground truth or constraints on the model's outputs.

There are ways to alleviate these apparent difficulties which we will discuss in section 3.4.

In the next section, we first take a step back to discuss how linguistic theories and our investigation of the knowledge captured by NLMs can contribute each other.

3.2 On the Relationship between Linguistic Theories and Investigation of Linguistic Abilities in NLMs

In this section, we discuss the relationship that lies between theoretical investigations in linguistics, and the goal pursued in this thesis, that is understanding the

linguistic abilities of NLMs. After reviewing some of the goals in linguistic theorizing which are useful to this thesis, we discuss the mutual contributions that both investigations can bring to each other.

3.2.1 Goals in linguistics

Before understanding how linguitic theories and NLMs can benefit each other, we review some of the objectives pursued by linguistic theorizing. We do not attempt to enumerate exhaustively these objectives and only borrow the portion that is useful to the questions addressed in this thesis.

Describing linguistic structure One of the first purpose of linguistic theories was to describe languages as structured systems, defining categories of linguistic units and their relations to each other. This descriptive objective dates back to at least De Saussure's structuralist approach [Saussure, 1916]. Structuralism proposed two types of relations to describe regularities in languages, paradigmatic and syntagmatic relations. Paradigmatic relations defined membership to categories of linguistic units which shared certain properties, while syntagmatic relations are shared between linguistic units occurring in a syntagm (a meaningful segment, or span, in a sentence). Paradigmatic relations describe categories at different levels of complexity (e.g. lexical categories at the word level, constituent categories at the span level...). In practice, paradigms are usually formed using substitution tests which grant a set of linguistic units similar roles or properties in given contexts. For example, nouns can be substituted for one another in a sentence while keeping syntactic well-formedness unaffected, and a noun referring to an edible object can replace another as the object of the verb "to eat" while keeping semantic well-formedness unaffected. Syntagmatic relations in turn are shared between different units in a well-formed syntagm, and define mutual constraints imposed by the position of units in sentences. For example, syntagmatic relations between grammatical constituents in a sentence are the mutual ordering constraint they impose on each other. While structuralism is considered to be obsolete by certain linguists, these principles laid foundations for subsequent linguistic theories as most of them inherit from these basic concepts in their descriptive goal.

Finding procedures which generate any possible string of a language Later, Louis Hjelmslev posited that the main requirement for linguistics is that the theoretical apparatus should be able to construct any possible string of a given language [Hjelmslev, 1957]. He also admits that various procedures could be adequate in

achieving the descriptive goal, by being both exhaustive and devoid of contradictions. In this case he argues that the simplest theory should be favored.

Chomsky builds on these principles in his first theory of Generative Grammar [Chomsky, 1957]. One goal sketched since the early stages of this theory is to propose a finite set of theoretical objects and rules forming a grammar which can generate exactly the well-formed, or grammatical strings of a given language, and none of the sentences that are not well-formed.[2] In this sense, such theories are computational, in addition to being formal descriptions: they provide procedures which construct sentences from a finite set of rules and categories. Chomsky defines two prerequisites for a grammar to be hypothetically valid. The first one is that sentences generated by the grammar should be acceptable to a native speaker. The second is that a grammar should be constructed based on a theory that is common to grammars for all languages. Finally, the decision criterion between grammars respecting such prerequisites is, as for Hjelmslev, the simplicity criterion. As grammatical strings don't need to be meaningful, one notable novelty in Chomsky's *Syntactic Structures* [Chomsky, 1957] is the independence of grammaticality from meaningfulness. Subsequent theories in generative grammar shared this objective, to account for all possible strings, and proposed updated their hypothesized representations of sentence structure to account for additional constraints. We should note that in this view, as grammatical sentences need not to be meaningful, these constraints are not derived from sentences found in corpora of natural language but rather in native speakers' intuitions about grammaticality.

Formulating assumptions about the computational system supporting linguistic abilities In a review to Skinner's behaviorist view on language [Chomsky, 1967], Chomsky later proposes that a native speaker's knowledge of grammars must originate from an innate faculty, sketching his first version of the Poverty of Stimulus argument.[3] These first hypotheses about the information processing system supporting language acquisition pave the way towards a computational understanding of linguistic abilities. Such novelty marks a fundamental distinction from the structuralist approach, as one goal pursued by the generativist program is to make predictions about the computational system which support the language faculty. In this regard, some generative linguistic theories make assumptions about the computations underlying linguistic competence, that is a speaker's knowledge of language. This view on the role of linguistic theories therefore grants them with a supplementary role, in addition to accounting for grammatical sentences:

[2] Hence, the binary view on well-formedness in our definition of natural languages dates back to at least these theories.

[3] Note that this term will only be used in later work [Chomsky, 1980]

they state computational principles supporting the proficiency of native speakers. These principles still guide more recent theorization of language [Chomsky, 1986; Chomsky, 1995; Adger, 2003; Boeckx, 2006; Berwick, 2015].

We do not attempt reviewing exhaustively all approaches to modeling grammar in linguistics as there have been a plethora of theories over the past decades. Let us keep in mind that each has its own procedure for generating and selecting hypotheses on how linguistic structure can be described. As such they can serve as inputs to understanding how the latter is supported. Moreover, some linguistic theories also make (generally implicit) cognitive assumptions about the information processing systems able to support linguistic knowledge. We nuance the utility of linguistic theories in providing concrete computational hypotheses regarding how languages are effectively processed, as (i) theoretical structures generally fail to achieve broad coverage of real-world data, although some theoretically motivated parsers based on Combinatory Categorial Grammar [Hockenmaier, 2002; Hockenmaier, 2003; Clark, 2007], Head-driven Phrase Structure Grammar [Zhou, 2019] and Lexical-Functional Grammar [Riezler, 2002] are robust to handle large corpora (ii) many of the computational principles remain implicit as most theories lack a clear causal description for how knowledge of certain linguistic constraints can be handled by an information processing system to generate sentences or predictions in context. However, linguistic theories can serve as a starting point for computational hypotheses. For instance, they yield constraints over certain categories of words, which can lead to formulating assumptions about the representations captured by a neural language model, and about the operations by which it abstract away from its input sentences. We discuss this in the next paragraph.

3.2.2 Linguistic theories help understand my model's abilities

With the previous two goals in mind, we underline the duality of the contribution of linguistic investigations to understanding NLMs. First, linguistic theories provide descriptions of linguistic knowledge in the form of constraints and rules, which can be used to assess a model's linguistic abilities. If one views linguistic competence as the capture of a set of constraints, then each of these, as described in a given theory, can be tested independently. A by-product of linguistic descriptions is that they provide a pool of corpora which distinguish grammatically acceptable and unacceptable sentences, as pairs are widely used by linguists to illustrate certain phenomena which grammars need to account for. Isolating specific phenomena can drive the analysis of a NLM's knowledge, either using a normative linguistic reference [Linzen, 2019], or by comparing the model's probabilities to grammati-

cality judgements of native speakers. Second, they serve as a pool of hypothetical principles regarding how languages are processed, that is regarding the computations supporting linguistic knowledge and the operations by which language can be interpreted and produced. Therefore they allow us to make assumptions regarding the nature of the representations and computations which support linguistic abilities in artificial systems.

3.2.3 Understanding the abilities of NLMs reveals how, or how else languages can be processed

Gaining better understanding of how NLMs process languages in turn can lead us to challenge hypotheses from linguistic theories. These can concern conditions under which processing systems can handle natural languages. These conditions can apply to the support itself, that is the architecture – whether it has certain biases, or the representations and computations that it supports – that is how structure is represented and by which processes the whole is built from parts. As an example, [Warstadt, 2019] argues that if a NLM with no prior knowledge of syntax reaches human-like performance on a domain-general set of acceptability tasks, it would put into question the Poverty of the Stimulus argument [Clark, 2011]. However, while many neural networks language models do not have any structural prior directly mirroring knowledge of syntax, it is not a generality as some language models do incorporate structural inductive biases [Shen, 2017; Shen, 2018; Kuncoro, 2018b], an issue extensively discussed in [Lappin, 2021]. [Linzen, 2021; Baroni, 2021] make a similar argument on the role of analyzing NLMs' linguistic abilities. For these reasons, studying state-of-the-art neural architectures should be of interest to linguists, in addition to being greatly useful to practitioners.

3.3 Decomposing the Analysis of Linguistic Knowledge in Information-Processing Systems

3.3.1 The hard problem of linguistic knowledge

Our earlier definition of linguistic generalization only encompasses a model's capacity to produce correct outputs, or producing probability estimates which respect given constraints. However, a comprehensive description for the nature of linguistic knowledge captured by NLMs cannot remain agnostic on the way they represent linguistic concepts and perform computations supporting generalization. It is equally important to understand what categories they encode and how they repre-

sents their mutual constraints to produce adequate responses. In this section, we stress the importance of understanding which mechanisms grant the output probability distribution desirable properties from the perspective of linguistic generalization, in addition to evaluating whether such distribution respects given constraints. Grasping the complexities of the model's inner mechanisms is an ambitious goal, as we have seen in section 2.2 transformer-based NLMs are complex systems. On the other hand, the problem of surface linguistic generalization is also difficult to address, as constraints on its surface probability might be hard to define as the complexity of phenomena addressed increases.

3.3.2 The three-levels of analysis

The multiple facets of our research question make it too arduous to be analyzed in a single level of description. Facing these complexities, we borrow principles from computational neuroscience to analyze the acquisition of linguistic abilities by NLMs. Specifically, we can draw inspiration from David Marr's three levels of analysis [Marr, 1982] for information processing systems, presented in table. 3.1.

At the **computational** level, we depart from the basic function of a NLM's architecture: reducing its loss and approximating p^{true} robustly while having access to limited data. This goal is pursued during training using an optimization scheme. As the loss is computed at the output level of the NLM, this level of analysis is tied to the surface behavior of the model: does the model generalize to unseen well-formed sentences? Do its outputs reflect human-like generalization? These questions should be asked in relation to the function they could play. The model might be capturing linguistic abilities precisely to reduce its training objective, as aspects of linguistic or human-like generalization could just entail statistical generalization – they could be strategies by which the model reduces its training objective. However, other strategies can be sufficient to reach statistical generalization but not linguistic generalization, which needs to be envisaged whenever formulating hypotheses or being faced with evidence that the model is able to produce correct outputs – it might not necessarily reflect the capture of linguistic abilities.

At the **algorithmic** level, we can formulate hypotheses on how an information processing system could capture and mobilize linguistic knowledge. This is the level in which we seek to describe the abstractions and representations captured by the information-processing system, and the operations which are applied to them. [4] These hypotheses can be borrowed from linguistic theory, or they can be alterna-

[4] While algorithms are discrete sets of instructions applied to discrete sets of variables, neural models compute continuous functions over real-valued vector representations. The algorithmic level is an abstract simplification of the computations and representations. These could for example

Level	**Main questions addressed**	**Application to the analysis of linguistic abilities in NLMs**
Computational	What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?	The learning objective is language modeling or masked language modeling. The goal of any computation in the model is to help approximate p^{true} for sentences or masked words. Therefore the goal of any computation taking place in the network is to produce certain outputs, or to produce certain behavior which approximates p^{true}. This could be achieved by acquiring linguistic knowledge similar to that of humans, or by relying on strategies that are specific to certain artificial systems.
Algorithmic	How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?	Assuming that the model captures a certain linguistic ability, what would be possible algorithmic models? Hypothetical models from linguistic theories provide structures and categories which could be represented by my model. Other hypotheses, potentially not found in linguistic theories, can and must be envisaged.
Implementational	How can the representation and algorithm be realized physically?	Given a hypothetical algorithm or strategy, how are its constituents – i.e. its elementary representations and operations – implemented in the NLM's layers?

Table 3.1: Marr's three levels of analysis of information processing systems (from [Marr, 1982]), their description in the original paper (second column) and how they can translate into understanding the linguistic abilities of NLMs

be derived from the constraints imposed on the output probability: the outcome of the computa-

tive hypotheses. Compositionality, systematicity, or memorization for instance are hypotheses at the algorithmic level.

At the **implementational** level, the questions asked in turn comprise whether a given architecture can support computations described at the algorithmic level, and how this can be implemented in representations and transformations which the network comprises in its layers. For example, this comprises questions related to how linguistic properties are encoded in the architecture, or which architectural biases can support which type of computation.

Uncovering the capture of linguistic knowledge by a NLM therefore implies understanding what information the model processes, and how that information is processed, at all these levels. In addition to studying the model's output behavior, it is necessary to formulate hypotheses regarding the algorithmic nature of the computations which take place in the model. Finally, it is equally important to understand which information the model encodes in its representations, and which operations are applied to these representations.

In cases where the model demonstrates linguistic abilities at the surface behavior level, it might rely on a vast spectrum of possible strategies, at the algorithmic level. Some of these could be robust and rely on learned rules which are applied systematically in any context. Others can be supported by the brittle memorization of lexical patterns or shallow heuristics which can cover some cases, such as frequent patterns seen during training, without extrapolating the rule accurately to its whole domain. [Baroni, 2020] argued that even if they do not seem to possess compositional skills, LSTMs are very proficient at handling natural language.

Note that at most, linguistic theories provide hypothetical constraints at the algorithmic level, and on the representations of linguistic units and the structure in which they occur. The implementational level is much more complex to address as hypotheses made on the neural substrate supporting a given algorithm or encoding a certain representation are very hard to verify in humans.[5] While there is a vast body of research in neurolinguistics [Zaccarella, 2015; Olstad, 2020], neural connectivity in the human brain is extremely complex. Hence, describing the networks responsible for high-level computations is a very arduous task, let alone measuring activity in a targeted area with sufficient temporal and spatial resolution to confirm hypotheses on the neural code supporting language.

tion. The relation between connectionist models and discretized representations and rules has been extensively discussed in previous work [Pinker, 1988; McMillan, 1988].

[5]This is not a generality, as the Poverty of Stimulus does make an assumption at the implementational level. A body of psycholinguistics also makes measurements of brain activity when processing language, which gives limited evidence for the neural substrate supporting abilities mobilized on given tasks.

We can sum up the interconnections between the different levels of analysis as follows:

- The output behavior, which is constrained to approximate the true probability of sentences or completions during training, is supported by a toolbox of algorithms. Behavioral evidence for a given linguistic allows us to formulate or disprove hypotheses at the algorithmic level.
- Algorithms rely on discrete representations and operations, which are implemented in neural states and layer transformations. We can prove that an algorithm is implemented in the architecture by decoding the neural substrate responsible for the hypothesized computations. We can also prove theoretically that an algorithm is not learnable by a given architecture, or empirically by using carefully designed datasets, e.g. artificial data.

Given this broad picture, we first see how linguistic abilities can be tested at the surface behavior level, before going deeper in our levels of analysis.

3.4 Linguistic Constraints over a Neural Language Models' Behavior

In this section, we present the different types of constraints which can be used to test a model's output probabilities. We first discuss the different sources of such constraints, which can represent binarily a sentence's well-formedness and be derived from a normative view on grammaticality, or they can rather take the form of a degree of adequacy and reflect gradient acceptability judgements. We then present the different scopes to which the constraint applies, either to a span or sentence-level probability, or a token-level probability. We further dress a typology of the constraints which can be tested at the surface behavior-level and finally introduce some metrics which can be used in the different cases mentioned.

3.4.1 On the nature of linguistic constraints

In section 3.2, we discussed how linguistic theories can contribute to understanding NLMs. In particular, they provide a vast testbed to examine whether NLMs capture certain phenomena at the behavioral level, that is whether their output probabilities respect certain constraints on well-formedness, or conform to human acceptability judgements. In their simplest form, linguistic descriptions of linguistic phenomena provide examples of minimally different sentences, or procedures to generate such

sentences, to pinpoint a specific phenomenon. Examples like these can distinguish sentences by their grammaticality or degree of acceptability.[6] These can be used to test whether the model's behavior, or output probability vector, captures the targeted phenomenon.

Before going further, it is important to note that the empirical validity of constraints proposed by linguistic theories has been the subject of debate over the last decades. The plurality of linguistic theories warns us that the constraints which they describe should be questioned, in particular if empirical evidence is missing or incomplete to support them. [Ferreira, 2005] raised several issues regarding developments in generative grammar. In particular, Ferreira argues that some of the theoretical apparatus is hard to translate into computationally testable material, and that the empirical evidence supporting theoretical claims is sometimes weak. [Phillips, 2009] further wrote a response to allegations like these. While being sympathetic to some of the criticism, he put into question that widely accepted theoretical claims are the fruit of misleading, intuitive judgements supported by poor evidence. In his paper, he argues that evidence for such cases is in itself missing, and puts into question that collecting more carefully acceptability judgement could be a solution to any of the problems raised. In a subsequent paper, [Gibson, 2013] challenge this statement and respond to several other arguments from the previous literature, addressing criticism to traditional methods in theoretical linguistics. The authors give three examples of theoretical claims based on intuitive judgements which led to erroneous yet influential generalizations in the field, advocating in favor of more quantitative approaches. [Sprouse, 2013] later responds to Gibson and Fedorenko that their claims are untrue, on the basis that empirical evidence predominantly supports the reliability of data provided by traditional methods. By providing large-scale assessments proving the well-groundedness of the traditional syntactic literature, they prove that an overwhelming majority of the data present in syntactic theory is verified empirically. In this thesis, we stand by this position, however, we also recognize that one should be critical when testing a processing system against constraints taken from linguistic theory. In this regard, we call for careful scrutiny regarding the validity of data used to test whether a computational system captures a given linguistic phenomenon.

Another important divide concerns the views which coexist in how they conceive grammaticality. We already introduced the binary view on well-formedness in section 2.1. Another main approach is to view grammaticality as a gradient property of sentences. We display these views in table. 3.2.

While the binary conception of well-formedness has been prominent in linguis-

[6] These two notions are not interchangeable, see [Lau, 2016].

View	Description
Binary	Grammaticality is membership to a set of well-formed sentences. Well-formedness is a binary property of word sequences.
Gradient	Grammaticality is real-valued, and not a binary property.

Table 3.2: Two possible views on grammaticality.

tics, there have also been views that grammaticality is rather gradient [Fanselow, 2006; Ambridge, 2016]. Under any of the two approaches, our first goal is to understand how it translates into evaluating a NLM's ability to capture aspects of grammaticality.

As neural language models are probabilistic models, their capture of well-formedness as a binary property of sentences cannot be directly assessed. Furthermore, conceiving grammaticality as gradient does not solve this issue as as it remains a theoretical concept, therefore it is not directly measurable. In principle, this poses a problem for the empirical evaluation of a model's grammatical knowledge. [Lau, 2016] propose one way to overcome this issue by considering acceptability, which in turn is an empirically grounded property which can be quantified over certain sentences by asking speakers to produce judgements over sentences. However, they argue that if we are to predict acceptability based on the magnitude of a language model's probabilities, a number of precautions should be taken. For instance, acceptability is not directly reflected by a model's probability estimates for word sequences due to length and frequency effects on sentence probabilities. They further propose to modulate the log-probability outputs of language models to neutralise such effects.

Another way to overcome this problem is to use minimal pair testing. Whether we subscribe to a binary or gradient view on grammaticality, we can make assumptions about the ordering of the model's outputs and compare the probability for two different completions given a context in sentences forming a minimal pair. Under the binary view, we can expect that the completion from the well-formed sentence has a higher probability than of the ill-formed sentence. This however holds only in the masked setting, in which length does not affect probability. Under the gradient view in turn, we can also expect that the more grammatical completion would be assigned a higher probability at the position in which it differs from the less grammatical sentence.[7] In this case, ordering constraints, as opposed to magnitude-based evaluation, can be envisaged to assess a language model's capture of grammatical knowledge.

A working hypothesis that can be adopted consists in seeing well-formedness

[7]Let us keep in mind that frequency could still play a role, as noted by [Lau, 2016]

as resulting from a set of constraints, some of which can be isolated. Under the binary view, well-formed sentences would be exactly the ones respecting all well-formedness constraints. Under the probabilistic view, each of the well-formedness constraints would impose degrees of grammaticality over sentences and completions. This idea has been proposed early in linguistics [Chomsky, 1965b] and has been developed in later work [Chomsky, 1986; Hayes, 2000]. As noted by [Lau, 2016], approaches like these remain marginal in and a comprehensive formal model of gradience is still missing. Well-documented discrete constraints could still be leveraged to test the capture of targeted knowledge. Considering that these constraints are disjoint, linguistic generalization can be studied as the extent to which the model is extrapolating each of these *well-formedness* constraints from its training data.

As noted previously, well-formedness constraints are not observable *a priori*, and remain hypothetical. Formulations of such constraints, their nature and scope, can further be tested regarding how well they hold against linguistic data, and whether they predict acceptability judgements. While the relationship between grammaticality and acceptability scores is unclear, one could expect that isolated well-formedness constraints not conflating factors affecting acceptability would lead to a greater acceptability score for the more grammatical sentence. Given well-established constraints supported by human judgements, our starting point to seek aspects of linguistic generalization in a probabilistic model is therefore whether such constraints translate into the model's probability distribution over sentences and completions.

As seen in the previous section, the starting point to investigate a NLM's linguistic abilities is its surface behavior as the training objective directly impacts the output probability distribution. When attempting to gain a better understanding about how the latter is shaped, one can design behavioral tests which define expected behavior under certain hypotheses, that is constraints over the predictions of the model. When analyzing its output probabilities, we can ask questions such as:

- Does the model extrapolate certain well-established linguistic phenomena to any sentence?
- Does the model make predictions which align well with human judgements?

The two main probability scores that we are interested in are sentence probabilities and token probabilities, as those are the two types of outputs we can collect from a NLM. Over the past decades, investigations on grammaticality have led to collecting judgements from native speakers on a wide array of isolated linguistic

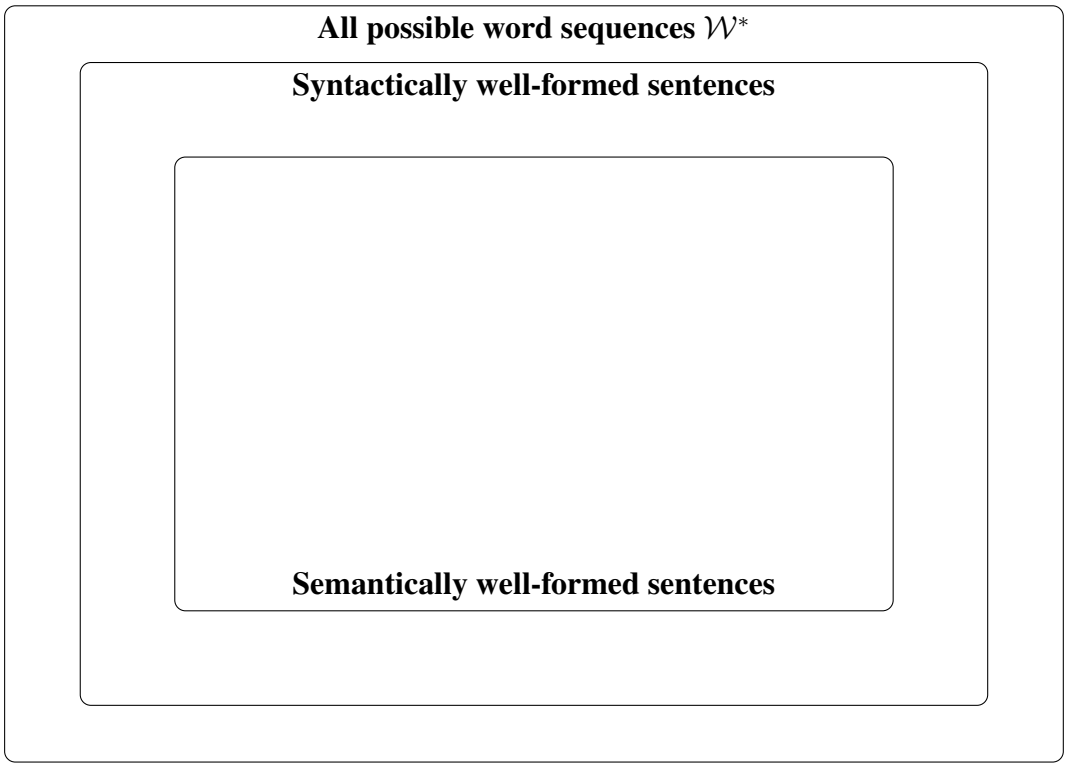


Figure 3.1: Schematic representation of well-formedness constraints as sets of sentences. This set-based representation mirrors the binary view on well-formedness, for simplification purposes.

phenomena. Behavioral tests have been largely deployed over the past decades in linguistic investigations, often taking the form of grammaticality or acceptability judgement tests. Such studies have permitted to compare judgements made by human native speakers of a given language to linguistic theories. In computational linguistics, they constitute a rich testbed to investigate the linguistic abilities of a model's linguistic knowledge. When treating models as blackboxes, one only has access to output probability estimates given a certain context to diagnose a language model.

Leveraging acceptability tests to diagnose a model's predictions As seen in the previous chapter, neural models learn a probability distribution over completions given a context, either a left context $p(t_i|t_1 \dots t_{i-1})$ for autoregressive language models, or the whole (left and right) context $p(t_i|t_1 \dots t_{i-1}\text{MASK}t_{i+1} \dots t_n)$ for bidirectional models. By their nature, autoregressive language models are suited to be tested against constraints over its probabilities which hold at the sentence-level, and bidirectional language models are best suited to be tested against token-level constraints. We can note that token-level constraints can also be tested in autoregressive language models, either using only the token probabilities given their

left context, or using the sentence probability even if the test targets a specific token in that sentence. This has implications regarding the type of phenomena which models can be tested against. In the following, we attempt taxonomizing the different types of linguistic constraints and phenomena which can drive the investigation of linguistic knowledge in language models.

3.4.2 Scope of linguistic constraints

Constraint over sentence-level probabilities As seen in eq. (2.1), an autoregressive language model's learned probability distribution can be iteratively applied from left to right to estimate a sentence probability $p(s = t_1 \dots t_n)$. Naively, the simplest constraint level that one could think of at the sentence level, would be the following:

$$\begin{cases} \text{if } s \in L,\ p(s) > 0 \\ \text{if } s \notin L,\ p(s) = 0 \end{cases}$$

In practice, it is impossible for a neural model to assign 0 probabilities due to the strict positivity of the widely used softmax function. A milder desirable property could be that incorrect word sequences are assigned arbitrarily low probabilities. In practice, it is not possible to define a threshold for grammaticality, as assuming the existence of such threshold τ would imply that there exist a finite number of well-formed sentences, at most $\lfloor \frac{1}{\tau} \rfloor$ [Lau, 2016]. Instead, linguistic tests in neural language models often take the form of preference judgements, where it is rather the ordering of probabilities that is considered.[8] For a pair of word sequences (s_1, s_2) such that $s_1 \in L$ and $s_2 \notin L$, a weak expectation is that probabilities assigned by the language model should respect the following constraint: $p(s_1) \geq p(s_2)$.

As neural language models are not perfect approximations of the true distribution underlying a given language, they are often tested using targeted evaluation tasks, which evaluate a targeted ability. Some linguistic phenomena naturally define preference for a sentence over another in a given minimal pair, either because one is judged acceptable but not the other, or because one is judged more typical than the other. For such a pair (s_1, s_2), testing a neural language model equates to comparing probabilities assigned to s_1 and s_2. This has led researchers to isolate linguistic phenomena which can be tested in such straightforward way [Linzen, 2016; Marvin, 2018; Warstadt, 2020b].

[8] In [Lau, 2016] though, authors propose to adopt functions which take logprobabilities as inputs, but do not sum to 1, to compute well-formedness scores and avoid the issue mentioned.

Constraint over token-level probabilities given contexts Bidirectional, or masked language models differ in nature from autoregressive models in that they do not model the probability of a word given only a left context. This has profound implications, as contrarily to autoregressive models, their learned probability distribution cannot be used to estimate sentence probabilities using the decomposition previously seen in eq. (2.1). However, they encountered undeniable success due to their proven robustness on a wide array of tasks are now extensively used by practitioners and researchers [Devlin, 2019a; Liu, 2019c; He, 2020b]. While the tests presented in the previous paragraph cannot be applied "as is" to masked language models, they can be evaluated at the token level, simply by masking one word in the sentence. This certainly is more limiting compared to autoregressive language models which are more flexible as the chain-rule can be applied to compute sequence probabilities. Nonetheless, many acceptability judgements tests can be performed by bidirectional models, in which minimal pairs only differ in one word, given that each word from the differing word-pair is part of the model's vocabulary as a single token. For any pair of tokens in our model's vocabulary $(t_1, t_2) \in \mathcal{V}^2$, such that t_1 is an acceptable completion given a context c, and t_2 is not, we can evaluate the model in its training setting and expect it to respect $p(t_1|c) \geq p(t_2|c)$. This also holds true if t_1 is judged a more typical completion that t_2.

3.4.3 Typology of constraints over sentences

A plethora of linguistic phenomena which have been documented in the past decades can be tested in the form of minimal pairs, as they are generally described using examples where one sentence is acceptable and the other is not. In this section we attempt taxonomizing such phenomena in terms of how sentences in a pair minimally differ from each other, and finally give some implications that such differences have on how these corpora can be leveraged to perform behavioral tests in language models. We present examples of such phenomena in table. 3.3.

Sentences as word sequences

As seen in section 2.1.1, sentences are sequences of words which respect a number of constraints, some of which are explicit, some of which are hidden. By examining corpora of utterances from native speakers, grammarians uncover linguistic phenomena by describing regularities in linguistic data. They might do so by giving examples of sentence pairs, one of which is acceptable and the other is not. These sentences often differ minimally to pinpoint how linguistic constraints apply to utterances. To our language models, input sentences are discrete word sequences,

Type	Subtype	Name	Paper	Example
Morphological	Number agreement	Word content	[Linzen, 2016]	The man laughs. / *The man laugh
Syntactic	Word order	Movement	[Baltin, 1982]	Would the men not enjoy that? / *Would not the men enjoy that?
	Word order	Comparative Clause	[Bresnan, 1973]	Jack eats caviar more than he sleeps. / *Jack eats more caviar than he sleeps.
	Word content	Comparative Clause	[Culicover, 1999; Bresnan, 1973]	I am more angry than sad. / *I am angrier than sad.
	Word content	Modality	[Dayal, 1998]	You may pick any flower. / *You must pick any flower.
	Word order	Coordination	[Gazdar, 1981]	I wonder who saw Bill and liked Mary / *I wonder who Bill saw and liked mary.
	Word content	Coordination	[Gazdar, 1981]	To which city and which conference did Bill go? / * Which city and to which conference did Bill go to?
	Word content	Negative Polarity	[Kadmon, 1993; Marvin, 2018]	I don't have any potatoes. / I have any potatoes.
	Word order	Passive	[Collins, 2005]	The book was written by John. / *The book was by John written.
	Word order	Predication	[Williams, 1980]	Who do you think of as silly? / *Of whom do you think as silly?
	Word content	Gapping	[Jackendoff, 1971]	Bill ate more peaches than Harry did. / * Bill ate the peaches and Harry did.
	Word content	Reflexive Anaphora	[Marvin, 2018]	The senators embarrassed themselves. / *The senators embarrassed herself.
Mixed	Word content	Resultative	[Goldberg, 2004]	They drank the pub dry / *They drank the pub
	Word content	Sluicing	[Chung, 1995]	She served the soup, but I don't know to whom. / *She served the soup, but I don't know who.
Semantic	Word order	Event knowledge	[Chow, 2015; Ettinger, 2020]	The restaurant owner forgot which customer the waitress had served / *The restaurant owner forgot which waitress the customer had served
	Word content	Negation	[Fischler, 1983; Ettinger, 2020]	A robin is a bird / *A robin is not a bird
	Word content	Hypernymy	[Fischler, 1983; Ettinger, 2020]	A robin is a bird / *A robin is a tree
	Word content	Commonsense	[Federmeier, 1999; Ettinger, 2020]	He caught the pass and scored another touchdown. There was nothing he enjoyed more than a good game of football / *baseball

Table 3.3: Tentative taxonomy of minimal pairs isolating certain linguistic phenomena. Most, but not all of these works were compiled in the Corpus of Linguistic Acceptability [Warstadt, 2019]

which can be seen as a set of word contents and an ordering. The differences in input between an acceptable and an unacceptable sentence might thus only be differences in ordering or in content.

Constraint types

This formulation leads description of linguistic phenomena described in the literature in the form of acceptability judgements to impose constraints on either (i) an *ordering* of words, or (ii) constraints over the *form* or *content* of these words. We also consider a third constraint as a special case of (ii), where one sentence contains a few more or a few less words than the other, and the rest of the content and relative ordering is intact. These can be described as (iii) constraints over the *presence* or acceptable *omission* of certain grammatical words.

Note that these constraints are not always exclusive. For instance, comparative

clause construction can impose constraints on the ordering of words in the following example:

(1) a. Jack eats caviar more than he sleeps.
b. *Jack eats more caviar than he sleeps.

But it can also impose constraints on the form of certain words:

(2) a. I am more angry than sad.
b. *I am angrier than sad.

Other phenomena, such as movement, generally impose constraints over the ordering of words:

(3) a. Would the men not enjoy that?
b. *Would not the men enjoy that?

Types of linguistic phenomena

(i) Syntax Some of the phenomena which are usually tested using minimal pairs address knowledge about syntax – the constraints which are described are only dependent on the syntactic structure of sentences, independently on the content of words.

(ii) Semantics Some linguistic phenomena are clearly semantic in the sense that they impose constraints which depend on the meaning of words used in the sentence. They result from semantic relations between words, among which we can cite hypernymy, synonymy, antonymy, encyclopedic and event knowledge, pragmatics, and reasoning abilities.

(iii) Morphosyntax Finally, some linguistic phenomena impose morphosyntactic constraints, as they impose a certain inflection of certain content words in the sentence. This is also at the interface between semantics and syntax, as the morphosyntactic constraint builds on the syntactic structure of sentences, while also bearing semantic information (e.g. number, gender, tense) of inflected words.

3.4.4 Examples of constraints and corresponding metrics

Note that in this section, we focus on evaluation methods which apply to probabilities of completions given contexts $p(y|x)$ as we mostly focus on bidirectional transformer-based models in this thesis. Many of the following metrics however,

can be transposed to sentence-level probabilities by substituting $p(x)$ to $p(y|x)$. Here, p denotes the probability estimate which the model outputs.

(A) Ordering-based evaluation

Some behavioral tests impose an ordering over output probabilities. In this setting, what is measured is the preference judgements of the model, that is whether it judges one sentence, or a completion given a context, more plausible than another. In this regard, the magnitude of probabilities is ignored.

(i) Binary ordering Can be evaluated using accuracy, where success is a binary measure of whether the model orders probability of two possible completions, or two sentences, in a given order.

The binary outcome for a single pair which comprises a correct completion y_t and an incorrect completion y_f writes:

$$\mathbb{1}[p(y_t|x) > p(y_f|x)]$$

which can be summed:

$$\text{Acc} = \frac{\sum_{x,y_t,y_f \in \mathcal{D}} \mathbb{1}(p(y_t|x) > p(y_f|x)}{|\mathcal{D}|}$$

(ii) Ordering of multiple answers Kendall's Tau correlation coefficient is suited to compare the ordering of a model's output probabilities to that of a reference.

$$\tau = \frac{n_c - n_d}{n_0 - n_1} \tag{3.1}$$

n_c is the number of concordant pairs between the two orderings, n_d is the number of discordant pairs, n_0 is the total number of pairs and n_1 is the number of ties for our linguistic reference. We do not count ties in n_c nor in n_d. It follows that if BERT's ordering follows our linguistic reference, this coefficient is equal to 1, as $n_c = n_0 - n_1$ and $n_d = 0$. In the worst case, this coefficient is equal to -1 as $n_d = n_0 - n_1$ and $n_c = 0$.

(iii) Set of correct and incorrect answers We adapt binary probability by comparing all pairs, given a set of true answers $\mathcal{Y}_t(x)$ and a set of false answers $\mathcal{Y}_f(x)$ for each context x:

$$\frac{\sum_{y_t,y_f \in \mathcal{Y}_t(x),\mathcal{Y}_f(x)} \mathbb{1}[p(y_t|x) > p(y_f|x)]}{|\mathcal{Y}_t(x)|}$$

Called equally-weighted in [Newman, 2021b].

(B) Magnitude-based evaluation

(i) Binary relative probability In cases where a researcher makes use of minimal pairs, or any pair of answers one of which is considered to be more acceptable given the context, the relative probability of such answers can be considered. Given a correct and an incorrect completion, one can compute their relative probability to measure how much the model captures the phenomenon under investigation:

$$\rho = \frac{p(y_t|x)}{p(y_f|x)}$$

Also called sensitivity in some papers

(ii) Set relative probability In some cases, output probabilities of the model are compared to a set of real valued scores. These include data collected from humans, such as scores of relatedness, or selectional preferences. The latter can be fitted against model probabilities. A linear model's slope for instance will scale probabilities and a good fit for such model indicates that the relative probabilities of predictions respect that of the reference with which it is compared (e.g. scores from humans). This is typically evaluated using the R^2 coefficient of determination between predictions and the reference, which is equivalent to Pearson's linear correlation coefficient.

(iii) Probability mass In cases where a set of predictions is considered correct and another set of predictions is not, instead of considering a binary comparison of each word pair in the correct set and the incorrect set, one can compute metrics over the probability mass of the two sets. For example, in [Newman, 2021a], authors use the following metric, which intuitively captures the model's systematicity in assigning higher probabilities to correct completions:

$$\frac{\sum_{y_t \in \mathcal{Y}_t(x)} p(y_t|x)}{\sum_{y_t, y_f \in \mathcal{Y}_t(x), \mathcal{Y}_f(x)} p(y_t|x) + p(y_f|x)]} \tag{3.2}$$

(C) Which should I use?

Depending on whether the reference's truth value makes sense as a magnitude or not, one might be forced to choose orderings instead of probabilities. For example, in the case where some completions are considered correct and others are not, e.g. under the binary view on well-formedness, one has to use ordering-based metrics.

In cases where the model's predictions or sentence probabilities are compared to real-valued scores, e.g. human judgements of acceptability, depending on the number of comparable data points, one can choose one of the magnitude-based metrics. In the empirical portion of this thesis, we will use both types of metrics.

In this section, we illustrated the diversity of constraints which can be tested against a NLM's output probabilities. In doing so, we presented the source of such constraints, reducing them mainly to normative, discrete, linguistically motivated constraints, or gradient constraints resulting from human judgements. These only apply to the model's surface behavior, the starting point of our analysis, according to our three-levels presented in section 3.3. To address questions at the deeper algorithmic and implementational levels of analysis, we describe in the next section a typology of linguistic units which could be represented by the model when processing sentences to output certain behavior.

3.5 Building blocks for the algorithmic level: A typology of linguistic units

In order to reach a comprehensive account for how predictions are made at a behavioral level, we eventually need to make hypotheses at the algorithmic level. This requires making hypotheses about the categories and relations represented by a given system. As seen in section 3.2, linguistic theories formulate hypotheses regarding how sentence structure is represented. For grammars to describe such structures, they first need to enumerate *inter alia* classes of words, their inflections, their functions and most importantly their relations in sentences. In doing so, grammars typically define categories of linguistic units at various levels of complexity and scopes, each with their distinctive properties. The motivation for giving broad formal definitions of such linguistic units is to distinguish building blocks found in models of syntax, which can further drive detailed accounts for how neural models process sentence structure. Note that there exist a rich tradition of formal language theory [Chomsky, 1956a; Chomsky, 1959; Chomsky, 1963], and mathematical linguistics [Bar-Hillel, 1953; Lambek, 1958; Kracht, 2006]. The goal of this section is not to attempt providing specific formal descriptions of sentence structure, or a detailed mathematical system comprising a large set of intricated definitions. Instead, we attempt providing a few simple definitions, and are aware that we are not exhaustive of objects as complex as natural languages. Our goal however, is to sketch a coarse-grained typology of linguistic units which lie in linguistic descriptions, to lay foundations for future computational investigations on how linguistic knowl-

edge is structured in NLMs. Another reason why we do not explicitly borrow from previous work in this section, is that our ultimate objective is to understand how abstractions are implemented in a neural language model, that is to *decode* them. As neural models process sentences as word sequences, and sometimes only see a portion of such sequences, we also desire that our unit evaluation functions are defined over word sequences and their parts. Additionally, NLMs produce a series of intermediate representations which lie in spaces that have the same structures, but encodings for a given linguistic unit might only lie in *some* of the intermediate space. For this reason, we also need our evaluation functions to take into account that the evaluation function can yield an "undefined" value.

In the following, we simply consider that computations supporting linguistic processing build on knowledge of (i) content categories, which are sets of words or word sequences, sharing certain properties[9] and (ii) relations between categories, which are sets of relations between content categories, and (iii) mutual constraints exerted by categories on one another (e.g. an acceptable relative linear ordering between two or more given categories, or the content which can occur at a certain position given its ordering relation to other categories). We define linguistic structure as the set of relations shared between content categories in a sentence. We further note that there are different various content and relation category systems. For example, syntactic dependency structure only define relations between words, while constituent structure, define relations between nested levels content categories.

Linguistic properties can describe a word in context, but also higher-level scopes, e.g. a word pair, a subsequence of a sentence, or a full sentence. Note that these properties are rarely defined over their scope independently of the context where they occur. For example, in English, the word "hit" has different attributes depending on the sentence where it occurs (see table. 3.4).

Context	**Lexical Category**	**Tense**	**Number**
"My brother hit the road after lunch."	Verb	Past	Singular
"Our country's economy was hit by the current crises."	Verb	Past Participle	Undefined
"That song is definitely a hit."	Noun	Undefined	Singular

Table 3.4: Attributes of the word "hit" in different sentences.

Note that the meaning of that word also varies greatly depending on the context in which it is used. For word sequences, the same holds true (see table. 3.5).

A property of a word (or a word sequence) in a sentence can therefore be defined as a function of that word (or sequence) and the sentence where it occurs.

[9]Such as contexts in which they occur

Context	Constituent Category
"The girl that I saw was gorgeous"	Noun Phrase
"I showed the girl that I saw an impressive footage of the riots"	Undefined

Table 3.5: Attributes of a word sequence in different contexts.

Property	Examples	Value
Lexical Category	"I love that idea."	Noun
	"He goes to the beach very sunday."	Verb
	"Is this the book you told me about?"	Determiner
Number	"This fact does not matter."	Singular
	"I received several letters from him."	Plural
	"There is a consensus on that in my group."	Singular
	"Jupyter revolves around the sun."	Undefined
Tense	"He left without saying a thing."	Past perfect
	"We meet around twice a year."	Present perfect
	"They stopped producing this model a few months ago."	Undefined

Table 3.6: Examples of word-level contextual properties

As seen previously, the property can be undefined for certain words or sequences, e.g. a noun has no tense. Word-level properties are then defined as a mapping between a word in a given sentence, and a set of values which includes the undefined value. Sequence-level properties are a mapping between sequences in a given sentence, and a set of values which include the undefined value. Some examples of word-level properties and sequence-level properties can be seen in table. 3.6 and table. 3.7.

Other properties for words or sequences in turn can be defined in relation to other parts of the sentence. For instance, the semantic role property defined in table. 3.7 can be defined relatively to a predicate, and dependency grammar assume that certain words share a dependency relation to their head in the sentence.

Definition 3.5.1. Broadly speaking, we can write contextual linguistic properties as a function defined over a sentence and some part of that sentence (e.g. a word or

Property	Examples	Value
Constituent	"The colors in his paintings are so vivid."	Prepositional Phrase
	"Farmers will have a hard time if it doesn't rain."	Noun Phrase
	"They really liked the illustrations Mary made."	Verb Phrase
	"Repeating this would certainly help remember."	Undefined
Semantic Role	"John closed the door when he arrived."	Agent
	"He ate his pizza in seconds."	Patient
	"They put at risk the success of this mission."	Predicate
	"It is not easy to choose between these options."	Undefined

Table 3.7: Examples of sequence-level contextual properties

a span), into the set of values which that property can take $\mathcal{V}$:

$$\mathcal{T} : (s, \bar{s}) \in \mathcal{W}^* \times \mathcal{P}(s) \rightarrow \mathcal{T}(s) \in \mathcal{V}^T \cup \{\text{undefined}\}$$

where $\bar{s}$ is used to denote any subpart of the sentence s and $\mathcal{P}(s)$ is used to denotes parts of s – spans are most often considered in this context.

Definition 3.5.2. In addition, we can define relational linguistic properties as a functions defined over a sentence and two parts of that sentence:

$$\mathcal{T} : (s, (\bar{s_1}, \bar{s_2})) \in \mathcal{W}^* \times \mathcal{P}(s)^2 \rightarrow \mathcal{T}(s) \in \mathcal{V}^T \cup \{\text{undefined}\}$$

In grammars, properties like these define categories which are the building blocks of linguistic structures. As seen previously, linguistic theories allow us to formulate hypotheses regarding how NLMs process linguistic structure. One starting point to examine whether certain types of linguistic structures are represented by NLMs would be to assess whether their categories are encoded by in the NLMs' intermediate representations, and what role such encodings play in determining the NLM's outputs. We will discuss how to reach this objective in chapter 7

3.6 Conclusion

In this chapter, we decomposed the hard problem of understanding the linguistic abilities of NLMs into smaller, more focused subproblems. By breaking down our investigation along various levels of analysis (behavioral, algorithmic and implementational), we laid down foundations to answer targeted questions by delimiting their scope, and their interactions with other levels of analysis. In this framing, the different aspects of the main question addressed in this thesis, that of linguistic generalization, are complementary and interdependent. The framework has strong methodological implications. First, none of the investigations on linguistic knowledge are addressed in total isolation from the model's initial goal: optimizing its objective function during training. The logical structure by which the subproblems are connected to this goal has epistemological implications, as it connects methodologies which at first glance were not translatable into each other, by describing explicitly the relationship that they have with each other. It remains agnostic however, on the order in which investigations ought to be performed, as these levels are interdependent. Bottom-up approaches where one targets an understanding of the implementational level – that of neurons and layer transforms – to reduce the hypothesis space of algorithms they can implement is as equally legitimate as starting from behaviorally-supported abilities, before trying to uncover the underlying computations in the network. In this systemic view on analyzing a NLM's abilities, the different approaches targeting different loci (encoding, output behavior, or components) can benefit each other as they are causally connected through the role they play in the model's systemic function - approximating a probability distribution. Additionally, we reviewed constraints which can be tested against the model's output probabilities for behavioral investigations, and taxonomized linguistic units and relations which can be represented by the model at an algorithmic level, which gives a toolbox for future investigations in this levels.

In the next sections, we first carry behavioral analyses and formulate hypotheses on the computations performed by the network at the algorithmic level: that is, which information it uses to perform certain linguistic tasks. In the last chapters, we target the implementational level and attempt understanding which representations and computations support the realization of a hypothesized algorithm.

Figure 3.2: "A man constructing a bridge between two piles of books floating in water, digital art", K.L. x DALL·E 2

Chapter 4

Behavioral Diagnosis of Syntactic Knowledge using Black Box Naturalistic Tests: the Number Agreement Task

Goals

The goal of this chapter is to present first results derived from the careful analysis of a neural language model's behavior. We examine the capture of linguistic knowledge on a syntactic task, subject-verb number agreement. In this setting, a model's behavior is evaluated in its training setting, masked language modelling. We make use of a carefully designed dataset to observe whether the model captures the number agreement rule in different settings, to test several hypotheses regarding the nature of such knowledge. We subsequently compare the model's behavior to human judgements, which allows us to observe whether the model's predictions and error patterns are concordant with that of humans. Behavior tests are useful as they are a direct reflection of the model's predictions in its normal functioning setting, the one in which it has been trained. In this regards, results from behavioral analysis are a first step towards gaining better understanding on the linguistic abilities captured by a neural language model.

Contents

4.1 Subject-Verb Number Agreement

As seen in the previous chapter, minimal pairs provide an interesting testbed, suited to investigate whether the behavior of a trained language model is consistent with grammaticality judgements that a native speaker could make. Pairs differing in only one word are particularly handy to diagnose bidirectional models as they allow us to test the model in its training setting – masked language modelling. The number agreement (NA) task is an example of such minimal pair tests, which has been largely studied in the past [Corbett, 2003]. In this section, we present the task and how it has been employed to test the behavior of neural language models.

4.1.1 Evaluating the capture of syntactic dependencies

Subject-verb number agreement is the rule according to which, in english, third-person present tense verbs must agree in number with their subject:

(1) a. The day is turning ghost.
b. *The day are turning ghost.
c. *The days is turning ghost.
d. The days are turning ghost.

The NA task consists in assessing whether a model's predictions show a preference for sentences that do not violate number agreement between a selected verb and its subject. The subject is typically called the **cue** of the agreement and the verb is called the **target**. Success on this task is usually taken as evidence that the model is able to track syntactic dependencies, a feature which is hypothesized to be key to many higher-level linguistic phenomena.

In (1), the verb immediately follows the subject. In this case, the model could rely a simple heuristic, that consists favoring a completion which has the same number as the preceding noun. To circumvent this issue, this test is often designed to contain sentences with varying linear distances between the subject and target verb of the agreement relation [Linzen, 2016]:

(2) a. The day when he saw his brothers is turning ghost.
b. *The day when he saw his brothers are turning ghost.
c. *The days when he saw his brothers is turning ghost.
d. The days when he saw his brothers are turning ghost.

In (2), the previous heuristic would fail as the noun preceding the target verb (which is typically called an **attractor**) can have a different number than that of the cue.

Years ago, a dataset of minimal pairs built from sentences extracted from Wikipedia has been released [Linzen, 2016], comprising examples with varying linear distances between the cue and the target. In their experiments, the authors tested the ability of LSTM models perform the NA task, and showed such models to capture syntax-sensitive dependencies when given targeted supervision. They show that a LSTM trained with a generic language modeling objective makes much more errors on the task and is not capable to perform well overall if its capacity remains unchanged. Additionally, they test a much larger publicly available language model, and show it to also perform poorly on the NA task, which leads them to conclude that explicit supervision is required to learn syntax-sensitive dependencies. Later work extends these results by testing a variety of supervised models and unsupervised language models on the NA task.

Assessing the capture of the rule on unseen sentences To demonstrate that the model captures the number agreement rule, we must show that it is able to generalize beyond its training data. In this thesis, we are interested in examining whether a model trained with a generic language modeling objective is able to capture linguistic abilities. In [Linzen, 2016], the authors show that to succeed on the task, LSTMs need to be explicitly supervised on number agreement, as a generic language model does not capture the rule. [Bernardy, 2017] strengthen these results, by showing the ability of various supervised models to learn the task. They additionally demonstrate that performance increases with the size of the training dataset, and with the size of the architectures under investigation. Additionally, they show that models need to be exposed with lexically rich data to learn the rule during training, demonstrating that they fail to generalize using only syntactic information in presence of limited vocabulary. More recently, transformer-based language models have also been shown to perform strongly on subject-verb number agreement in a wide array of settings, suggesting that they learned to track syntactic dependencies during their training even without explicit supervision. [Goldberg, 2019] in turn shows that BERT, a pre-trained transformer model, is able to perform very well on stimuli from [Linzen, 2016]. However, this evidence alone is not sufficient to demonsrate extrapolation of the number ageement rule, as this data is extracted from Wikipedia, which BERT is trained on. Other experiments in [Goldberg, 2019] however, show the model to perform well on data generated from diverse templates [Marvin, 2018], which is very unlikely to be part of the training data. This evidence suggests that such models are able to learn the rule beyond their training data, and without explicit supervision on the task, as they are only trained on a generic masked language modeling objective. This also seems to show

an advantage of Transformers compared to LSTMs as [Marvin, 2018] had showed that there was considerable room for improvement for LSTMs on some challenging syntactic structures.

This first series of experiments asks a simple question about NA: does my model generalize on this rule?

4.1.2 Colorless green ideas agreement

Another question that is usually asked when testing a model's behavior against NA stimuli is whether the processes supporting linguistic generalization on this task are purely syntactic, or whether semantic features also play a role. By showing that the amount of vocabulary seen during training affects the capture of the rule, [Bernardy, 2017] first show an influence of semantics to learn this ability. However, [Gulordava, 2018] showed that LSTMs trained as language models are able to succeed on NA even on meaningless sentences. They generate meaningless sentences by replacing the lexical content in the used stimuli while keeping the syntactic structure unchanged. These two findings are not contradictory however as they do not ask the same question. The first study investigates the influence of semantics during learning, while the second investigates the influence of semantics once the rule is learned. Evidence from [Gulordava, 2018] suggested NLMs can acquire grammatical competence that goes beyond meaningful lexical patterns they have seen during training on a language modeling objective. Another method that has been proposed to investigate a model's capture of the rule beyond specific lexical combination has been proposed by [Newman, 2021b], who have recently tested generalizations beyond [Marvin, 2018]'s data. They do so by extending the vocabulary at the target verb position in order to measure the degree of systematicity in finding the correct answer, considering the set of plural verbs and the set of singular verbs, and using metrics such as eq. (3.2). They show that though NLMs' top predictions are generally correct verbforms, the models still struggle on the NA task for infrequent verbs.

This second series of experiments asks a question at the algorithmic level: does my model rely only on syntax to perform well on NA, a syntactic rule?

In the following, we take a closer look at this question.

4.2 Behavioral Evaluation of Syntactic Knowledge

Linguistic theories generally assume that NA obeys two main principles: i.) **structure dependence** (SD) - NA is governed by phrasal structure, rather than surface

Struct. ID	Structure description	Example
A	Simple agreement	The **boy** **laughs** / laugh*
B	In a sentential complement	The boy knows the **girls** **play**/plays*
C	Across a prepositional phrase	The **plate** near the glasses **breaks**/break*
D	Across a subject relative clause	The **cat** that chases the mice **runs**/run*
E	In a short verb phrase coordination	The **boy** smiles and **laughs**/laugh*
F	Across an object relative clause	The **mouse** that the cats chase **runs**/run*
G	Within an object relative clause	The mouse that the **cats** **chase**/chases* runs
H	Across an object RC (*no that*)	The **mouse** the cats chase **runs**/run*
I	Within an object RC (*no that*)	The mouse the **cats** **chase**/chases* runs

Table 4.1: Agreement structures used in this study. These structures are taken from Marvin et al. [Marvin, 2018]. The cue is in blue and the target is red. For each target, we display the pair of both the correct and incorrect verb form. In structures C, D, E and H, the attractor is underlined.

linear order (i.e., the verb agrees with the syntactic subject); ii.) **meaning independence** (MI) - SVA is a morphosyntactic constraint that holds for meaningless sentences too (e.g., *Colorless green ideas sleep furiously*) [Chomsky, 1956b; Chomsky, 1971; Chomsky, 1976].

In a first experiment, we examine the extent to which BERT is able to perform lexically-independent subject-verb number agreement (NA) on targeted syntactic templates. To do so, we disrupt the lexical patterns found in naturally occurring stimuli for each targeted structure in a novel fine-grained analysis of BERT's behavior.

For example, when presenting a masked language model with sentences (3b) and (3d) (below), we mask the token at the target position, and compare the output probabilities for sleep and sleeps. The model succeeds when it assigns a higher prediction probability to the right target form.

(3) a. Colorless green ideas sleep furiously.
b. Colorless green ideas that cook the door sleep furiously.
c. *Colorless green ideas sleeps furiously.
d. *Colorless green ideas that cook the door sleeps furiously.

4.2.1 Does BERT *really* capture syntactic dependencies on colorless green sentences?

In this first series of experiments, we test BERT against Marvin et al. [Marvin, 2018]'s number agreement dataset, which comprises sets of manually crafted sentences for simple retained structures described in 4.1. In addition to testing the effect of meaningfulness by performing replacements at all positions of the sen-

tence similarly to Gulordava et al. [Gulordava, 2018], we control for the syntactic constructions from Marvin et al. [Marvin, 2018]: given a syntactic template, can BERT generalize to *any* syntactically well-formed, but meaningless sentence? If not, when does lexical content matter?

4.2.2 Datasets

We test BERT's ability to solve the NA task using three different, but complementary datasets all consisting of sentences controlled by the syntactic templates described in table. 4.1:

a) **M&L**. This is the original dataset released by Marvin et al. [Marvin, 2018], containing the syntactic constructions we use in this study. We use it to replicate Goldberg's (2019) results as a comparison point. These sentences were designed to respect semantic constraints using a limited, but semantically controlled vocabulary.

b) **WIKI**. For each template in **M&L**, we collected naturally occurring sentences from the Wikidumps used to train BERT, to test whether the model performs better on sequences of words it could have memorized during training. We extracted raw text from the Wikidumps using WikiExtractor[1], and collected sequences of word that corresponded to the sequence of POS tag for each template in **M&L**.

c) **NONCE**. For each template in **M&L**, we generated "nonce", meaningless sentences keeping the syntactic structure unaffected[2]. To do so, we replace each word in the sentence with a word of the same lexical category (and same number if applicable) using a large set of words for each POS-tag, similarly to Gulordava et al.'s (2018) stimuli. When a noun intervenes between the cue and the target (e.g., in condition C from table. 4.1), it is systematically assigned a different number from the cue, in order to test attraction effects[3]. These nonce sentences are meaningless, therefore they violate selectional restrictions contrarily to **M&L**. They also differ from [Gulordava, 2018]'s stimuli as we additionally test the effect of the syntactic construction, having separate conditions for each template. This dataset allows us to test the extent to which the model's ability to perform the agreement on nonce sentences is dependent on their syntactic structure. Each set contains 10000 sentences, with balanced proportions of singulars and plurals, making chance level at 50%.

[1] https://github.com/attardi/wikiextractor

[2] We release this data on https://github.com/karimlasri/does-bert-really-agree

[3] That is whether the model succeeds despite the presence of a distractor noun between the cue and target of the agreement.

4.2.3 Behavioral test on naturally occurring vs. nonce data

In a first experiment, we test whether the model's success over the NA task on syntactic templates from [Marvin, 2018] requires satisfying mutual semantic constraints. To do so, we compare the NA task accuracy on **M&L** and **NONCE**. We also use **WIKI** as a comparison point, to observe whether the model succeeds better on sentences it could have memorized during training than on **M&L**'s meaningful but unseen sentences.

The results from fig. 4.1 show that even though BERT is quite robust against all templates on stimuli from Marvin et al. [Marvin, 2018], it fails on some templates in **NONCE**. Little performance reduction occurs when there is no intervening attractor (A, E, G, I), that is when the cue and target are within the same clause. This shows that the model can solve the NA task in the absence of attractors, even when there is a violation of semantic selectional restrictions. The only exception is when the cue occurs in a sentential complement (B). In the absence of the complementizer *that*, the model might be perturbed by ambiguity, expecting a direct object noun (e.g., *The boy knows the mathematics lessons*). Therefore, we tested two supplementary conditions: one with the overt complementizer (B-2), and another where the verb that introduces the complementizer is constrained to be a stative verb (B-3). The results confirm our hypothesis: BERT carries out the task successfully on **NONCE** when the complementizer makes the sentence syntactically unambiguous, which also suggests that the model relies on heuristics that are partly lexicalized.

On the other templates (B, C, D, F and H) which present an attractor (that is a noun with an opposite number which can distract the model), performance drops close to chance level on **NONCE**. This means that BERT is not able to perform lexically-independent generalizations when the target and the cue are separated by a hierarchically embedded phrase containing an attractor noun. Interestingly, the model often performs better on **WIKI** than on **M&L**, which suggests that memorized lexical patterns can help solve the task in addition to being meaningful.

4.2.4 Influence of one-word replacements

In a second experiment, we measure how performance is affected when replacing words at one position at a time in the templates, on **WIKI**. Our goal is to understand whether the performance drop observed in EXP. 1 is due to the lexical content filling specific syntactic positions in our templates. In particular, this setting makes it posible to understand whether most of the effect is due to replacing the cue, the target, the attractor (if present) or words in none of those three categories.

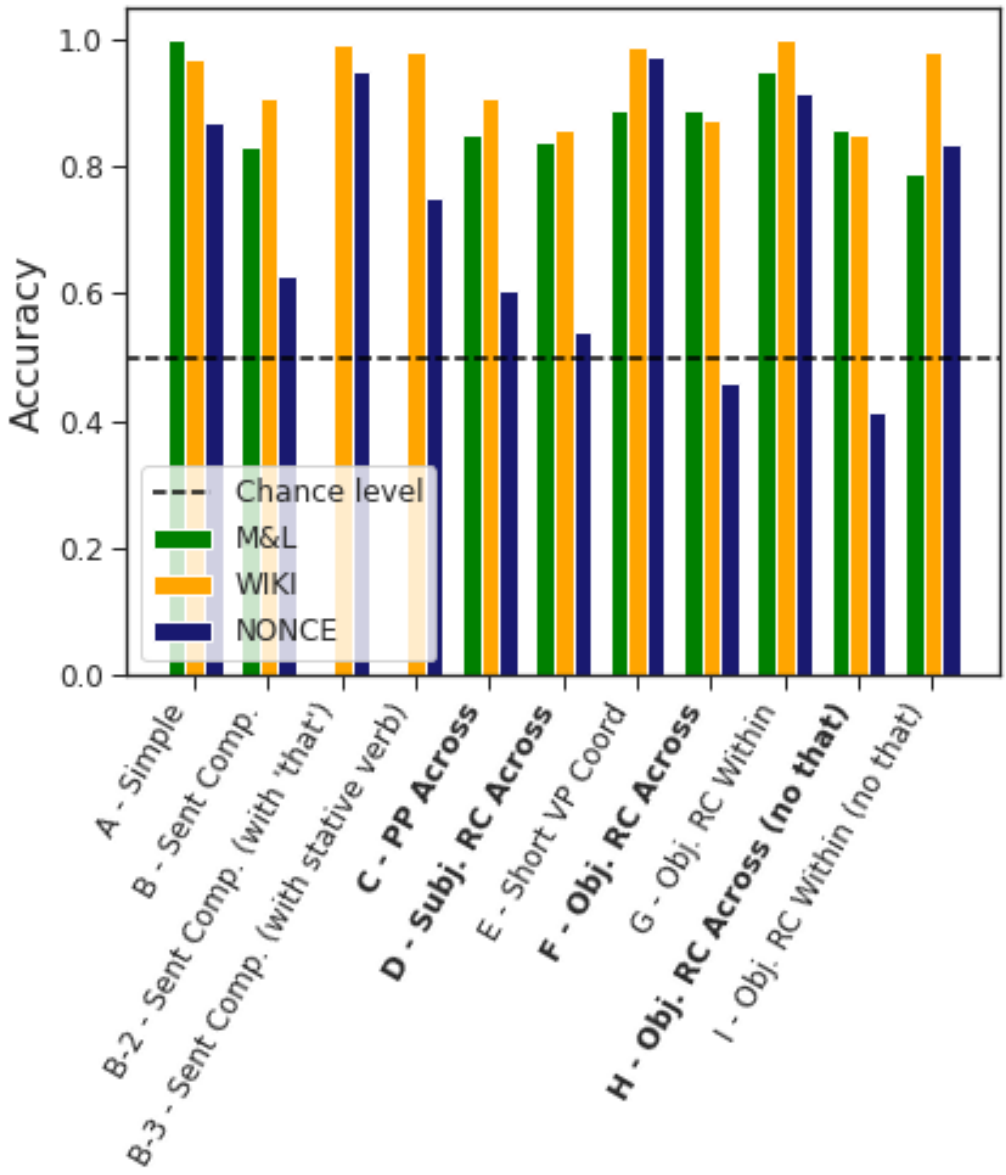


Figure 4.1: Accuracies on the number agreement task for the retained structures obtained by BERT Base. Templates where an attractor is present are displayed in bold. Note that conditions B-2 and B-3 were not present in the original **M&L** stimuli

The results in fig. 4.2 show that in sentences with no attractor (A, E, G, I), one-word replacement results in low performance drops, consistently with observations from EXP. 1. When the stimuli contain an embedded phrase containing an attractor, replacing the target itself, but also words close to the target verb (in D, F and H) can significantly harm performance. The cue is linearly distant from the target in sentences with attractors, and its replacement has little impact on performance. Replacing the attractor also has a limited impact on the task, as templates D and H show. We note a general tendency that replacing closest words results in higher performance drop than replacing farther ones, including verbs in embedded clauses.

This suggests that the model's ability to deal with attractors is not due solely to hierarchical, lexically independent generalizations acquired during training. Instead, our observations show that the model is also sensitive to the content of syntactically-independent intervening material linearly close to the target verb.

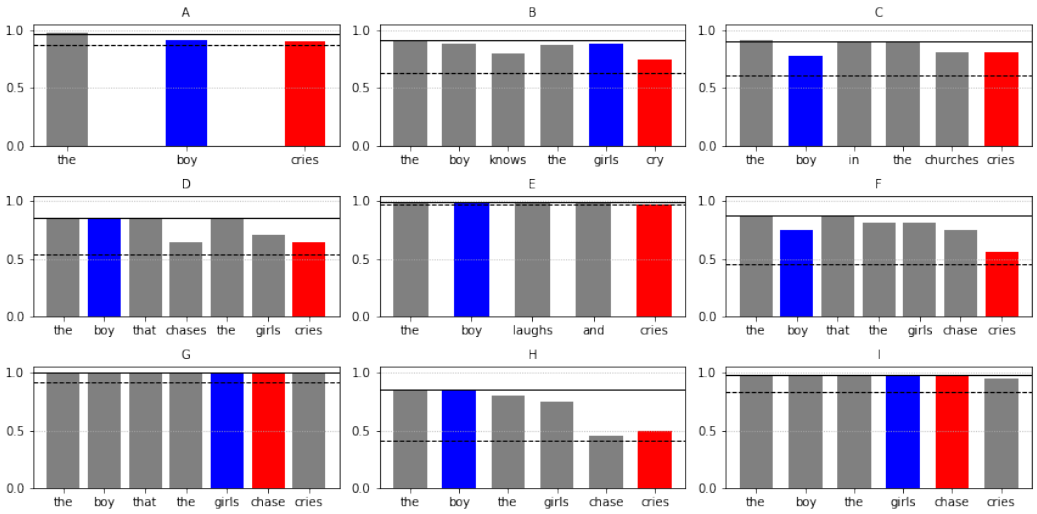


Figure 4.2: Accuracies on the NA task after one-word replacement. Each column represents the model's performance after intervening at the position exemplified by the word displayed in the x-axis. Attractors are represented in bold. Replacements are performed over sentences from **WIKI**. For each syntactic template, the performance on **WIKI** (continuous line) and **NONCE** (dashed line) is represented as a comparison point. The cue's replacement is represented in blue and the target's in red.

4.2.5 Discussion

Previous NA studies have led Baroni [Baroni, 2019] to claim that "the linguistic proficiency of neural networks extends beyond shallow pattern recognition".

Though it is undeniable that BERT does generalize beyond its input and is able to carry out the NA task on the simplest templates, our experiments also suggest that these generalizations can be lexically dependent. When naturally occurring lexical patterns are replaced with syntactically well-formed, but meaningless combinations, the model's syntactic ability seems to be heavily compromised, contrary to Goldberg [Goldberg, 2019]'s reported results on the Gulordava et al. [Gulordava, 2018] stimuli.

Moreover, most disruption is caused by replacing the words closest to the target within the embedded phrase, that in principle should not affect the agreement relation. These two facts together indicate that some of BERT's syntactic abilities are limited to specific word sequences that the model could have memorized during training, including words that are linearly close but belong to a different embedded phrase or clause. Furthermore, the fact that the model improves its performance on data it has been trained on (i.e., the **WIKI** dataset) over other meaningful, unseen sentences (i.e., the **M&L** dataset) is further evidence that at least part of its alleged generalization abilities might be just the effect of memorization.

We can surmise that the model relies on a variety of heuristics acquired during training to approximate syntactic generalizations, in line with Finlayson et al. [Finlayson, 2021], who found two distinct mechanisms to accomplish agreement in Transformer-based architectures. We find that those heuristics can therefore tend to be highly lexicalized, similarly to Newman et al. [Newman, 2021b] who showed that generalization is not systematic by testing a wide range of verbs. This is confirmed by BERT's sensitivity to the main verb when there is no overt complementizer[4], which prevents it from solving the NA task. This suggests that the model has acquired semi-lexicalized syntactic information about verb subcategorization preferences.

Although BERT's ability to approximate syntactic rules is probably more brittle than previously argued, this should not lead to rejecting its ability to learn natural language grammar. For instance, constructionist approaches [Hoffman, 2013] have argued since long against a purely abstract grammar detached from lexical meaning, despite what the data in (3) have often been claimed to prove. The alternative view is a grammar consisting of constructions that differ in their level of abstractness and lexicalization. BERT's lexically-driven behavior could therefore be consistent with this less abstract conceptions of syntax. Finally, given previous experiments [Laurinavichyute, 2022], we can speculate that humans could also similarly manifest patterns of errors driven by semantic, or lexical interferences from words linearly close to the target. Though such patterns seem to differ between language

[4]cf. sentence type B *no that*

models and humans [Linzen, 2018], this in turn leads us to questioning our expectations regarding the syntactic abilities of neural language models.

4.3 Number Agreement Error Patterns: a Comparison between BERT's Behavior and Human Judgements

Previous research has shown that humans are also prone to making number agreement errors with specific constructions [Bock, 1991; Hartsuiker, 2001], for example when an attractor is present. See (4) from Bock et al. [Bock, 1991] for an example where agreement can be disturbed by an attractor, as human subjects often show preference for a syntactically ill-formed sentence:

(4) [The **readiness**]$_{\text{subject}}$ [of our conventional $\underline{\text{forces}}_{\text{attractor}}$]$_{\text{PP}}$ [*are*]$_{\text{verb}}$ at an all-time low.

This evidence suggests that the SD principle of number agreement might be weaker than it is typically assumed and can be disrupted or disturbed under specific conditions even for humans. At the same time, such violation prompts the need to carefully test whether the MI principle of NA is also compromised in subjects' grammaticality judgments. With this in mind, we further compare the observed behavior of our model, `bert-base`, to human preference judgements, obtained with a psycholinguistic online crowd sourcing experiment. To do so, we first collect human responses on meaningful and meaningless sentence pairs featuring syntactic structures of varying complexities; then we analyze and compare the error patterns in humans and in BERT. This allows us to address the following questions: do human judgments show evidence for structure independence and meaning independence when solving the number agreement task? Do humans and NLMs make similar subject-verb agreement errors in structures with attractors and/or in meaningless sentences? Comparing the error patterns of humans and NLMs on number agreement is a crucial piece to understand our model's linguistic knowledge, as errors made by the model do not mean that the model is missing crucial aspects of linguistic competence if humans themselves are make similar errors.

4.3.1 Experimental Setup

In this section, we describe the procedure to construct the experimental items used to collect human judgments with crowd-sourcing and to test NLM behavior on NA.

Items

In this experiment, we test humans against sentences using the same syntactic structures as in the previous set of experiments, in Table 4.1, four of which present an attractor. For each syntactic template, we generate 30 meaningless sentences using the same procedure. Every minimal pair consists of sentences similar to (5). We also sample 30 meaningful sentences for each structure from Marvin et al. [Marvin, 2018] to collect human data.[5]

We use the vocabulary of Lasri et al. [Lasri, 2022] for the generation of our NONCE items and filtered it. We selected a vocabulary of nouns and verbs that checked the following criteria:

1. We filter out tokens that are ambiguous, i.e. tokens which can either be a noun or a verb. We used WordNet [Miller, 1995] implemented in the NLTK library [Bird, 2009] in python 3 to check whether a word was not classified as a noun and a verb by checking whether there was no synset in the other category.

2. We filter out words using their relative frequency measured by using the python library wordfreq [Speer, 2018].[6] We choose to filter too frequent words because some were ambiguous with another category (*e.g.* in the noun vocabulary we can find *good, well, one*). We decided to remove infrequent words to prevent that participants would not know their meaning. For example, this filtered out *polynomial*, and *consonant* from the noun vocabulary.

3. We only keep words with a length ranging from 3 and 8 characters, in order to prevent big differences in size between the items produced by one template.

4. We make a subdivision between transitive and intransitive verbs in order to correctly fill the templates.

(5) a. *The admissions sings.
b. The admissions sing.

We thus collect human performance on 30 items for each of our 2 conditions (Nonce and M&L), and each of our 9 syntactic structures, for a total of 540 items.

[5]While the number of items is reduced in this experiment, we still call these data sources **NONCE M&L**. We filter out sentences where the target verb is 'be', as this verb is very frequent in English which might influence results

[6]https://pypi.org/project/wordfreq/

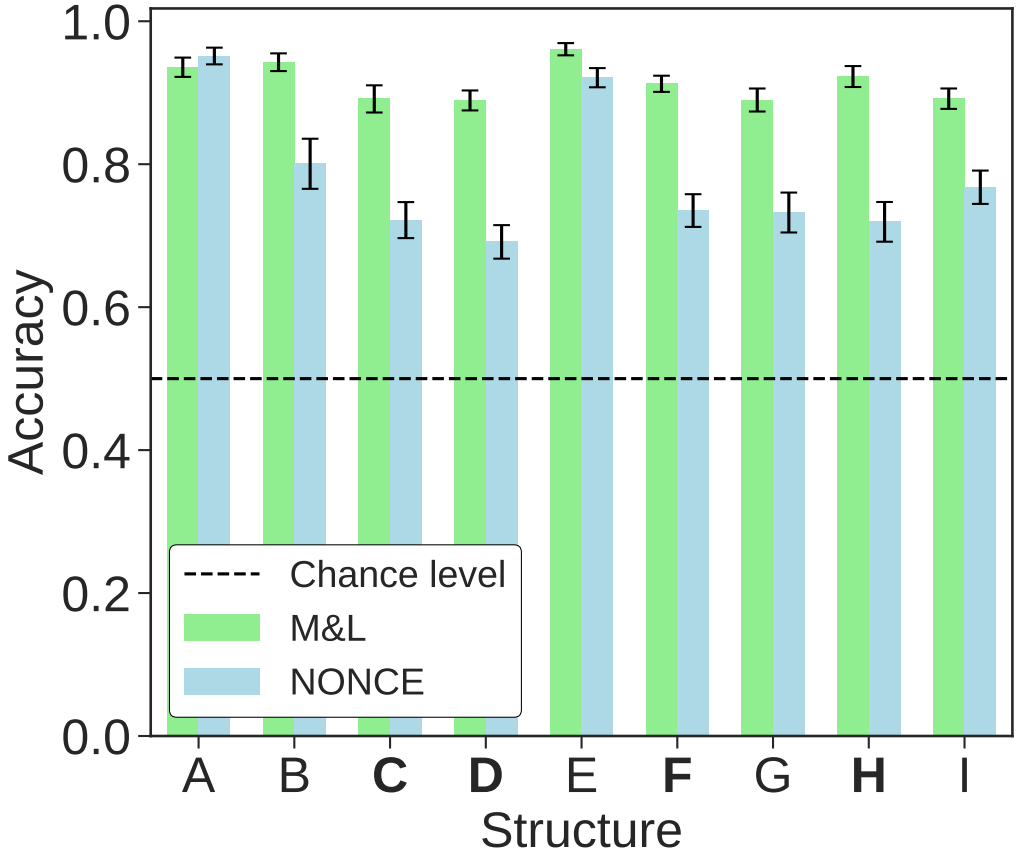
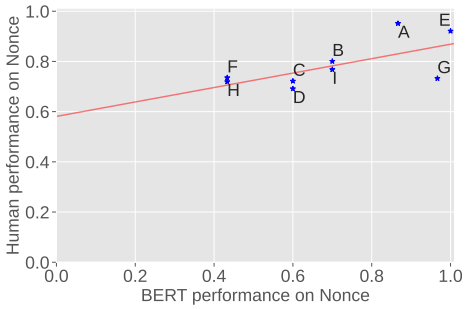
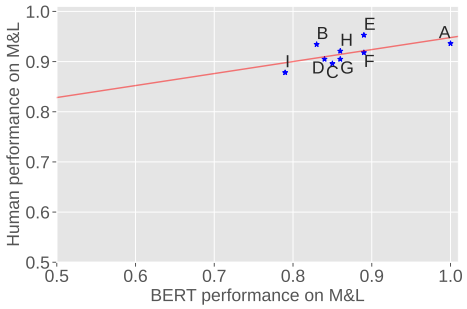


Figure 4.3: Human accuracies on the NA task. Structures where an attractor is present are displayed in bold.



(a) Human nonce performance vs. BERT



(b) Human M&L performance vs. BERT

Figure 4.4: A comparison of human performance against BERT's performance on each of our structures.

Collection of Human Judgements

We collect our human data using the online click working platform Prolific.[7] We implemented a binary choice experiment in Psychopy [Peirce, 2007] hosted on Pavlovia[8] where participants were presented with a minimal pair, such as in (5), and asked which sentence was the most correct. In order to prevent habituation to our stimuli and task, we used 64% of filler items. We recruited 300 participants to obtain 20 responses per item and kept the responses of 270 participants in our final data set. Their mean speed to judge one item was 6.9 seconds.

Setup To collect our human judgements, we recruited participants on the working platform Prolific. Participants were redirected to the Pavlovia page hosting our experiment. First, they had to give their informed consent. Their data was processed in accordance to the European General Data Protection Regulation [Commission, 2018], and no sensitive data has been collected. After being informed, participants were shown brief instructions about the forced choice task. For each item, they were presented with two sentences, and asked to select the one that seemed more acceptable using the keyboard arrows. Each session started with three training items followed by feedback. When the training was finished, they were notified that the experiment started and that they would not receive feedback anymore. Each participant was presented with 100 items and thereafter received a message that the experiment was over.

Number of Items and Participants In total, the participants replied to 100 items: 64 fillers, and 36 experimental items. 18 were from the nonsense condition and 18 from the M&L data set. As every condition features 9 different structures, 2 structures of each category where shown per participant. In order to collect 20 responses per item, we recruited 300 participants with 15 different versions of the online experiment, which can be found in the supplement material of this article.

Fillers Our filler items where from Ettinger [Ettinger, 2020]. They feature semantically appropriate and inappropriate completions. We also used filler items with correct and incorrect determiners among 'a/an' depending on the following noun to feature syntax-oriented fillers as well.

Selection of Participants We only accepted participants with the United States nationality with English as a first language between the age 18 and 60 years old. We

[7] http://prolific.co
[8] https://pavlovia.org

excluded participants that already contributed to another version of the experiment.

Reward Participants got rewarded 2.25£ for a participation that was estimated to take 15 minutes. This was estimated to be a 'good' hourly pay by the Prolific platform. We rejected participants that performed the experiment in less than 5 minutes.

Filtering and Loss of Participants In our final data set we have the contributions of 270 participants. Participants that performed at chance level (50 % accuracy) were filtered out. Furthermore, we lost some data of participants that did not close the experiment correctly in Pavlovia.

4.3.2 Results

We first analyze our collected human judgements, which we then compare to BERT's performance.

Error patterns in humans In this analysis, we compare the human accuracies on the nonce stimuli and on Marvin et al.'s (2018) sentences. In fig. 4.3, we break down the results by syntactic structure to observe whether the construction type affects the human judgments. We notice a performance drop in all structures with nonce sentences, except for A where the apparent increase is not significant, as shown by the error bars. Interestingly, the structures for which an attractor is present (bolded in the *x*-axis) are those for which the performance drop seems to be the highest. We also observe high performance drops in sentences where there is no attractor (B, G and I). Looking at table. 4.1, we can see that these structures are more complex than the structures where the effect of meaningfulness is low (A and E). Indeed, they contain either a complement (B), or a relative clause (G, I). Surprisingly, we observe a similar pattern on meaningless sentences in (F) and (G): Humans seem to be perturbed as much by the attractor within the object relative clause (F), as they are by the material in the main clause (G), if sentences are meaningless. This evidence in comprehension seems opposite to Bock et al.'s (1992) claim that agreement production is only sensitive to information within the clause of the target. This evidence hints at the possibility of a difference in the mechanisms that support NA in production and comprehension.

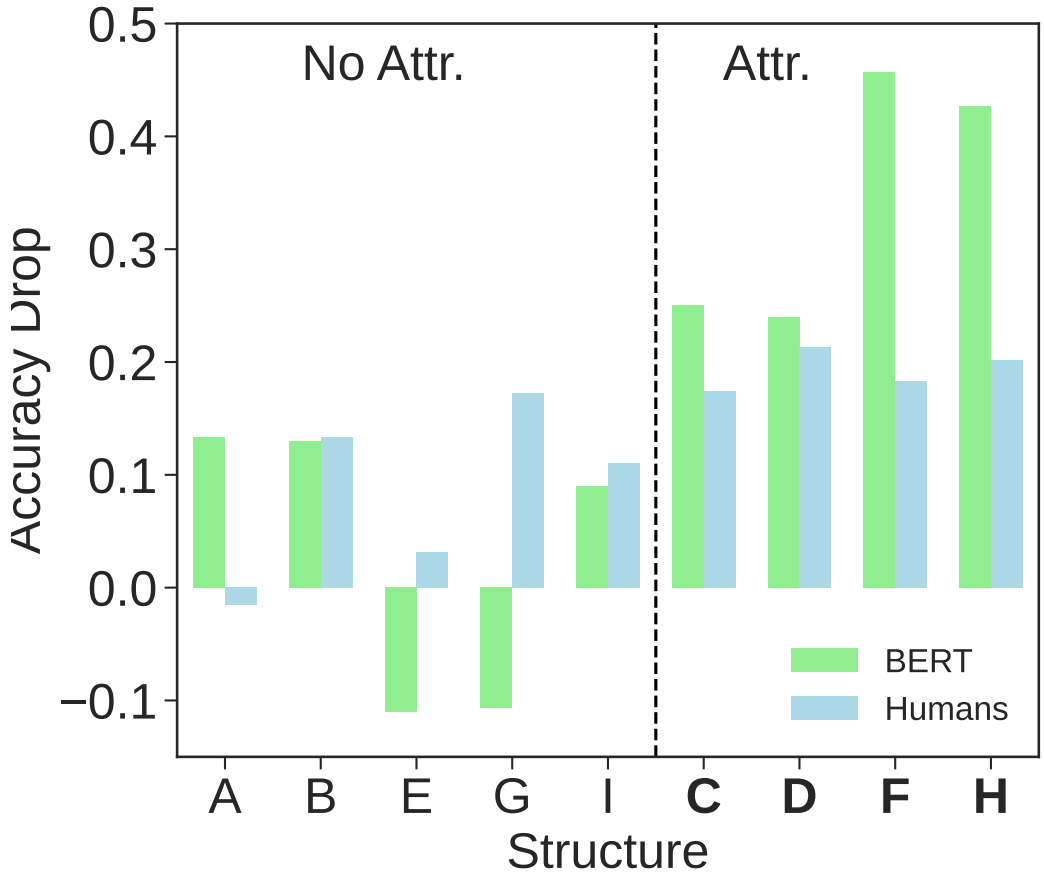


Figure 4.5: Performance drops between M&L and nonce stimuli.

Metric	Correlation
M&L Accuracy	0.61
Nonce Accuracy	0.65
Accuracy Drop	0.52

Table 4.2: Coefficient of determination between BERT's and human performance on NA, averaged across syntactic structures. The accuracy drop condition represents the difference between average performance on M&L's stimuli and our nonce stimuli, as seen in fig. 4.5

4.3.3 How similar are error patterns in humans and BERT?

In this analysis, we compare the performance achieved by BERT with the human performance, on each of our stimuli types. fig. 4.4 displays the result obtained by humans against BERT's performance for each syntactic template, for both meaningful and meaningless sentences. Interestingly, there seems to be a fairly high alignment between the results for each syntactic construction, and for each source of stimuli. We display the R^2 correlation measurement of our fit in table. 4.2. The latter confirms the observed alignment, as we obtain quite high correlations (0.61 for meaningful sentences and 0.65 for nonce sentences). This observation aligns well with that of [Lau, 2020], that bidirectional transformer models can predict well acceptability judgements from humans. However, we observe that while the variation in performance obtained by humans across templates seems quite low, BERT's performance does seem to be more affected by the different structures. This is especially true in the case of nonce sentences, as seen in fig. 4.4a. We also observe a difference in performance decrease in fig. 4.5, as BERT's performance drops are overall higher in presence of an attractor compared to those of humans. On the other hand, BERT has a higher performance drop on (A) and humans on (G). This in turn could be explained by the fact that (G) is a hard sentence to process for humans, the target of the agreement being within an embedded relative clause, while BERT could rely on local context in this case as observed by Lasri et al. [Lasri, 2022].

4.4 Discussion

4.4.1 Lexicalization and syntactic generalization

While subject-verb agreement is sometimes considered as a purely syntactic phenomenon, our results show that actually humans also rely on semantics, which goes against the meaning independence hypothesis. Our results also show that

BERT is also highly dependent on semantics, a finding in line with Bernardy et al. [Bernardy, 2017], who mention that "*deep neural networks require large vocabularies to form substantive lexical embeddings in order to learn structural patterns*". This highlights the strong connection between the ability to process linguistic structure and the semantic content of sentences.

4.4.2 Structure dependence

Throughout this study, we observed that the performance of both humans and BERT were sensitive to the syntactic structure used in our items. Humans clearly obtain lower performance on sentences that are more complex to process when they are meaningless, including but not limited to sentences presenting an attractor. This variation in performance seems to reflect variation in structure complexity, which upholds SD. On the other hand, BERT seems to be mostly sensitive to sentences with attractors. This evidence rather shows a violation of SD, as attractors are only related to the target by linear order, in line with evidence found by Lasri et al. [Lasri, 2022]. While human and BERT's results seem to correlate to a large extent, these divergences could reflect a difference in processing. For instance, NA in sentence comprehension for humans could depend on having read the whole sentence, while BERT could rely more on local context for this task. Indeed, a fine-grained analysis performed in previous work showed BERT to be mostly sensitive to the replacement of linearly close tokens [Lasri, 2022].

4.5 Conclusion

Using a behavioral syntactic task, subject-verb number agreement, we were able to demonstrate that straightforward hypotheses can be tested regarding the nature of the knowledge acquired by a NLM on this task. In this series of experiments, we addressed two independent questions. The first question was addressing the algorithmic level of a NLM's linguistic knowledge on the NA task, and dealt with the nature of the information processed by the model to perform this task. As its description relies purely on the syntactic structure of input sentences, we investigated whether the computations supporting successes relied purely on information about the input's syntactic structure, independently of the meaning of its lexical items. Even though our tested model performs excellently on meaningful sentences, regardless of their structure, we have shown that the ability to perform NA is highly dependent on the syntactic construction when meaningfulness is disrupted. This informs us on the processes underlying generalization on the number agreement task: the model is not extrapolating the agreement rule to sentences which are deprived from meaning, suggesting that the implementation of this rule is semantically-dependent.

We then asked a second question, as this evidence contradicts the independence between syntactic and semantic knowledge hypothesized in some linguistic theories: does this observation hold for humans too? Our careful comparison with humans showed similarities between their mistakes and BERT's errors. In particular, sentences with attractors tend to compromise meaning independence when processing the agreement relation. Given this observation, we can rule out that number agreement, as a syntactic rule, is processed using knowledge of syntax only. This puts into question the separation between syntactic and semantic processes supporting linguistic competence.

Despite these similarities, we further find some differences in failures, as the performance drop is generally higher in BERT on meaningless sentences, and that humans are more perturbed by complex constructions without an attractor. This finding can in turn reflect differences in processing syntactic structure which could be the source of the partial mismatch in the observed error patterns, such as more reliance on local context for BERT, suggestive of more fragile heuristics.

Figure 4.6: "A research scientist examining the linguistic abilities of an artificial intelligence, digital art", K.L. x DALL·E 2

Chapter 5

Behavioral Evaluation of Language Models and Word Order Usage

Goals

In the previous chapter, we performed behavioral analyses to investigate the nature of syntactic abilities captured by a neural language model. In particular, using controlled datasets, we could test various hypotheses on the nature of the information processed by the model to demonstrate abilities on a syntactic task. Such analysis allow us to rule out certain hypotheses for the nature of the computations underlying the targeted ability – in particular meaning-independance on a syntactic task – using the model's failures. This evidence limits our understanding of the computations supporting successes: when the model is able to respect certain constraints, we know little about the information it uses. To demonstrate this, we leverage linguistically motivated behavioral tests in this chapter to explore the limits of behavioral approximation in expressing the nature of the computations supporting predictions. When models seem to capture a given linguistic phenomenon, one does not know how the model comes to making the correct decision. In turn, we test an alternative hypothesis to contrast the linguistic interpretation of our model's behavior. Specifically, we test how the behavior of our NLM compares to scores assigned by a model deprived of word-order, on two tasks which require order information. While it is plausible that correct predictions point at the capture of linguistic knowledge, it is equally possible that the model relies on statistical co-occurrences alone to mimic such abilities.

Contents

5.1 Introduction

As discussed earlier, one of the main limitations of behavioral analyses is that they only inform us regarding whether the model is capable of giving certain answers in certain contexts. How the model makes such decisions remains opaque in most studies examining output behavior. As mentioned earlier in this thesis, one of the reasons why we probe language models is because their capacity to solve downstream tasks is opaque, and they might rely on heuristics to solve such high-level tasks. Examples of studies showing the model is relying on heuristics without really learning the task [McCoy, 2019] are a good source of inspiration to know more about the model's true abilities. For instance, in our behavioral tests, we can also design hypotheses for how tasks are solved – which information is used and which is not. In our previous experiments, we employed a strategy similar to this by breaking meaningfulness in input sentences, to see if the model performed number agreement by relying solely on its input's syntactic structure. While we previously intervened on the distribution of input sentences, we can also test hypotheses by comparing the model's predictions to alternative signals at the output level. In this chapter, we employ this technique to test whether the output behavior of the model can be accounted for using a simple co-occurrence based model, deprived from word-order information. In recent work, authors questioned whether BERT's ability in capturing linguistic properties, which should ideally consist of abstract generalizations, is actually a reflection of shallow properties of the training corpus, such as the frequency of words or combinations [Yu, 2020a; Newman, 2021b]. These studies investigate the relation between statistics extracted from the training corpus and the models' linguistic competence.

In a recent study [Yu, 2020b], authors measured the correlation between the ability of transformers models to solve a suite of syntactic tasks and the training corpus frequency of nouns in the input stimuli, but they did not find significant correlation. These findings differ from another study [Wei, 2021a], in which increasing the frequency of a verb in BERT training corpus generally led to improvement in the ability of BERT to solve the NA task involving that verb. The latter study builds on a line of research studying the impact of the training corpus (in particular the training data size and distribution) on a neural language model's ability to capture linguistic generalizations [Warstadt, 2020c; Lovering, 2021].

5.2 Testing concurrent hypotheses over the model's behavior

As seen previously, NLMs can be evaluated either against hypotheses derived from linguistic theory – i.e. rules which constrain the output probabilities, or by comparison to a set of completion scores. In the latter case, we have seen previously that the model's predictions can be compared to human preferences. In this section, we test several hypotheses regarding the way BERT's predictions are ranked in its cloze setting. Specifically, we investigate whether its predictions approximate rule-based constraints using a syntactic task, and human judgements on a selectional preferences task. In addition to comparing the model's behavior to constraints from linguistic theory and human scores, we also test whether predictions are concordant with purely distributional signal, deprived from word order.

The notion of selectional constraint or restriction is based on the idea that each predicate comes with a semantically coherent set of typical and plausible arguments [Katz, 1963]. For example, subjects of the verbs *smile* and *love* are mostly animate. In computational linguistics, the capture of this knowledge is called selectional preferences acquisition (SP), and models are evaluated against sets of predicate-argument-syntactic relation triplets with different degrees of typicality. [Metheniti, 2020] used the SP10K dataset (see 5.3.1) as a benchmark to test if BERT has acquired SP. For each head-syntactic relation-dependent triplet in the dataset, they collected a set of sentences where the head and the dependent occur in the relation. They masked the dependent in each sentence and computed the correlation between the SP score and the average of the probabilities assigned by BERT to the dependents of the collected sentences, claiming that the model captures selectional preferences.

To contrast our analyses, we examine to which extent BERT's completions only mirror order-free co-occurrences seen in its training corpus. Our experiments show that BERT's output aligns well with both rule-based syntactic constraints and human-like judgement of selectional preferences. Additionally, we find that they are often equally approximated by co-occurrence statistics that do not capture word order. Finally, we find evidence that the model's abilities extend beyond order-free co-occurrence statistics in a subset of conditions analyzed in this study.

5.3 Experimental setting

As mentioned above, we diagnose the behavior of our language model using two tasks, a syntactic completion task, and a selectional preference task. In each case,

we use templates to generate sentences and mask a token at a given position. On each task, we then test two concurrent hypotheses over the model's output probabilities. On the one hand, we test whether the model succeeds on the task, that is whether it captures a normative rule-based constraint on our syntactic task, and whether it aligns well with empirically grounded human judgements on the second. On the other hand, we test whether the degree to which it is approximating the adequate behavior can be accounted for by a purely distributional baseline, using a co-occurrence based model, *a priori* unable to capture structural properties of input sentences. This allows us to challenge our observations regarding how the model behaves in its normal training setting, masked language modeling.

5.3.1 Tasks and Hypotheses

Our two tasks involve different hypotheses that can be evaluated based on the model's behavior in its normal training setting - masked language modeling. They both allow us to test whether the model's behavior reflects knowledge about constraints imposed by the context over the completion, which in principle shouldn't be reduced to the memorization of statistical co-occurrences deprived from word order information. Our chosen experiments are complementary in the sense that one addresses a syntactic phenomenon and the other a semantic phenomenon, both requiring information about word-order to be performed.

Syntactic completion For our first experiment, we test a rule-based syntactic prediction hypothesis. To do so, we generate sentences using syntactic templates, as illustrated in Table 5.1. For each construction, we fill the positions of content words with randomly chosen words, using a predefined vocabulary. This generation procedure allows us to build a dataset of syntactically well-formed, nonce sentences, with the aim of isolating the behavioral syntactic abilities of the model. In particular, we investigate the model's ability to recover three parts-of-speech (POS): nouns, verbs and adjectives.[1] In this regard, we test the following hypothesis:

H1-a: Rule-based hypothesis. *The model's behavior reflects the acquisition of linguistically plausible symbolic rules.*

Selectional Preferences In our second experiment, we test a human-like prediction hypothesis on a selectional preferences task, using the **SP10K** dataset [Zhang, 2019]. This dataset is a collection of pairs of words, where each pair is assigned a score from 1 to 10, expressing how much annotators perceive the use of the two

[1] In the case of ambiguous tokens (e.g. "cook" can be a verb and also a noun), we include such tokens in the set of the expected POS (e.g., verb) but not the others (e.g., noun).

Struct. ID	Structure name	Structure description	Example
A	Simple	Det Noun Verb	*The car cooks*
B	Simple with Adj.	Det Adj Noun Verb	*The violent car cooks*
C	Comp.	Det Noun Verb Det Noun	*The car cooks the air*
D	Adj. and Comp.	Det Adj Noun Verb Noun	*The violent car cooks the air*
E	Comp. with Adj.	Det Noun Verb Adj Noun	*The car cooks the brown air*
F	Adj. and Comp. with Adj.	Det Adj Noun Verb Det Adj Noun	*The violent car cooks the brown air*

Table 5.1: Syntactic templates used in this study for our syntactic completion task.

words in a given syntactic relation as plausible. As an example, the *eat-meal* pair in **dobj** relation has been assigned a score s=10 by humans. Scores were obtained by averaging judgments from crowdworkers. 5 relations are considered, and for each, the dataset contains 2,000 word pairs. We only use the **nsubj** relation since it is most suitable for our cloze task design. Given a word pair, we generate sentences using templates that are described in section 5.4.2. We then investigate the probabilities that the model assigns at to different arguments after masking the sentence. Using our human judgements of selectional preferences, we test the following hypothesis:

H1-b: Human-like hypothesis. *The model's behavior shows preferences similar to those of humans*. Note that **H1-a** and **H1-b** are not tested concurrently, as they are tested each on a different experiment.

5.3.2 Co-occurrence extraction from training corpora

In addition to each test presented above, and for each task, we evaluate whether order-free distributional information extracted from the training corpora can account for the model's behavior. To do so, we count co-occurrences in the Wikipedia dumps[2] and the BookCorpus[3], used to train the original BERT model. We leverage the extracted information to obtain the following measures: i.) *Co-occurrence of word pairs*; ii.) *Frequency count of single words*, using sentences as context windows. As such measures do not take word order into account, they shouldn't in principle capture syntactic structure or argument structure. Though they do not contain word order information, they contain a vast amount of distributional information the model has seen during training, as they are extracted from the training corpus. It is worth noticing that the BookCorpus has only rarely been taken into account in studies that investigate BERT's training corpus, as it was not public until recently [Bandy, 2021]. This collection of co-occurrences allows us to test the following hypothesis:

[2]Extracted using https://github.com/attardi/wikiextractor
[3]https://huggingface.co/datasets/bookcorpus

H2: Order-free co-occurrence hypothesis. *The model's behavior is driven by surface order-free co-occurrence statistics from its training corpora.*

The baseline model used for each experiment is described below.

5.3.3 Models

Neural Language Model As in previous experiments, we analyze BERT [Devlin, 2018], a bidirectional transformer-based language model with 12 layers. We use the `bert-base-uncased` version in our experiments. As our goal is to understand the model's behavior, we test it in its normal functioning setting, masked language modeling. Specifically, to evaluate our different hypotheses, for each task, we feed it with templatically-constructed templates and extract the probabilities assigned by the model to each word from its vocabulary at the masked position, and apply our analyses to the output probability vector at this position.

Co-occurrence based model In each experiment, we test (H2) by building a co-occurrence based model and measuring the extent to which it accounts for our NLM's predictions on each task.

Our first experiment consists in measuring whether the model assigns higher probability scores to tokens with the right lexical category, so we are interested in the ordering of our NLM's scores on a large vocabulary selected for this experiment. To build our co-occurrence model, we need to score each token from this vocabulary when given a context. We rely on the matrix storing co-occurrences for word pairs in the training corpus, described in section 5.3.2. Each row of this matrix assigns a co-occurrence vector to each word from the vocabulary. For each masked sentence, we first compute a co-occurrence vector representing the context by averaging rows corresponding to words appearing in the context. Then, we compute co-occurrence based prediction scores at the masked position by simply computing the similarity between the context vector and each token's co-occurrence vector. This procedure is detailed in section 5.4.1. Scores assigned by our co-occurrence model can further be compared to those assigned by BERT using an *ordering-based* evaluation method, as imposed by our task.

In our second experiment, for each templatically-constructed sentence, we have a unique score assigned by humans to a given argument, associated with the verb present in the context. Our goal in this task is to predict log-probabilities assigned by BERT to the completion scored by humans, based on their selectional preference judgements. In this case, we test whether different co-occurrence based models based on various corpus-based statistics can also predict BERT's probability. The

first one is based on the verb-argument pair's co-occurrence frequency for each example, as described above. The second one is based on the log frequency of the argument only in the training corpus, and the third one is based on both variables. This is detailed in section 5.4.2.

5.4 Experiments

5.4.1 Exp. 1 - Syntactic Completions

We first test BERT against our syntactic completion task using generated sentences. We first present our cloze task design, and then introduce two complementary measurements to test our different hypotheses.

Cloze task design We generate 1,000 instances for each of the templates in table. 5.1 using words randomly sampled using a dictionary extracted from BERT's training corpus, where each token is labeled with its part-of-speech.[4] We only keep only words that appear as a single token in BERT's vocabulary in our dictionary. We denote our resulting vocabulary $\mathcal{V}_{synt}$. We then mask one word at a time in each sentence to test the model, and extract BERT's probability distribution over predictions at the masked position. For example, applying this procedure to structure A – *Det Noun Verb* can result in the input sentence *"The [MASK] eats ."*, for which we collect the logit vector at the masked position.

(a) Examining BERT's ordering of probabilities In this experiment, we analyze the ordering of BERT's predictions. Given a masked sentence, we analyze the probabilities assigned by BERT at the masked position. We thus get, for each masked context c:

$$\forall w \in \mathcal{V}_{synt}, p_w^c = p(w|c)$$

Our mapping between each word and its part-of-speech defines a ground truth:

$$l_w^c \in \{0, 1\}$$

where the binary feature indicates whether w is part of the expected lexical category. We further analyze the ranking of output probabilities $(p_w^c)_{w \in \mathcal{V}_{synt}}$. To test **H1-a**, we rank all words in $\mathcal{V}_{synt}$ such that $\{w|l_w^c = 1\}$ occupy the first positions and $\{w|l_w^c = 0\}$ occupy the next ones. We then compare this reference to the ordering of probabilities generated by BERT using a modified version of Kendall's

[4] We filter out ambiguous tokens

Correlation Ranking:[5]

$$\tau = \frac{n_c - n_d}{n_0 - n_1} \tag{5.1}$$

On the one hand, we compute this metric over BERT's predictions and the linguistic reference as described above, to test **H1-a**, and BERT's ability to systematically assign a higher rank to all words with the correct part-of-speech.

On the other hand, we compare the ordering of BERT's predictions with a corpus-based co-occurrence ordering for each word $w \in \mathcal{V}_{synt}$ to test **H2**. Given the context c and the co-occurrence count matrix M, we first compute the context's mean co-occurrence vector representation:

$$\mathbf{v}^c = \frac{1}{|c|} \sum_{w \in c} M_{w,*}$$

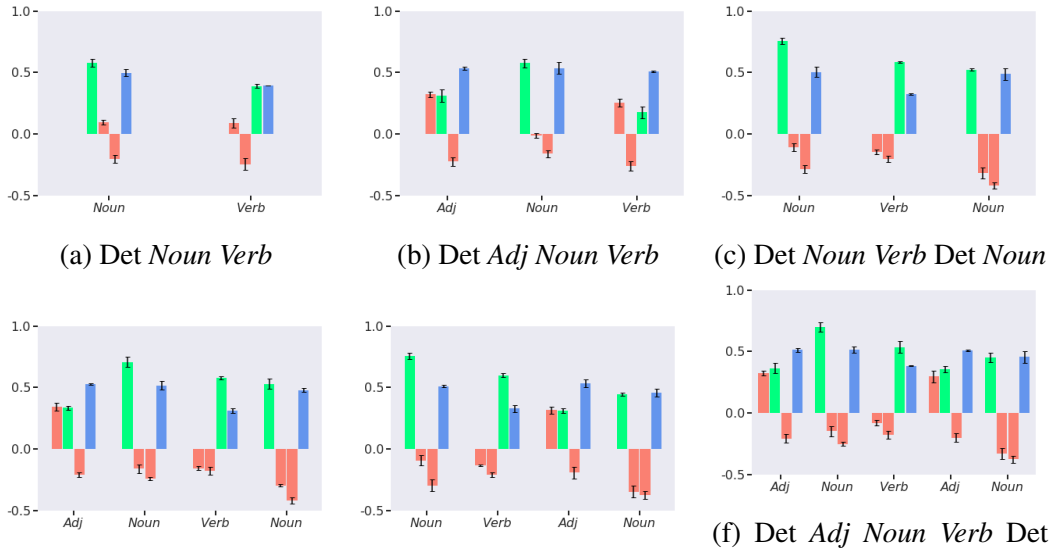
where $M_{w,*}$ is w's co-occurrence vector. After obtaining v^c, we can compute each completion's similarity to this vector. We thus get a metric that captures the distributional similarity of each completion to the context, defining an ordering over words in $\mathcal{V}_{synt}$:

$$\forall w \in \mathcal{V}_{synt}, m_w^c = \langle M_{w,*}, \mathbf{v}^c \rangle$$

This ordering and that of BERT's are further compared using Kendall's correlation coefficient presented above.

(b) Measuring the overlap between syntactic and co-occurrence based completions As our hypotheses **H1-a** and **H2** are not exclusive, we further test the extent to which both our reference orderings – defined the syntactic ground truth and the co-occurrence metric – are favoring the same words in context. To do so, we simply examine BERT's top 100 predictions. We compute the proportion of words with the right part-of-speech in this set. Denoting k the number of such words, we also compute the coverage of k closest co-occurrences in these top 100 predictions. Finally, we compute the overlap between words covered by the former portion and the latter, to see if co-occurrences alone seem to account for BERT's correct predictions.

[5]This score takes ties into account, which is convenient in our case as all correct completions occupy the same position, and all incorrect completions as well. n_c is the number of concordant pairs between the two orderings, n_d is the number of discordant pairs, n_0 is the total number of pairs and n_1 is the number of ties for our linguistic reference. We do not count ties in n_c nor in n_d. It follows that if BERT's ordering follows our linguistic reference, this coefficient is equal to 1, as $n_c = n_0 - n_1$ and $n_d = 0$. In the worst case, this coefficient is equal to -1 as $n_d = n_0 - n_1$ and $n_c = 0$.



(a) Det *Noun Verb*

(b) Det *Adj Noun Verb*

(c) Det *Noun Verb* Det *Noun*

(d) Det *Adj Noun Verb Noun*

(e) Det *Noun Verb Adj Noun*

(f) Det *Adj Noun Verb* Det *Adj Noun*

Figure 5.1: Kendall's Tau correlation coefficient (y-axis) for the syntactic completion experiment. The x-axis represents the sequence of part-of-speech for content words in each template. The green bars represent the expected part-of-speech while the red bars represent the incorrect part-of-speech (from left to right, red and green bars represent nouns, adjectives and verbs). The blue bar in turn represents the correlation coefficient computed with the co-occurrence-based similarity of each completion to the context.

5.4.2 Exp. 2 - Selectional preference

Cloze task design For each subject-verb combination in the **nsubj** set of SP10K we generate a sentence using the template "**The *subject verb* (present perfect) .**" (e.g., *The team has lost* .) We do not test relations other than **nsubj** (e.g., **dobj**, **amod**) as they require completing the word-pair with other content words to form meaningful sentences (e.g., specifying the subject), which could add confounds to our analysis.

We filter out combinations such that at least one word is not part of BERT's vocabulary as a single token, thereby resulting in 1,961 word pairs. We further mask the subject in the generated sentence (e.g., *The team has lost* . $\rightarrow$ *The [MASK] has lost* .) and analyze the probability that BERT assigns to the masked token (we write such probability vector $\mathbf{p_{subj}}$)

Separate hypothesis testing We first measure the extent to which BERT's behavior upholds **H1-b** and **H2**.

To test such hypotheses, we built several linear regression models and try to predict probabilities from p_{subj} using different variables. The models differ with respect to the predictors used: θ_{H1_b} uses the human typicality annotations in SP10K

(**h**), θ_{H2} uses the log frequency of the argument ($\mathbf{f_{arg}}$) and the log co-occurrence frequency of the argument-verb combinations ($\mathbf{f_{comb}}$) in BERT training corpus.[6] We quantify the extent to which each hypothesis is supported by the experiment as a function of the output of a measure of fit of the corresponding model (θ_{H1_b} for H1-b and θ_{H2} for H2). As a measure of model fit, we use multiple R-squared (the output ranges from 0 - no fit - to 1).

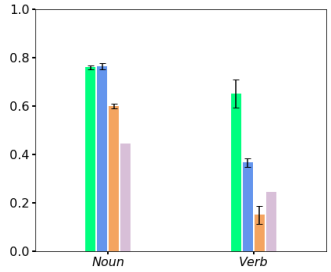
5.5 Results

5.5.1 Syntactic completions

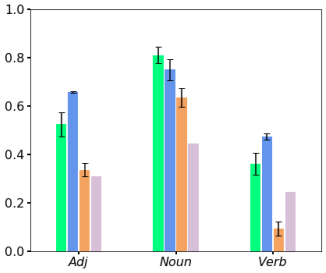
(a) Ordering of probabilities The Kendall's ranking correlation results for our first experiment are displayed in fig. 5.1. In this plot, green and red bars represent the correlation between BERT's ordering of predictions, and the ordering imposed by a rule-based predictor which would rank first all words of one part-of-speech among nouns, adjectives and verbs (represented in this order in the plot). Green is for the expected POS while red is for incorrect POS. The blue bar in turn represents the correlation with the ordering derived from our co-occurrence based predictor.

The main observation is that BERT's predictions produce a higher correlation with syntactic rule-based ordering (as the green bars show) rather than the ordering based on our co-occurrence predictor (blue bars) in almost all cases. While the opposite seems to occur, this is only the case at adjective positions where the correlation coefficient is comparable to that of nouns. In that case, the syntactic completion is ambiguous, which explains the low score using only adjectives. In structure B as well, the model seems to fill the last position with nouns, which can be a correct local completion (e.g., *The blue car keys*). This means that the model does not systematically expect sentences containing a verb. This explanation seems plausible given that noun phrases are frequent as section titles in the Wikipedia corpus, and verbless sentences could also be present in the BookCorpus. From this perspective, it seems that in most cases, BERT's behavior hints at the capture of syntactic knowledge, but in many cases performance is not significantly higher than that of a word-order blind co-occurrence predictor. This has two implications: (i) BERT's behavior partly obeys rule-based syntactic constraints, (ii) knowledge of these constraints is imperfect, and sometimes comparable to that of an order-agnostic predictor. As the two correlation measures are sometimes close, we want to determine whether the co-occurrence based predictor captures syntac-

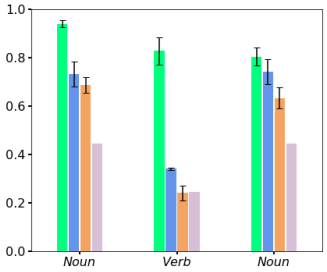
[6] The assumption of linearity between the predictors and $\mathbf{p_{subj}}$ is reasonable for all the predictors, as can be seen from the relations between variables shown in fig. 5.3



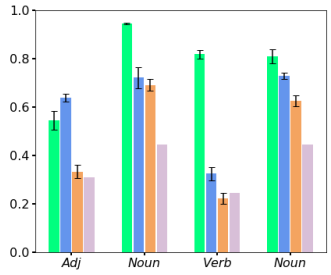
(a) Det *Noun Verb*



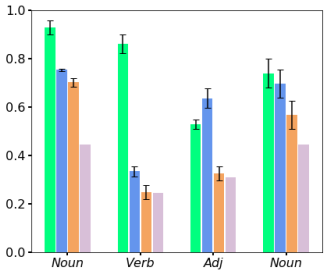
(b) Det *Adj Noun Verb*



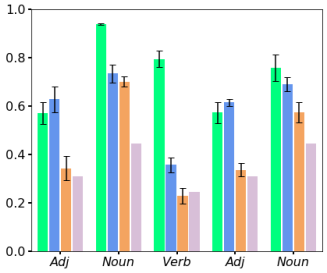
(c) Det *Noun Verb* Det *Noun*



(d) Det *Adj Noun Verb Noun*



(e) Det *Noun Verb Adj Noun*



(f) Det *Adj Noun Verb* Det *Adj Noun*

Figure 5.2: Proportion of tokens from each set under investigation (syntactically correct, and closest co-occurrences) in BERT's top 100 predictions at each position of our templates. We display the proportion of syntactically correct completions (green bars), the proportion of k closest co-occurrence completions, given that k is the total number of correct completions (blue bars), the overlap between these two portions (orange bars). and a POS-dependent baseline measuring coverage of top 100 predictions when the ordering is random (grey bars).

tically correct predictions, which brings us to the next experiment.

(b) Overlap in coverage of top 100 predictions We then evaluate the extent to which similar correlations between orderings previously analyzed reflect an overlap in orderings. To do so, we examined the coverage of top 100 predictions of the model in our cloze setting. fig. 5.2 presents our analysis of the model's completions for each of our selected syntactic templates. We see that, accordingly to the previous results, looking at the top 100 predictions points towards the model's capture of aspects of syntactic generalization overall. Indeed, the top predictions contain mostly plausible syntactic completions, with coverage way beyond a random baseline in all cases. At first glance, the model struggles at some positions (e.g., the verb in structure B – "Det Adj Noun Verb"), but these are characterized by context ambiguity, as noted previously.

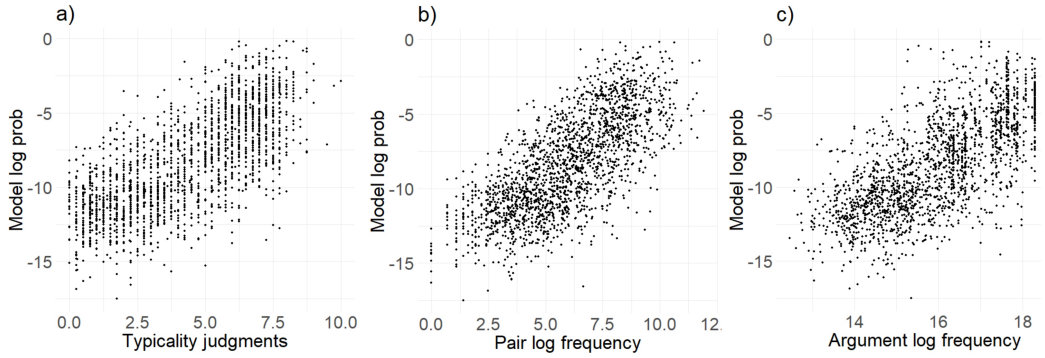


Figure 5.3: Relation between pairs of variables in the SP experiments.

On the other hand, we observe that purely distributional, order-free completions (as opposed to linguistically expected completions) also cover a great portion of the top predictions and often overlap significantly with the syntactically correct predictions of the model. This in turn raises the question of whether the model's abilities truly correspond to the capture of syntactic generalizations, or whether order-free co-occurrences alone suffice to produce the correct output in these contexts. However, in some cases, the model's syntactic ability strongly outperforms this baseline (e.g., for verbs apart from (B)), suggesting that the model's abilities do extend beyond purely corpus-based statistics.

5.5.2 Selectional preferences

Evaluation of human-like vs. co-occurrence based hypothesis Our results on the SP experiments are shown in Table 5.2, and we plot our variables against each other in fig. 5.3. According to our interpretation, R-squared scores confirm the

Metric	Model (predictors)	Score
R^2	θ_{H1} (**h**)	.51
	θ_{H2} ($\mathbf{f_{arg}}$ and $\mathbf{f_{comb}}$)	.58
Deviance	θ_{H2} ($\mathbf{f_{arg}}$ and $\mathbf{f_{comb}}$)	4,421
	θ_{H2+H1_b} ($\mathbf{f_{arg}}$, $\mathbf{f_{comb}}$ and **h**)	4,126

Table 5.2: Metrics of the models for the SP experiments.

adequacy of both H1-b and H2, as the variation in BERT probabilities can be explained equally well by crowdsourced data and the training corpus co-occurrence frequencies. The score for θ_{H2} is slightly higher, but this can be due to the model being more flexible due to the higher number of parameters and thus prone to overfitting. Overall, the main observation that we can make in this experiment is similar to that made on the syntactic task. While the model's behavior seems concordant with the human scores, its behavior can be almost equally well approximated using our co-occurrence based model, deprived from word order information. This hints at a potential limitation of behavior tests. In absence of a strong baseline which questions the true capture of the ability targeted by a given test, one would be tempted to conclude that the model under investigation might be capturing linguistic knowledge. In this series of experiments, we show that alternative hypotheses can account for the model's behavior, but this contradictory evidence is not always easily accessible as one cannot test all possible ways to approximate certain surface behavior. A similar argument had been made regarding the dangers of making claims about a model's ability to capture syntactic dependencies based on its ability to perform well on the number agreement task without considering alternative hypotheses [Kuncoro, 2018a].

5.6 Conclusion

In this chapter, we demonstrated the limitations of exploring a NLM's linguistic abilities using behavior tests only. On the surface, our experiments provide behavioral evidence supporting the capture of both syntactic and semantic knowledge. Indeed, the model's predictions approximate completions obeying syntactic constraints on the one hand, and reflecting human-like judgements of selectional preferences on the other. However, while our tasks are structure sensitive, our comparison with a purely distributional, order-free reference surprisingly shows that the model's abilities are comparable to that of a co-occurrences based predictor deprived from word order. As knowledge of syntactic constraints, and of selectional preferences cannot be captured without relying on word order, this evidence casts doubts regarding the extent to which behavioral success alone can be informative regarding the nature of the information processed by NLMs. In particular, they do not provide direct evidence for the inference schemes employed by the model when it succeeds in producing certain behavior. This finding also calls for testing several baselines when performing behavioral analysis, inviting us to investigate with more scrutiny which information the model uses to produce surface behavior. This finding raises the need to employ causal methods aimed at investigating more precisely what knowledge is used by a NLM on a given task. It also raises important questions regarding the utility of word order information, a feature that is key to structure-sensitive linguistic phenomena, for the model to mimick linguistic abilities. In the next chapter, we test the extent to which MLMs relies on their position encodings to optimize their training objective.

Figure 5.4: "A painting of an AI solving a puzzle with text on it", K.L. x DALL·E 2

Chapter 6

Investigating reliance on Word Order in Neural Language Models

Goals

In the previous experiments, we showed that a NLM's predictions on structure-dependent tasks can be partially retrieved by a co-occurrence based model deprived from word order information. In this chapter, we go beyond behavioral tests and investigate a NLM's reliance on word order. We analyze its performance on the masked language modelling objective, using a synthetic dataset. Our experimental setting is tightly controlled as we can compute the true probability of expected predictions given that the model is relying on word order or not. In doing so, we have two objectives. The first goal is to assess the importance of word order to the model in its normal functioning setting, following our previous observations. Word order is a crucial piece of information to encode sentence structure, and investigating its usage can shed light on the model's ability to capture higher-order linguistic abilities. We further build on our argument that behavioral evidence alone makes it difficult to pinpoint specific aspects of information processing in the neural network as it requires laser-focused interventions on the stimuli. Hence, our second objective is to demonstrate how targeted interventions applied to components of the model's architecture can lead us to conclude firmly regarding the information it processes.

Contents

6.1 Introduction

As we found that a NLM's behavior can be comparable to a simple order-free model based purely on co-occurrence statistics, we need to investigate more deeply whether the model relies on information about word order. A recent line of research investigated the importance of word order information during pre-training for models to solve downstream tasks, showing little variations when their input sentences are shuffled [Pham, 2021; Sinha, 2021a; Hessel, 2021]. In a similar line of research, [Haviv, 2022] found that even in absence of position encoding, models were still able to reconstruct the latter when probed for tokens' absolute position information in their intermediate layers. This finding in turn questioned the need for injecting explicitly position information in language models. [Abdou, 2022] also showed that shuffled models were still able to capture position information even when information about word order was removed after subword segmentation, likely because of the dependency between unigram occurrence probability and sentence length. Given all this work, it is surprising that the importance of explicit word order information in a neural language model still eludes us. In this study, we choose to investigate more carefully this phenomenon, and propose a methodology carefully designed to evaluate the importance of position encoding for the pre-training objective.

6.2 Encoding Position Information in Transformer-based NLMs

As seen previously, sentences seen at the input level of any neural language model can be seen as bearing information on the token content, and the token relative order. While tokens are always converted in vector form using token embeddings, processing information about their relative positions in the input sentence led to various proposals to encode word order information.

6.2.1 Different needs for autoregressive and bidirectional models

Word-order processing in autoregressive language models In autoregressive transformer language models, tokens are processed sequentially. When processing the i-th token, the model has access to the output of processing the previous sequence of token $(t_1, \ldots, t_{i-1}, t_i)$ (including itself), and their sequence of input token representations $(\mathbf{r}_1^{(0)}, \ldots, \mathbf{r}_i^{(0)})$ at each layer l. Denoting $f_{1:i}$ the subsequence

of i first output elements of f:

$$\begin{aligned}\mathbf{r}_i^l &= f_i^{(l)}(\mathbf{r}_1^{(l-1)}, \ldots, \mathbf{r}_i^{(l-1)}) \\ &= f_i^{(l)} \circ f_{1:i}^{(l-1)} \circ \cdots \circ f_{1:i}^{(1)}(\mathbf{r}_1^{(0)}, \ldots, \mathbf{r}_i^{(0)})\end{aligned} \tag{6.1}$$

In such models, information about position is essentially accessible as it could be inferred from the number of tokens that have been processed processed at each position. [Haviv, 2022] show that in such models, the absolute position information at each iteration can be reconstructed from its corresponding latent vector, which hints at the possibility that it could emerge from this implicit access to positional information.

Position encoding in bi-directional transformer models In bi-directional models in turn, each token is processed as a function of all tokens' intermediate representations, at any layer l, given a sequence of input tokens $(t_1, \ldots, t_n)$ and their input token embeddings $(\mathbf{r}_1^{(0)}, \ldots, \mathbf{r}_n^{(0)})$:

$$\begin{aligned}\mathbf{r}_i^l &= f_i^{(l)}(\mathbf{r}_1^{(l-1)}, \ldots, \mathbf{r}_i^{(l-1)} \ldots \mathbf{r}_n^{(l-1)}) \\ &= f_i^{(l)} \circ f^{(l-1)} \circ \cdots \circ f^{(1)}(\mathbf{r}_1^{(0)}, \ldots, \mathbf{r}_n^{(0)})\end{aligned} \tag{6.2}$$

By the nature of transformations applied in each transformer block, the permutation of indices would lead to exactly the same result at any position. This arises from the fact that each transformation of a transformer block is applied at each position independently, apart from the self-attention layer, where a commutative operation is applied over previous representations:

$$\forall 1 \leq i \leq n, \mathbf{h}_i^{(l)} = \sum_{j=1}^{n} A_{i,j}^l (W_V^l \mathbf{r}_j) \tag{6.3}$$

where

$$A_{i,j} = \frac{\exp e_{i,j}}{\sum_{k=1}^{n} \exp e_{i,k}} \tag{6.4}$$

and

$$e_{i,j} = \frac{S_{i,j}}{\sqrt{d_z}}$$

where $S_{i,j}$ is an attention score from position i to position j.

This invariance to permutation raises the need to encode position information in neural language models.

6.2.2 Contrastive Properties of Position Encodings

While a variety of distinct position encoding schemes are in use, they tend to differ from each other in a small number of properties [Dufter, 2021]. We review below several encoding schemes which, when paired, allow us to directly analyze binary decisions in the design of an encoding scheme.

Relative vs. absolute encoding In some models, it is the absolute position of tokens that is explicitly injected in the model while in others, it is the relative position of all token pairs that is injected. Both are perfectly equivalent in term of information they carry, as each can be reconstructed from the other.

Learned encodings vs. fixed position information injection Some models have learned encodings, either a set of learned position embeddings or a set of learned bias terms, while other models make use of a predefined function of position – relative or absolute – to inject that information in models.

Locus: adding embeddings vs. modifying self-attention Finally, models differ in the locus where position information is taken into account. A variety of models take position information into account by having position embeddings added, either at the input level or in intermediate layers. For a second category of models, position information is taken into account directly at the attention-level, and can take the form of biases which depend on position, added to the attention scores.

6.2.3 Position Encoding Schemes

In the following we review some position encoding schemes to give the reader examples of how position has been injected in some famous transformer-based architectures.

Position embeddings

Sinusoidal In the vanilla transformer from the first work which introduced transformers to NLP research [Vaswani, 2017], fixed position embeddings are added at the input level. The input of such model is thus the elementwise sum of token embeddings and absolute position embeddings. The latter are fixed during training

and generated using the sinusoïdal function:

$$\forall 1 \leq i \leq n \begin{cases} \forall t, 1 \leq t \leq \\ \text{if t is even, } t = 2k, \ \mathbf{p}_{i,t} = \sin(\frac{i}{10000^{2k/d_E}}) \\ \text{if t is odd, } t = 2k+1, \ \mathbf{p}_{i,t} = \cos(\frac{i}{10000^{2k/d_E}}) \end{cases} \tag{6.5}$$

These position encoding vectors are injected once and are only present in subsequent layers as a transformation of the injected input position embeddings, entangled with the content embeddings in the intermediate contextual representation.

Learned absolute position embeddings In BERT [Devlin, 2019b] and some subsequent models [Liu, 2019d], absolute position embeddings are also added at the input level, but are learned instead of fixed. Learning embeddings instead of using a fixed embedding function could add more flexibility to the processing of position information, in comparison to the previous solution.

Injected relative position embeddings In the previous proposals, it is the absolute position of each token that was injected in the models. The set of absolute token positions is equivalent to the set of relative positions as both can be reconstructed one from another. Following that observation, [Shaw, 2018] proposed to inject information about the relative positions of tokens instead of their absolute positions. In their implementation, no position embeddings are added at the input level of the network architecture, but two sets of $2k+1$ relative position embeddings $(\mathbf{p}^K_{-k}, ..., \mathbf{p}^K_k)$ and $(\mathbf{p}^V_{-k}, ..., \mathbf{p}^V_k)$ are learned (distances greater than k are clipped) and added at the attention level:

$$\alpha^K_{i,j} = \mathbf{p}^K_{\text{clip}(j-i,k)}$$

$$\alpha^V_{i,j} = \mathbf{p}^V_{\text{clip}(j-i,k)}$$

And the self-attention operation becomes, at each layer $l \geq 1$:

$$\forall i, \mathbf{r}^{(l)}_i = \sum_{j=1}^{n} Ai, jW^{(l)}_V(\mathbf{r}^{(l-1)}_j + \alpha^V_{i,j})$$

More recently, [Huang, 2020] proposed to build on this strategy and propose a variation of this encoding scheme to encourage more interactions between query, key and relative position embeddings.

Position-aware self-attention

Other work takes position information into account when processing intermediate representations, without injecting this information in intermediate representations [Dai, 2019; Raffel, 2020; He, 2020b]. In these alternative proposals, position information is only taken in weights, rather than intermediate representations.

Position encoding as an attention bias [Raffel, 2020] proposed to add a learned bias term directly in the computation of attention scores.

$$A_{i,j}^{(l)} = (W_Q(l)\mathbf{r}_i^{(l-1)})(W_K^{(l)}\mathbf{r}_j^{(l-1]})^\intercal + b_{i,j}^{(l)} \tag{6.6}$$

where $b_{i,j}^{(l)}$ is a trainable bias weight.

More recently, [Press, 2021] proposed to add a fixed bias term at each position of the attention matrix instead of learning that term. This bias term at the attended position is simply its relative distance to the attending token, multiplied by a slope weight that decreases geometrically, given $s < 1$:

$$b_{i,j} = (i - j) * s^{|j-i|}$$

Disentangling content and position at the attention level In [He, 2020b], the authors represent each token at position i using two vectors $\mathbf{h}_i$ and $\mathbf{p}_{i,j}$ which respectively represent its content, and relative position to the token at position j. Thus:

$$\begin{aligned} A_{i,j}^{(l)} &= (\mathbf{r}_i + \mathbf{p}_{i,j}) \times (\mathbf{r}_j + \mathbf{p}_{j,i})^\intercal && (6.7)\\ &= \mathbf{r}_i\mathbf{r}_j^\intercal + \mathbf{r}_i\mathbf{p}_{j,i}^\intercal + \mathbf{p}_{i,j}\mathbf{r}_j^\intercal + \mathbf{p}_{i,j}\mathbf{p}_{j,i}^\intercal && (6.8)\end{aligned}$$

Actually, the *position-to-position* term $\mathbf{p}_{i,j}\mathbf{p}_{j,i}^\intercal$ is removed as it brings no additional information, the embeddings already encoding relative position. Thus:

$$\begin{aligned}\hat{A}_{i,j} = (\mathbf{r}_i W_Q)(\mathbf{r}_j W_K)^\intercal \\ + (\mathbf{r}_i W_Q)(\mathbf{p}_{\text{clip}(j-i,k)}\hat{W}_K)^\intercal \\ + (\mathbf{p}_{\text{clip}(j-i,k)}\hat{W}_Q)(\mathbf{r}_j W_K)^\intercal \end{aligned} \tag{6.9}$$

and

$$A_{i,j} = \frac{\hat{A}_{i,j}}{\sqrt{3d_E}}$$

Now that we discussed why and how position is injected in transformer-based

neural language models, we study its importance for the masked language modeling objective in the next section.

6.3 Does my model even use position information?

As neural language models need to process information about the position of their input tokens to capture structural generalizations, we have seen in the previous section that a plethora of proposals to encode such information in transformer models have been made [Press, 2021; He, 2020a; Su, 2021; Chang, 2021; Chen, 2021a; Chen, 2021b]. Recent work, however, questioned whether word order information is really useful for pre-trained models to solve downstream tasks [Sinha, 2021a], showing that models could perform well when using only higher-order co-occurrence statistics. Other authors [Haviv, 2022] have shown that some transformers could reconstruct partly position information without it being explicitly injected. Examining performance on downstream tasks can show that the task simply does not require order information, or that the dataset used to test the model is too easy [Abdou, 2022], leading to indirect observations regarding a model's ability to reconstruct position information.

In turn, we choose to test the importance of position encodings for the pre-training task itself, masked language modeling, to get more direct evidence about whether and when position matters to language models. We do so under different amounts of masking, as intuitively, position information should be increasingly important when more tokens are missing from the context. Our experiments show that when masking only one token, the absence of position encoding has little effect on the model's performance. However, its importance increases with the number of masked tokens, forcing the model to leverage position information to perform better on its training objective. This finding should draw our attention towards choosing more carefully the amount of masking to train masked language models – a choice as important as the position encoding scheme itself.

6.3.1 Experimental Setup

Methodology

In our experiments, the goal is to investigate the extent to which a transformer neural model requires explicit position encoding to perform well on the masked language modeling objective. We do so under different amounts of masking to examine how this parameter affects the need for explicit position encoding. We make use of two variants for each trained model, one in which we inject position

information, and one deprived from explicit access to that information. To evaluate whether the trained model reconstructs its input sentences using position information, we compare its probability estimates q to two versions of the language's true probabilities p on the validation set. The first version, p_o, represents the probability of completions given the original, ordered input context. The second version, p_u, is the probability given unordered contexts. In the following sections, we explain how we perform this comparison to evaluate the extent to which explicit position encoding is required for the masked language modeling task.

Data

When using natural languages, it is hard to assess whether the model indeed relies on order information because it is not easy to design a dataset controlled to target specifically the usage of position information. In particular, as one does not have access to the true probability distribution of natural languages, it is hard to make clear predictions regarding how a model not using position information should behave. On the other hand, artificial languages obtained from a generative procedure that is known *a priori* make it possible to get tight estimates of their true probability distribution, both with, and without access to position information. The use of artificial languages has sparked interest over the past years, as a proxy to test targeted properties of neural models in controlled settings [White, 2021a; Wang, 2016a]. In our experiments, we make use of data released by White et al. [White, 2021a]. The dataset consists of sentences generated from an artificial grammar, using a PCFG such that all production rules have fixed probabilities.[1] This design makes it possible to evaluate the true probability of completions given masked input sentences, as a comparison point to the model's observed behavior.

The grammar comprises 1254 unique terminals, where 120 of the words are ambiguous. It is designed to implement morphological agreement, as its start symbol S appears in the rules $S \rightarrow NP\_Subj\_S, : VP\_S$ and $S \rightarrow NP\_Subj\_P, : VP\_P$. Derivations of noun phrases and verb phrases lead to non-terminals which keep track of number, leading to sets of terminal which carry this information. Additionally, different types of verbs exist in verb phrases: transitive verbs, intransitive verbs, and verbs which expects a sentential complement (comprising a complementizer). Nouns in turn can be either subjects or objects, which is indicated by a particle, and they can be modified by an adjective. Nouns can also be modified by prepositional phrases and relative clauses which respectively contain a preposition and a relativizer. Additionally, the grammar comprises conjunctions of

[1] The artificial language features certain constraints present in natural languages such as morphological agreement relations.

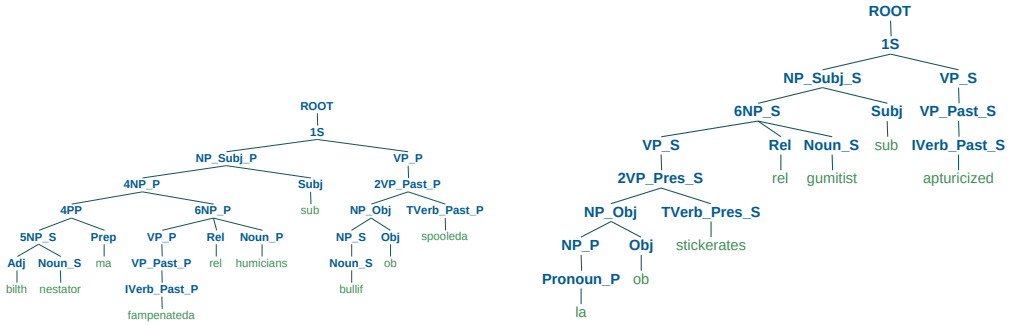


Figure 6.1: Examples of sentences found in the artificial language used for our analysis.

noun phrases, and conjunctions of verbs which bear the same number. We display examples of generated trees in fig. 6.1.[2]

Train size	100000
Test & Validation Size	10000
Vocabulary Size	1261
Mean Sentence Length	12.51

Table 6.1: Statistics of the dataset used to train our models.

Estimating the true probability distribution of the task

We exploit our direct access to the generative procedure which produces our input sentences to estimate the true probability distribution of the masked language modeling task. We do so by assuming that the context is either ordered or not. Specifically, we generate sequences recursively using the artificial language's production rules, until the probability sum of fully expanded sentences[3] reaches a certain coverage.[4] We then iterate over these sentences to mask words at each position and aggregate completions for sequences that share the same unmasked context in the Masked Language Modelling setting. We thus obtain a probability distribution of completions Y given (ordered) masked contexts X_o, which we write $\{p_o(y|x) \mid x, y \in X_o \times Y_o\}$.[5]

[2] In our experiments, we used the unaffected artificial language (Grammar 000000) released by [White, 2021a]. For more details, the grammar and original code can be found at `https://github.com/rycolab/artificial-languages`

[3] i.e. sequences that have no non-terminal label.

[4] We generate sentences along with their true sentence probability in our artificial language until we reach a probability sum superior to 0.75

[5] Note that when using natural languages, automatic collection of sentences in real corpora does not allow access to all possible completions in context, in addition to only providing sparse, and often biased, samples of sentences. Thus the true probability remains unknown, as noted in section 6.3.1.

We also compute a second version of the probability distribution that assumes no ordering of the context, aggregating completions for unmasked sequences whose unordered masked context is the same in X_u, obtaining $\{p_u(y|x) \mid x, y \in X_u \times Y_u\}$. To get the probability for unordered contexts, we simply group input sequences by sorting their elements alphabetically to remove order information and sum their probabilities for each unordered context. As we only use this procedure to remove information when estimating the task's true probability, the inputs which are seen by our models remain unchanged. As this removes all word order information when estimating the MLM task's probability distribution, our estimate is only dependent on information about each token's number of appearances in each input.

Is position information necessary for the task?

Given the true probabilities p_o and p_u for our task, we want to measure how different these are. We compute the KL-divergence:

$$D_{KL}(p_o, p_u) = \sum_{x,y \in X_o \times Y_o} p_o(y|x) \log \frac{p_o(y|x)}{p_u(y|x)} \tag{6.10}$$

This statistical distance allows us to estimate how different are the two distributions. We predict that by masking more tokens, the task would increasingly require position information and the divergence would also increase.[6]

Is position encoding useful to the model?

We test two variants of the BERT architecture [Devlin, 2019b], using Huggingface's Transformer library [Wolf, 2020]. In the first model, position information is encoded using learned absolute position embeddings,[7] while such explicit encoding is removed from the second. We call such models **BERT** and **NP**. Their hyperparameters are described in section 6.3.1.

For each model, we compare its probability estimates q in context to the task's true distribution assuming both that position information is present in contexts p_o, and absent p_u. We do so by computing the KL-divergence between q and $p \in (p_o, p_u)$ as follows:

$$D_{KL}(p, q) = H(p, q) - H(p)$$

We estimate the true entropy $H(p)$ for the masked language modeling (MLM)

[6] Note that while the KL-divergence is asymmetric, in this order the quantity represents the information gain achieved by having access to position information.

[7] This encoding scheme is widespread in transformer-based models, see Dufter et al. [Dufter, 2021] for an overview

Layers	3
Attention Heads	4
Hidden Size	256
Intermediate Size	1024
Training steps	300000
Weight Decay	0.01
Learning Rate	5e-5
Batch Size	8
Optimizer	Adam

Table 6.2: Hyperparameters for training and architecture of our models.

task using either p_o or p_u on our set of generated sentences:

$$\begin{aligned} H(Y|X) &= -\sum_{x,y \in X \times Y} p(x,y) \log \frac{p(x,y)}{p(x)} \\ &= -\sum_{x,y \in X \times Y} p(y|x)p(x) \log p(y|x) \end{aligned} \tag{6.11}$$

For each context, we compute the true entropy of its completions:

$$\forall x \in X, h_Y(x) = -\sum_{y \in Y} p(y|x) \log p(y|x)$$

And we finally compute the task entropy by averaging these context entropies over our kept masked contexts X_o or X_u:

$$H(Y|X) = \sum_{x \in X} p(x) h_Y(x)$$

We obtain two true task entropy estimates, $H(p_o)$ for ordered contexts, and $H(p_u)$ for unordered ones. For each model, we then estimate the cross entropy to each true distribution. Denoting the model's output probability q, the cross-entropy writes as follows

$$H(p,q) = -\sum_{x,y \in X \times Y} p(y|x) \log q(y|x)$$

We then use the task's true entropy and the model's cross-entropy to compute the KL-divergence. For each model, by comparing $D_{KL}(p_o, q)$ to $D_{KL}(p_u, q)$, we can assess whether the model's estimates fit better the task's probability for ordered contexts, or unordered contexts. If explicit position encoding is necessary, we predict that $D_{KL}(p_u, q)$ should be greater than $D_{KL}(p_o, q)$ for **BERT**, and lower for

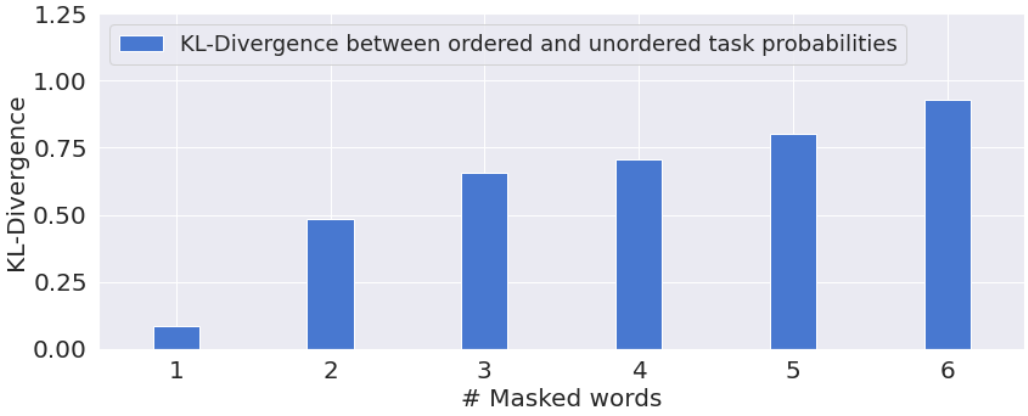


Figure 6.2: KL-Divergence between the ordered and unordered true task probabilities.

NP. Otherwise, both models should have similar behavior.

Testing the effect of masking

In this study, we compare **BERT** and **NP** under different amounts of masking. We surmise that increasing that parameter should increase the necessity of using position information, as measured by eq. (6.10). If this is the case, varying this parameter will allow us to investigate whether position encoding is necessary as the task increasingly requires using that information.

6.3.2 Results

We first display the KL-divergence between true probability distributions assuming ordered and unordered contexts in fig. 6.2. In accordance with our expectations,[8] when increasing the amount of masking, the true distribution of completions given ordered contexts diverges from that of unordered contexts. Interestingly though, when only one token is masked, the divergence is low. This suggests that in this setting, models should have little difference regardless of whether they have access to explicit position information. By increasing the amount of masked tokens, we can further observe that the two considered true probabilities p_o and p_u diverge. We thus expect that models should increasingly rely on position information to approximate the true ordered distribution.

We further display how well each model approximates each probability estimate in fig. 6.3 to verify whether the presence of position encoding is useful to the masked language modeling task under different amounts of masking. Expectedly, the model with no position encoding scheme performs similarly to the BERT

[8] see section 6.3.1

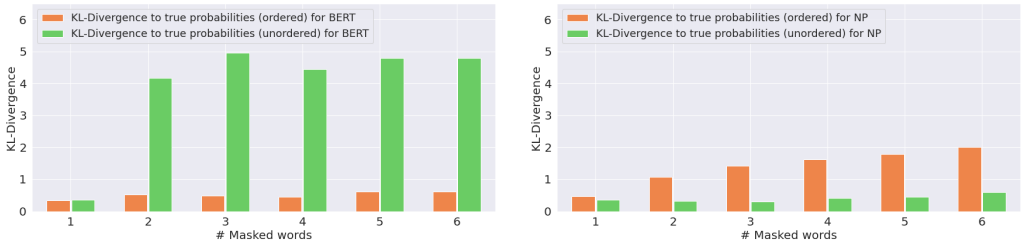


Figure 6.3: KL-Divergence between the true task probabilities and our models' probability estimates (**BERT**-top and **NP**-bottom), assuming contexts are ordered (orange bars) and unordered (green bars).

model when only one token is masked. In this setting, the context contains enough information for the model regardless of whether it sees its input tokens as ordered or as a bag of words. When masking more tokens however, this difference becomes increasingly marked. We display the perplexities reached by our models on our validation sets in table. 6.3. Note that these perplexities are obtained in the traditional masked language modelling setting, where only one word is considered to be the ground truth. This explains the discrepancy when compared to model cross-entropies in fig. 6.4. Contrarily to the rest of our analysis, these perplexity scores do not take the true probability distribution of the task into account, as only one label gets all the probability mass.

Model	**# Masked Words**					
	1	**2**	**3**	**4**	**5**	**6**
BERT	15.12	17.06	17.6	19.02	20.24	20.37
NP	20.14	41.93	55.45	70.64	93.58	107.46

Table 6.3: Perplexities reached by our tested models for varying numbers of masked words.

Further, we observe that the **BERT** model has a low divergence to the true probability assuming ordered contexts regardless of the amount of masking, while it diverges increasingly from the distribution that assumes no ordering of the context. The opposite pattern holds for the **NP** model. Taken together, these results show that position encoding is necessary to approximate the true distribution of the task when it requires position information, that is when the number of masked tokens is increased.

In fig. 6.4, we compare our models' cross-entropies to the task's true entropies. The figure aggregates the two main observations made in this article, that when the number of masked tokens increases: (i) the true entropy of the data with and without position diverge from each other, and (ii) that position encoding is required to approximate the task's true probability distribution assuming ordered contexts. Ac-

cordingly to our previous observations, the **NP** model, which does not have access to the ordering of tokens, has a cross-entropy that fits the true probability distribution's entropy assuming no ordering of the context (red lines). Looking at **BERT**'s cross-entropy, we see that this model, which has access to position information, rather fits the true probability distribution assuming the context is ordered.

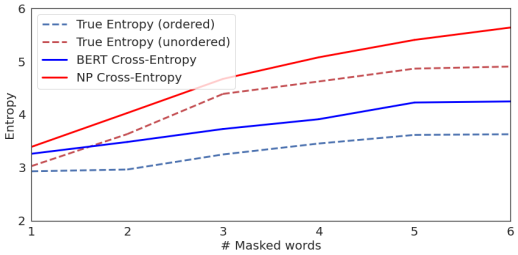


Figure 6.4: A comparison between entropies of true probabilities for the MLM task (assuming ordered and unordered contexts), and our models' cross-entropies

6.3.3 Discussion

Position encoding and language modeling

Previous work claimed that transformer autoregressive language models without position encodings could reconstruct position information by inferring the number of preceding tokens, but not bidirectional transformer models [Haviv, 2022]. Testing a RoBERTa model [Liu, 2019c] led to great difference in perplexity when removing position information at the input level. However, we show this difference to strongly depend on the amount of masking: as autoregressive language models predict only one token at a time, the task could be equally easy for models deprived from position information. Our results call for increased scrutiny when comparing autoregressive and masked language models, making sure that they are asked to predict comparable numbers of tokens.

Mask more !

In our study, we have shown that the utility of explicit position encoding increases with the number of masked tokens. This finding echoes Wettig et al. [Wettig, 2022]'s study, showing that masking 40% of tokens rather than 15% during pretraining leads to better performance on downstream tasks. This evidence could draw more attention towards understanding how different amounts of masking can lead models to rely on position information, and capture more structural knowledge about the languages they model.

Limitations

The results we have presented in this paper were obtained over artificial languages. Adapting the method to natural languages may be difficult.

The true probability distribution is not accessible for natural languages. In this study, we investigate how the amount of masking impacts the usage of position encoding by a neural language model. We chose to carry out this experiment on an artificial language, because of the ease to access the true probability distributions in each setting. While this result informs us that the amount of masking could be key for masked language models to use and abstract away from position information extracted from their input, this methodology is not easy to adapt to natural languages, because the true probability distribution is not accessible for natural languages. In future work, one could try to find proxies to estimate reference points for natural languages, with potentially looser estimates than the one used in this study.

Training several masked language models on natural languages is computationally expensive. In order to investigate how the amount of masking impacts the degree to which a NLM makes use of its position encodings, or higher-order structural properties of natural languages, one would need to train a large neural model for each condition under investigation, and for each retained amount of masking. This, added to the potential hyperparameter space search would require substantial computing resources as training a model on natural languages requires large amounts of data during training.

Natural languages are more flexible regarding word order. In our experiments, we investigate the impact of masking on using position information using artificial languages where word order is fixed. We conclude that neural language models make use of position information on the masked language modeling objective when the number of masked tokens increases. However, while this should hold true for data similar to ours, where the word order is fixed and hence position information greatly affects which token needs to be predicted at a certain position, we cannot make claims regarding the impact of masking on languages where word order is more variable, which is the case of any natural language. Further analyses are needed to evaluate whether position encoding impacts language modeling in different ways when word order is rather fixed (like English), compared to when it is more variable (like in Latin or Finnish).

6.4 Conclusion

In this chapter, we evaluated the importance of position encoding for a masked language model. We showed that without explicit access to position information, a model can obtain performance similar to a model that learns position embeddings, when only one token is masked. We find that when increasing the number of masked tokens, the output probability distribution assuming unordered inputs diverges from that which assumes ordered sentences, reflecting that the task increasingly requires making use of position information. We further show that under this condition, models with explicit position encoding outperform their counterpart deprived from position information. This in turn should raise awareness that the amount of masked tokens might be a crucial parameter for models to abstract away from their input sentences' position information, in addition to the chosen position encoding scheme. With this evidence, we can rule out that pre-trained language models rely exclusively on statistical co-occurrences. We also show them to be perfectly able to memorize mappings between inputs and outputs when only one token is masked, as they do not seem to rely on word order in this case. By investigating whether the presence of position encoding matters to the model, we start opening the blackbox and understanding what function its different components play in supporting the observed linguistic abilities. While we ensured that position information and encodings are indeed useful above a certain amount of masking, we do not know how position information is used by the model, as it is a low level-feature that is not easy to map to interpretable linguistic knowledge. In the next chapters, we go further and propose a framework to find linguistic representations in intermediate layers, which are used by a NLM in the context of a linguistic task.

Figure 6.5: "A drawing of a man examining a network", K.L. x DALL·E 2

Chapter 7

Beyond Behavior - Inner Mechanisms Supporting Linguistic Knowledge

Goals

In this chapter, we complement behavioral analysis by taking a look inside the black box neural model. While behavioral adequacy under certain constraints is a good starting point, we have seen that in the general case, it provides limited information on the nature of the algorithm supporting such behavior. For instance, when relying purely on the observation of behavioral outcomes in controlled settings, the set of possible algorithms yielding certain surface behavior remains large. In this chapter, we discuss the importance of understanding how a model's behavior is produced by taking a look at its internal representations and computations. In doing so, we go beyond behavioral evaluation and explore the algorithmic level of analysis of NLMs. Specifically, we show how this can be done by intervening at the implementational level. We focus specifically on linguistic representations. Under this lens, learning a linguistic concept equates to representing and using that concept. We further attempt defining criteria over the representations of linguistic properties to assess that a model captures abstract representations of linguistic concepts in its intermediate layers, which it uses to produce output behavior.

Contents

7.1 Motivation: is my Model Right for the Right Reasons?

As pointed out in section 2.2.4, one of the main reasons why tremendous efforts are invested by NLP researchers in probing neural language models is to ascertain that they truly capture linguistic knowledge when they seem to succeed in downstream usage. The overarching goal of probing would be to provide a clear picture of the knowledge possessed by NLMs and the way they mobilize such knowledge on high-level tasks – if they do.

In previous chapters, we have shown that careful analysis of a neural language model's behavior informs us on whether the latter is able to demonstrate linguistic abilities. In the worst case, the model is only able to mimic and approximate such behavior without having acquired any linguistic knowledge, and in the best case behavioral success reflects its acquisition of adequate knowledge which it is able to mobilize at inference time. Distinguishing those two extreme cases is important, so as to not mistake a stochastic parrot with a system capturing subtle abstractions due to limited data. Using carefully designed datasets, behavioral tests allow us to rule out aspects of systematic linguistic generalization, by pinpointing contexts of failure for the model, which partly sheds light on the processes underlying behavioral success. Behavioral tests can thus be tailored to target knowledge that is more fine-grained than higher order linguistic abilities such as language inference. Such tests also possess an advantage on downstream evaluation by usually observing the model in its normal training setting, with no further fine-tuning. Another advantage of testing the model against tasks targeting more fine-grained linguistic knowledge is that they allow for more control, as they are simpler tasks: in principle, less parameters should influence success on a task which requires less knowledge.

However, behavioral tasks do not completely circumvent the issues of downstream evaluation as the model is still treated as a blackbox, therefore their epistemic validity is also questionable in the general case. In this section, we explore potential causes for incorrect generalization schemes.

7.1.1 Heuristics can fail to capture linguistic rules

As seen in section 2.2.4, models which seem to produce correct answers on a high-level downstream task such as NLI [McCoy, 2019] might be relying on shallow heuristics, or be right for the wrong reasons. The same is true for the apparent capture of finer-grain linguistic abilities. Shallow heuristics can be one reason why the model could succeed on behavioral tasks without having appropriate knowledge.

Adequate linguistic knowledge is adequate knowledge of categories and their relations, that is the mutual constraints they impose on each other. The model could represent adequately linguistic categories on which a given constraint is imposed, but not their correct mutual constraints. For instance, it could know what a singular noun and verb are, but rely on shallow word-order based heuristics to guess the correct number instead of learning to track the dependency relation between the cue and the target of the agreement relation.

As a synthetic example, if my test set is the following :

$$\sqrt{25} =?$$

$$\sqrt{36} =?$$

The model can guess the correct answers by applying the rule *"return the unit number of the argument"*. Another heuristic in the context of number agreement would be to return the number of the preceding noun, which works when applied to "The *key* **is** on the table" but not "The *boy* that painted the walls **is** leaving".

The ability to make firmer conclusions in turn comes from the ability to craft controlled settings, that is conditions controlled to test precise hypotheses, as done on downstream usage to test reliance on certain heuristics [McCoy, 2019].

However, relying on carefully designed settings to test specific heuristics is limiting, as the evidence for generalization is based on negative examples: when ruling out a given heuristic, it is only more likely that the model possesses linguistic knowledge. But *how likely is it*? If one can rule out certain heuristics using behavior tests, one cannot rule out all of them. So unless the controlled conditions cover the whole space of possibles (as we did when investigating word order: either that information is present or it isn't), it is in practice hard to ensure representativity of the data for each chosen condition, and the cause underlying success or failures of the models in given conditions might only be **concomitant** to the control parameters. A causal account of how the model came to produce an output prediction in turn can give more direct evidence for sources of successes and failures.

7.1.2 Memorization of lexical patterns does not require representing linguistic categories

Another reason why the model might be generalizing without having appropriate knowledge could be due to memorizing patterns without having any representation of linguistic categories. This of course, is an extreme case and the extent to which the model captures knowledge of categories or is a pure memorizer of input/output

mappings is not measured on a binary scale. For example, instead of representing verbs as a set of tokens which play a common syntactic role and occur in similar contexts, it could be just memorizing specific word combinations and sequences, as pointed out in our experiments on number agreement. If equipped with enough capacity, a NLM can memorize a wide array of sentences without deriving the general rule underlying the acquisition of a given concept. Using artificial data in chapter 6, we demonstrated that a model deprived from word order information is able to be as good as its counterpart armed with position encodings, in reducing its loss on the MLM objective when only one token is masked. The MLM task however (guessing which tokens fit in a masked position, for sentences which are generated from a CFG grammar) seems to require knowledge of position information. It is likely that in this case, the model is only learning to memorize a mapping between sets of lexical categories present in the input context to the masked output category, instead of learning to rely on representations of local and global structures. Its exposure to large pre-training data can make it very robust on linguistic tests, while it could have just memorized patterns. This has two disadvantages:

The model has access to limited data As the model cannot memorize any possible word combination, it falls prey to severe limitations by memorizing lexical patterns without deriving knowledge of structure and categories. Generalization should in turn be the derivation of a general rule, or general categories. While we tested this in the last section for syntactic generalization, we can push further this by searching for generalized representations of linguistic properties, as well as how they are mobilized at inference time.

Memorizing patterns is less memory-efficient Learning to predict specific words in specific contexts is more memory-expensive than memorizing rules applied to categories. Given that there are n verbs in the model's vocabulary, and given $c =$ *"The boy [MASK]"*, memorizing that each singular verb should be predicted with a higher probability than each plural requires memorizing $n \times n$ independent orderings of probabilities if the model memorizes each ordering for each pair (e.g. p("goes") > p("eat")). While this looks suboptimal at first glance, the model's large number of parameters could support sufficient capacity to memorize a large number of combination without deriving general structural patterns, that is general regularities, or rules, which apply to linguistic categories instead of individual word combinations.

Learning general representations in turn should be more memory-efficient: if the model encodes singularity and plurality in its last layer's representations (two

sets of n elements), it is in turn sufficient to memorize that the probability of all singulars needs to be lifted against that of plurals, so it only needs to memorize a single ordering, in addition to the two sets of n tokens for each category in (singular, plural). Lifting each probabilities individually probably requires more complex computations but should in principle be possible, as we have shown in the previous chapter that under certain conditions, neural models have the capacity to be perfect memorizers.

For these reasons, understanding which computations take place inside the neural architectures is a crucial area of exploration to know more exactly how the model is making decisions, and in particular how it produces correct predictions on behavioral tasks.

7.1.3 The need for causal methodologies

As seen previously, one cannot know how the model produces output decisions, which limits the scope which can be explored using behavior tests, and the extent of the model's linguistic knowledge which can be uncovered using uniquely such methods. Causal methods in turn make it possible to formulate more precise hypotheses, and allow for more direct evidence that certain linguistic information is used by the model, or that a certain components plays a given role in producing the correct decision. As causal methods are tailored to intervene precisely and measure the effect of granular bits of information or components of the model, they in principle do not fall prey to taking spurious parameters as explanatory for the model's successes or failures. Common examples include generating counterfactual inputs, ablation experiments, or setting the value of intermediate states to nonzero values[1]. Last chapter's experiment was one such example as the position encoding scheme has been disabled in the original transformer architecture, resulting in inability for the model to recover position information. This was made possible because we could on the other hand intervene precisely on the true probability distribution $p(y|X)$ in addition to the position encoding, and translate our hypothesis into clear expected observations. In most cases, it is not possible to intervene precisely on the probability distribution which should be estimated at the output level $p(y|X)$, but it is possible to perform causal intervention on input sentences, by generating counterfactuals which differ minimally from their original counterparts. The model itself in turn bears a lot of information on how it processes linguistic information, which can be uncovered by performing causal experiments to evaluate the function of its different components in transforming input information and transmitting it

[1] Ablation experiments generally can be seen as setting some values i the model's weights or intermediate representations to zero

to the decision layer. In this chapter, we will see how to open the blackbox and intervene causally on the model's components.

7.2 Inside the Blackbox: the Functions of a NLM's Components

For any component of a NLM, acquiring a function during training is equivalent to specializing into accomplishing a specific role. In the MLM setting, the component should play a role in reading the information in the input, and transforming that information into a decision at the output layer. In this section, we present the different types of functions we can look for in different components of a NLM.

7.2.1 Different functions for representations and transformations

As we have seen in section 2.2, Transformer-based NLMs are the composition of transformer blocks:

$$R^{(l)} = f^{(l)} \circ f^{(l-1)} \cdots \circ f^{(1)}(R^{(0)})$$

Which themselves result from the composition of parametric functions. Each layer, or function, transforms its input representations, a sequence of vectors, into another set of vector representations.

Representations and layer transforms naturally acquire different functions as representations typically bear information about the input context, while layers and transforms read and pass information to the subsequent layers. While input embeddings bear information about a single position or a single token type, higher-layer representations start gathering information about the context and can bear information about a span or sequence of tokens, or about a token-pair, and more generally speaking a subpart of the context greater than the token. Transforms and layers in turn can be responsible for transforming the format of information, carrying information from one position to another, or merging information from different positions. As the self-attention mechanism is the only place where information is passed across positions, it is by default the mechanism responsible for merging information from different positions and copying information in other positions, while the other layers should in principle only modify the format of the information present after the self-attention mechanism is applied.

Evidence for specialization in attention heads Not long after transformer-based NLMs mushroomed in NLP pipelines, researchers turned to the attention mechanism as the distinctive feature of transformers, looking for interpretable patterns. Quickly, some authors found patterns supporting specialization in attention heads [Clark, 2019; Kovaleva, 2019]. Some of these studies find salient patterns which seem to play a function at a very low-level, and are thus not directly interpretable [Kovaleva, 2019]. In other studies, authors have tried to find attention maps in which attention weights reflected that two words share a specific dependency relation [Clark, 2019]. More recent work showed feed-forward layers stacked on top of the self-attention mechanism to be memorizers [Geva, 2021]. Still, there seems to be an interpretation gap between these observations, and how we could imagine linguistic phenomena to be handled. While first studies looked at attention because it is the distinctive feature of transformers, and because attention maps are easy to visualize as a 2D matrix, other components of the network's layers could equally specialize into capturing certain functions. One could argue that it is more difficult to understand how layers transform information (e.g. by grouping, merging, or transforming information from various positions), so an easier quest for starters could be to seek knowledge encoded in intermediate representations.

Understanding the model's representations - causal evidence for encodings of linguistic representations Previous research also shows evidence for specialization of neurons into capturing specific linguistic properties. For instance, [Lakretz, 2019] find two "number units", or neurons responsible for encoding number in LSTM language models. [Vig, 2020b] in turn are able to find neurons and attention heads responsible for mediating gender bias in a NLM. Contrarily to experiments cited in section 2.3.1, these experiments are causal in nature, thus they are able to provide evidence for representations which play a role in determining the model's output.

Desired functions On the other hand, when seeking linguistic functions inside the neural architecture, one would seek representations of a given linguistic property or unit (see section 3.1), or a linguistic constraint (such as "transfer number information from the cue to the target of an agreement relation"). It is desired that the way information is passed by layers, and the nature of information which is encoded in layers, is interpretable from a linguistic point of view and constitutes adequate knowledge of categories, and of the mutual constraints they share. Such investigation is guided by the assumption that some linguistic knowledge can be represented by means of categories, and rules systematically operating over cate-

gories using algebraic operations. We can keep in mind that it is not reasonable to assume that all of linguistic knowledge and linguistic processing can be easily translated into precise categories or functions which would have a precise location in the network. For instance, some authors found syntactic processes to be distributed across the whole language system in the human brain [Blank, 2015].

7.2.2 The challenge: decode me if you can

Due to the complexity of their inner structure and the large number of parameters they bear in their layers, Transformer-based neural architectures are difficult to apprehend. As discussed in the previous section, gaining better understanding of these architectures requires getting to know better what information their representations encode, and understanding the computations which take place inside the network to transform the format of that information. To achieve this goal, we need to get a clearer picture of the causal relations which connect these encodings to the model's outputs, and eventually the function which the architecture's components play in formatting the information encoded in intermediate layers, so it can be read to make a correct decision at inference time. Achieving this requires being able to decode intermediate representations by mapping them into interpretable, potentially discrete, states[2] which represent meaningful information about the presented context.This also requires understanding how layers combine and read information about the input context. At the lowest level, the only readable information present is the set of token contents and their position. We can imagine that pieces of local context information are merged to form pieces of information relative to a greater scope, until the model is able to confidently make a prediction given the whole context. This interpretability entreprise is certainly not easy. First, functions are emergent – the mechanisms which gives rise to specialization in the network remain unknown and no component of the network is forced to learn any of the functions that should be necessary to produce correct outcomes, so one cannot know where to look beforehand. Furthermore, one needs to have an idea *a priori* on a function or an inference scheme before looking for it in the network, whether it is a defective heuristic or a general algorithm which always works. These difficulties are almost the same as the ones encountered when trying to assess the function of brain areas, with at least three advantages though: (i) simpler mechanisms and simpler state spaces (ii) perfect access to intermediate representations without any ethical consideration (iii) only language.

[2]While a major part of linguistic knowledge, such as knowledge of lexical categories and syntactic relations, is discrete, that is not necessarily true of all linguistic knowledge. The evidence for gradience in human judgements is one argument against discreteness.

7.3 What does it mean to *represent* a linguistic property?

In this section, we take a closer look at the function of representations, and in particular define desiderata to assess that a representation truly captures a given linguistic property of interest.

A layer represents a linguistic property iff it encodes information about that property such that this information is decoded by subsequent layers, that is iff it encodes information in a format that is read by the model itself. Under these conditions, the function of the encoding is to represent that property. Spurious encodings in turn have no function in the architecture. Without evidence that an encoding plays a function in the network, it might just be a spurious encoding which does not *represent* the linguistic property.

7.3.1 What is an encoding?

As seen previously section 3.1, (contextual) linguistic properties are defined over parts of a sentence, and consist in mapping such into a set of values. An encoding of such property is a mapping from a representation space, into that same set of values.

In a canonical intermediate representation space, the source property should in principle be encoded in only a subspace that specialized into its representation for subsequent layers. A candidate encoding of a property (identified to its set of values $\mathcal{L}_s$) in a subspace $\mathcal{U}_i$ of an intermediate space $\mathcal{R}_i$ (at position i) is a mapping from a partition of $\mathcal{U}_i$, $\mathcal{E} = (P_1^{\mathcal{U}_i}, ..., P_{|\mathcal{L}_s|}^{\mathcal{U}}) \in \mathcal{P}(\mathcal{U}_i)$ into the set of values $\mathcal{L}_s$ that the property can take. The candidate encoding thus defines a function $f_{\mathcal{E}}$ from $\mathcal{U}_i$ to $\mathcal{L}_s$ (from elements, or parts, in the partition $\mathcal{E}$ to the property's values in $\mathcal{L}_s$).

The function f defines a mapping from a partition of $\mathcal{U}$, $\{f^{-1}(v)|v \in \mathcal{L}_s\} \in \mathcal{P}(\mathcal{U})$ into $\mathcal{L}_s$, we write $f : \mathcal{P}(\mathcal{U}) \mapsto \mathcal{L}_s$.

For each element of our dataset $(x, \pi_s, \ell_s, \pi_t, \ell_t) \in \mathcal{D}$, we denote the intermediate representation in the considered space $\mathbf{r}_{x,i} \in \mathcal{R}_i$. The considered subspace is $\mathcal{U}$ and we write $\mathcal{R}_i = \mathcal{U}_i \oplus \mathcal{U}_i^{\perp}$. We thus denote $\mathbf{r}_{x,i} = \mathbf{u}_{x,i} + \mathbf{u}_{x,i}^{\perp}$.

Functional encoding An encoding is functional if this mapping is readable by the next layer, that is, the bottom architecture encoded the information about that property using this partitioning of space, to pass it to the upper architecture, which needs that information to produce a correct decision.

Format of the information The format of the encodings that one can search for is defined by the family of boundaries used to partition the representation space. Linear classifiers partition the space linearly[3], but more complex boundaries can separate the space. In addition to being the simplest boundary one can think of, another potential motivation for linearity is that at the decision layer, each ordering of any two tokens is defined by a linear constraint over the output representation, so readable information should in principle be encoded linearly. This logic however doesn't apply to lower layers, as successive non-linearities allow more complex encodings to be perfectly readable by subsequent layers, which can transform it into information that is encoded linearly at the decision layer.

7.3.2 Extractability of information

As seen in section 2.3.1, researchers widely accepted for several years that being able to extract information about a given property from representations meant that the property was encoded in representations. We have seen however that this common wisdom raised serious questions and rapidly led to absurd conclusions, such as that representations encode random labels. This condition is necessary but not sufficient, as the extractability condition could be verified for a spurious encoding which is not read by the next layers.

Consider the following two statements:

My NLM encodes information about property P in layer l. (P1)

Information about property P is extractible from layer l. (P2)

It follows that $(P1) \implies (P2)$, but the opposite is not true, as the encoding might just be spurious.

7.3.3 How is the information encoded ? the format

An encoding, that is a partition of a representing space, can be sought in a family of mappings (e.g. linear mappings, but also Kernelized classifiers...). The simpler the transform, the easier it will be to perform causal interventions on representations.

In short, the encoding will (either implicitly or explicitly) partition the space $\mathcal{R}_t$ into $|\mathcal{P}|$ parts, one for each value the property $\mathbf{p}$ can take. In its turn, this partitioning can be easily represented as a function $\text{map}_{\mathbf{p}_{\mathbf{r}_t}} : \mathcal{R}_t \to \mathcal{P}$, which

[3]To be more precise, the space is rather split by affine hyperplanes

predicts the value an analysed property should take given a sub-representation. Formally, we can write:

$$\mathbf{p}_{\mathbf{r}_t} = \mathrm{map}_{\mathbf{p}_{\mathbf{r}_t}}(\mathbf{r}_t) \tag{7.1}$$

The format of an encoding can be seen in this case as equivalent to the "shape" of the boundaries delimiting the different partitions.

7.3.4 Where to look?

Candidate encodings can be sought in a subspace chosen in any slice of the causal graph representing the neural network. Often, it is more convenient to choose a vector space at a specific token's position if the property is a token-level property. The candidate representing spaces can depend on the architecture however, in any case in transformers representing spaces correspond to token positions. This implies that by choosing such spaces, we can see where information is passed from specific positions to other positions, but it makes it harder to look for properties defined over a scope that is above the token position. This difficulty can be circumvented by simply concatenating spaces over all positions which compose the span or combination under consideration, or by considering a combination of representing vectors for each tokens such as a weighted sum. In the general case, a property $\mathbf{p}$ can be encoded in a subspace $\mathcal{R}_t$ of the representations space $\mathcal{R}$, we write $\mathcal{R}_t \subseteq \mathcal{R}$. The motivation behind identifying these subspaces is that parts of this representation space may contain information that is unrelated to our analysed property—identifying $\mathcal{R}_t$ may thus improve our model's interpretability and allow us to make target interventions on them.

Looking for representations equates to finding a subspace encoding properties in intermediate representing spaces. The subspace which supports the encoding is therefore found by partitioning intermediate representation spaces using complementary functions f_t and $f_{\neq t}$, as in:

$$\mathbf{r}_t = f_t(\mathbf{r}), \qquad \mathbf{r}_{\neq t} = f_{\neq t}(\mathbf{r}) \tag{7.2}$$

In this equation, $\mathbf{r}_t \in \mathcal{R}_t$, and $\mathbf{r}_{\neq t} \in \mathcal{R}_{\neq t}$ represent subsets of the representation which, respectively, encode or not the probed property. Notably, this function $f_t(\cdot)$ is usually defined such that $\mathbf{r}_t$ encodes the target property $\mathbf{p}$. In turn, the function $f_{\neq t}(\cdot)$ is typically defined as:

$$f_{\neq t}(\mathbf{r}) \equiv \mathrm{id}(\mathbf{r}) - f_t(\mathbf{r}) \tag{7.3}$$

This ensures that the original representation can be additively reconstructed from

its subparts, i.e.:

$$\mathbf{r} \equiv \mathbf{r}_t + \mathbf{r}_{\neq t} \tag{7.4}$$

Further, we have that $\mathcal{R} = \mathcal{R}_t \cup \mathcal{R}_{\neq t}$. In this sense, the method to find a subspace encoding a property split the representations into two separate subspaces—represented by $\mathbf{r}_t$ and $\mathbf{r}_{\neq t}$—which, when put back together, reconstruct the original representations entirely.

Note that, given equation eq. (7.3), the method to isolate the subspace can be uniquely identified by its choice of $f_t(\cdot)$. Several recent papers perform subspace probing, either identifying individual relevant neurons which encode some property [Torroba Hennigen, 2020; Vig, 2020b] or identifying entire high-dimensional subsets of the $\mathcal{R}$ space [Ravfogel, 2020; Elazar, 2021].

7.3.5 Measuring causal effect over behavior

As soon as the subspace encoding a given property communicates that information to the next layers, we can safely say that this information was encoded to be decoded by the network itself. As discussed in table. 3.1, the ultimate function of any component in the network is to reduce the training objective, that is to produce correct completions in context. Determining that an encoding is meaningful to the model thus requires providing evidence that the information encoded in that subspace plays a causal role over the model's predictions.

7.4 Wrapping up: from Encoding to Usage

While the questions addressed in this chapter apply equally to all components of a given NLM, including representations but also transforms, we accepted that linguistic representations are easier to seek for starters. As we focused on such objects in the last section, we finally give a concrete procedure to find encodings which are useful to the model. In our framework, an account for a property's encoding amounts to a description of its format – or geometric structure – in an intermediate representing space such that this format is used (or decoded) by the model itself in its normal functioning setting – e.g. predicting the content of a masked word for masked language models (MLMs).

Prerequisites Two sanity checks can be performed to find linguistic representations that are actually meaningful to the model under investigation.

- Extractability: Information first needs to be extractible from layers. Extractible information in turn can be estimated using various measurements. The most widespread measurement to quantify extractability is simply accuracy on the probing task, but other measurements have been proposed, such as selectivity [Hewitt, 2019a], MDL [Voita, 2020], or $\mathcal{V}$-information [Xu, 2020b]. As seen in the previous section, this is only a necessary condition. It can be tested as a preliminary, otherwise looking for a functional encoding is vain. This makes it possible to know in which format one can hope to find a functional encoding, representing the linguistic property.

- Behavioral success: A task requiring knowledge about the linguistic property for which we seek representations is needed beforehand, and we require that the model succeeds on instances of that task. Without this condition respected, it is not possible to find a linguistic representation that is mobilized and useful to the model.

Those two conditions are necessary, but not sufficient. The extractability condition defines a candidate encoding (clf(-1)). We found an encoding of the property if modifying the state in that candidate encoding modifies the output accordingly. A weaker condition consists in simply erasing the state. If task performance drops to chance level, it is very likely that the encoding was actually bearing that information for the model to produce the correct behavior.

On the one hand, we need a **behavioral task** defined as correct outcomes given a set of carefully chosen contexts. As MLMs produce output probabilities for a discrete set of words (the model's vocabulary), success on the behavior task is

equivalent to favoring a subset of the vocabulary $\mathcal{V}$ in a given context. Producing the correct response in principle requires the usage of our encoded property in the model's layers. Each of such subsets forms an expected response at the behavioral level, and can be written $(R_1, ..., R_n)$, where $\forall i, R_i \subset \mathcal{V}$.

Just describe here the methodology :

(I) Selecting a behavioral task We choose a behavior task on which the model performs well, reflecting the capture of a linguistic ability that we wish to understand.

(II) Choosing a piece of linguistic information We then choose a linguistic property. As discussed before, representations for token-level properties are easier to look for in transformer-based architectures as each token has an intermediate representation at each layer.

(III) Picking a format and location, and making sure information is extractible After choosing a location (a representation space) and a format for the encoding (a family of boundaries to partition that space), we make sure information is extractible, i.e. we can reach good accuracy using a classifier which splits the space using these boundaries.

(IV) Causal intervention These boundaries can then be used to perform a causal intervention. Either we can remove information and see how it impacts performance on the behavior task, or set the value to another state, for which we expect another behavioral response.

If one wishes to apply this generic procedure to find transformations which play a causal role over the model's predictions, this only needs to be slightly modified. (II) and (III) can be replaced by a step in which one formulates an algorithmic hypothesis. In addition to representing properties, we would make an assumption for how information about that property is passed or transformed in the network. Picking a component of the network responsible for that operation, (IV) would be applied equally, by disabling that component. Note that choosing a location for an encoding when only looking for representations is already an algorithmic assumption, that information is copied (or transferred) in that location, even if it does not seek the transformation by which this is made possible.

7.5 Conclusion

In this chapter, we discussed the importance of understanding what information NLMs encode in their intermediate layers, and which computations allow this information to be transferred, that is which transformations make it readable by subsequent layers. To do so, we adopt a functionalist approach to probing representations and layers, in which the properties encoded in layers and the role of inner operations should be examined on the basis of their causal role in producing certain surface behavior. We defined criteria to assess that an encoding is meaningful to the model and used to make predictions. These criteria naturally define a methodology to find linguistic representations in the model's inner structure, which are the building blocks necessary to providing a mechanistic account for the abilities of NLMs. This framework provides a practical procedure to connect the implementational and the algorithmic levels of understanding of a given NLM's abilities. Equipped with this background, we next present a case study on grammatical number, in which we seek representations of a property which are used by a NLM on the number agreement task.

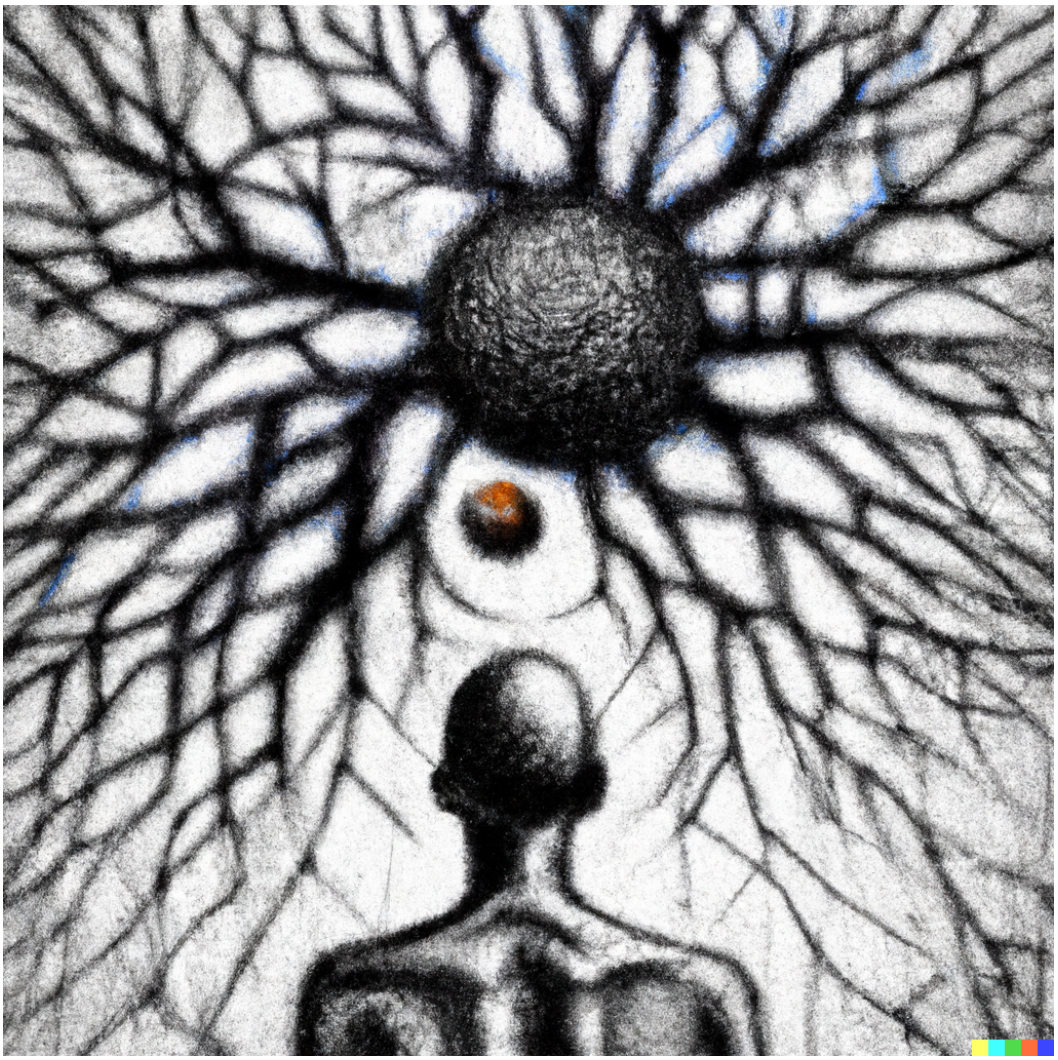


Figure 7.1: "An artistic painting of a neural network with a silhouette looking up in front, drawn using charcoal", K.L. x DALL·E 2

Chapter 8

From Representations to Behavior: Probing for the Usage of Linguistic Knowledge

Goals

Equipped with the framework presented in the previous chapter, we now investigate how a given model represents its linguistic knowledge, by looking for encodings which play a causal role on the model's outputs, i.e. functional encodings. Specifically, we open the black box model and seek representations used to produce behavior reflecting the capture of linguistic abilities. We will apply empirically our functionalist view, in which we defined desiderata to ascertain that an encoding of a given linguistic property is meaningful to the model and not spurious. To do so, we formulate simple algorithmic assumptions, regarding the location where functional encodings of a given property lie in the network, and attempt to remove that property from the model's representations. We contend that, if an encoding is used by the model, its removal should harm the model's performance on a behavioral task which requires it. As a case study, we take a look at whether and how BERT encodes an abstract representation of grammatical number in its layers, and how it mobilizes such encoding on the number agreement task. We show that BERT relies on a linear encoding of grammatical number to produce the correct behavioral output, ruling out that it relies on shallow memorization processes on the number agreement task. We also find that BERT uses a separate encoding of grammatical number for nouns and verbs. Finally, we identify in which layers information about grammatical number is transferred from a noun to its head verb.

Contents

8.1 Introduction

Under our usage-based perspective, our goal is to find a functional encoding—i.e., an encoding that the model actually uses when making predictions. We achieve this by relying on a combination of the paradigms discussed in section 2.3. To this end, we first need a behavioral task that requires the model to use information about the target property. We then perform a causal intervention to try to remove this property's encoding. We explain both these components in more detail now.

Behavioral Task. We first require a behavioral task which can *only* be solved with information about the target property. The choice of task and target property are thus co-dependent. Further, we require our model to perform well on this task. On one hand, if the model cannot achieve high performance on the behavioral task, we cannot be sure the model encodes the target property, e.g., grammatical number, at all. On the other hand, if the model can perform the task, it must make use of the property.

Causal Intervention. Our goal in this work is to answer a causal question: Can we identify a property's functional encoding? We thus require a way to intervene in the model's representations. If a model relies on an encoding to make predictions, removing it should harm the model's performance on the behavioral task. If follows that, by measuring the impact of our interventions on the model's behavioral output, we can assess whether our model was indeed decoding information from our targeted encoding.

Case study: Probing for the Usage of Grammatical Number As a first case study of this methodology, we focused on how BERT encodes grammatical number, and on how it uses this encoding to solve the number agreement task. Experimentally, We found that BERT relies on a linear encoding of grammatical number to produce the correct behavioral output. We also find that BERT uses a separate encoding of grammatical number for nouns and verbs. Finally, we identify in which layers information about grammatical number is transferred from a noun to its head verb. Our analysis of grammatical number allows us to track how a simple morpho-syntactic property, grammatical number, is encoded across BERT's layers and where it is transferred between them before being used on the model's predictions. Using carefully chosen causal interventions, we demonstrate that forgetting number information impacts both: (i) BERT's behavior and (ii) how much information is extractable from BERT's inner layers. Further, the effects of our

interventions on these two, i.e., behavior and information extractability, line up satisfyingly, and reveal the encoding of number to be orthogonal for nouns and verbs. This finding is surprising given that number is a linguistic property common to both part-of-speech.

This first case study is encouraging as we succeeded in making a causal connection between the model's representations of a linguistic property and its output behavior. This holds as strong evidence for the acquisition of linguistic abstractions.

8.2 Grammatical Number and its Usage

The empirical portion of this paper focuses on a study of how BERT encodes grammatical number in English. We choose number as our object of study because it is a well understood morpho-syntactic property in English. Thus, we are able to formulate a simple algorithmic hypothesis about how BERT passes information about number when performing number agreement: we look for encodings of grammatical number assuming that information about that property is simply read from intermediate representations at the cue's position and transferred to the target's intermediate representations. If this algorithmic hypothesis is true, we can simply try to perform causal interventions at these positions to track information transfer.

We use the same stimuli as in chapter 4 [Linzen, 2016] for number agreement, as our behavioral task.

8.2.1 Related Work on Grammatical Number

A number of studies have investigated how grammatical number is encoded in neural language models.[1] Most of this work, however, focuses on diagnostic probes [Klafka, 2020; Torroba Hennigen, 2020]. These studies are thus agnostic about whether the probed models actually use the encodings of number they discover. Some authors, however, do consider the relationship between how the model encodes grammatical number and its predictions. Notedly, Giulianelli et al. [Giulianelli, 2018] use a diagnostic probe to investigate how an LSTM encodes number in a subject–verb number agreement setting. Other approaches [Lakretz, 2019; Finlayson, 2021] have been proposed to apply interventions at the neuron level and track their effect on number agreement. In this work, we look for functional encodings of grammatical number—encodings which are in fact used by our probed model when solving the task.

8.3 From Encoding to Usage

We discuss how to identify and remove an encoding from a set of contextual representations using diagnostic probing. Our use of diagnostic probing is thus twofold. For a model to rely on an encoding of our property when making predictions, the property must be encoded in its representations. We thus first use diagnostic probing to measure the amount of information a representation contains about the tar-

[1] We focus on grammatical number here. There is, however, also a vast literature investigating how BERT encodes number from a numeracy point of view [Wallace, 2019; Geva, 2020; Spithourakis, 2018].

get linguistic property. In this sense, diagnostic probing serves to sanity-check our experiments—if we cannot extract information from the representations, there is no point in going forward with our analysis. Second, we make use of diagnostic probing in the context of amnesic probing [Elazar, 2021], which allows us to determine whether this probe finds a functional or a spurious encoding of the target property.

8.3.1 Estimating Extractable Information

In this section, we discuss how to estimate the amount of information about grammatical number that is extractable from our probed model's representations. The crux of our analysis relies on the fact that the encoding extracted by diagnostic probes is not necessarily the functional encoding used by our probed model. Nevertheless, for a model to use a property in its predictions, this property should at least be extractable, which is true due to the data processing inequality. In other words, extractability is a necessary, but not sufficient, condition for a property to be used by the model. While a probing classifier's performance – taken as an estimate of the information present in a probed architecture – is often measured with accuracy metrics, information-theoretic views on probing have been proposed, e.g. as extracting mutual information Pimentel et al. [Pimentel, 2020b], computing minimum description length [Voita, 2020], or usable information Hewitt et al. [Hewitt, 2021].

Given a labeled sentence (typically one which would be used in our NA task), a specific number label is associated with a given the cue-target pair. Given that we wish to extract information about number from a set of intermediate representations (e.g. at a given layer, and at either the cue's position or at the target's), let us denote R the representation-valued random variable and N the number-valued random variable. Those simply map each randomly sampled label sentence into the targeted representation and number.

Formally, the mutual information between R and N is defined as:

$$\mathrm{I}(R; N) = \mathrm{H}(N) - \mathrm{H}(N \mid R) \tag{8.1}$$

Theoretically, this value represents the amount of information about obtained about number (that is, the label associated with an input sentence's cue-target pair) by observing the model's representations. However, if we see probing as estimating mutual information Pimentel et al. [Pimentel, 2020b], any injective function has the same mutual information as the input. It follows that, theoretically, under the mild assumption that any two representations for any two sentences are different,

number can be perfectly retrieved from representations. This isn't useful in our case, as any injective model (even a randomly initialized one) theoretically leads to perfect extractability of number.

Let us get back to our probing classifiers. The family of functions used to predict number given representations imposes certain constraints on how number is extracted from layers. For instance, if one uses a linear classifier, its accuracy is indicative of how well it is possible predict number linearly from representations, i.e. how well a linear hyperplane can separate representations of singular instances and plural ones. This constraint over the way information is structured in the representation space can be naturally integrated into mutual information by using a modified version, $\mathcal{V}$-information [Xu, 2020b]. In our case, the $\mathcal{V}$-information between R and N is the amount of information about number that can be extracted from representations using a certain family of functions $\mathcal{V}$.

Broadly speaking, [Xu, 2020a] defines $\mathcal{V}$-information as:

$$\mathrm{I}_{\mathcal{V}}(R \to N) = \mathrm{H}_{\mathcal{V}}(N) - \mathrm{H}_{\mathcal{V}}(N \mid R) \tag{8.2}$$

where $\mathcal{V}$ is a variational family determined by our diagnostic probe, and the $\mathcal{V}$-entropies are defined as:

$$\mathrm{H}_{\mathcal{V}}(N) = \inf_{q \in \mathcal{V}} \mathbb{E}_{n \sim N} \log \frac{1}{q(n)} \tag{8.3}$$

$$\mathrm{H}_{\mathcal{V}}(N \mid R) = \inf_{q \in \mathcal{V}} \mathbb{E}_{n, \mathbf{r} \sim N, R} \log \frac{1}{q(n \mid \mathbf{r})} \tag{8.4}$$

To compute this, we must first define a variational family $\mathcal{V}$ of interest; which we define as the set of linear transformations. In these conditions, we are estimating linearly extractible information using the classifier's $\mathcal{V}$-information.

In eq. (8.3), the first term $\mathrm{H}_{\mathcal{V}}(N)$ represents the minimal log-likelihood in predicting number which can be achieved by a model q in $\mathcal{V}$. In our case, linear models span the whole range of possible probabilities over N, so this $\mathcal{V}$-entropy is simply equal to the entropy of N, a quantity determined by the proportion of singulars and plurals in the dataset. The second term $\mathrm{H}_{\mathcal{V}}(N \mid R)$ represents the minimal expected log-likelihood of a model in $\mathcal{V}$ predicting number from representations. The minimum value this term can take is 0, if the predictor always assigns a probability $q(n) = 1$ to the correct number. The maximum value it can take is $\mathrm{H}_{\mathcal{V}}(N)$ itself, if R adds no information, that is if the classifier guesses number randomly.

According to eq. (8.2), $\mathcal{V}$-information can therefore vary in the range $[0; \mathrm{H}_{\mathcal{V}}(N)]$. As this higher bound depends on the distribution of N, we can define a more interpretable value (which we call the **$\mathcal{V}$-uncertainty**) by normalizing $\mathcal{V}$-information,

as:

$$\mathrm{U}_{\mathcal{V}}(R \to N) = \frac{\mathrm{I}_{\mathcal{V}}(R \to N)}{\mathrm{H}_{\mathcal{V}}(N)} \tag{8.5}$$

This quantity in turn varies in $[0; 1]$.

Note that a perfect classifier reaching 100% accuracy will not necessarily assign 1 probabilities to the correct outcomes and 0 probabilities to the wrong ones (in the latter case, $\mathrm{H}_{\mathcal{V}}(N \mid R)$ would be equal to zero), so this measurement is more informative on the probability distribution learned by a probing classifier.

We also note that the $\mathcal{V}$-information lower-bounds the mutual information: $\mathrm{I}_{\mathcal{V}}(R \to N) \leq \mathrm{I}(R; N)$. It follows that, if we can extract some $\mathcal{V}$-information from a set of representations, they contain at least the same amount of information in Shannon's (1948) more classic sense.

Further, if we denote our analyzed model's (i.e., BERT's) hidden representations as:

$$\mathbf{r}_{t,l} = \mathrm{BERT}(\texttt{sentence})_{t,l} \tag{8.6}$$

we define a linear diagnostic probe as:

$$p_{\boldsymbol{\theta}}(n_t = \textsc{sing} \mid \texttt{sentence}) = \sigma(\boldsymbol{\theta}^{\intercal}\, \mathbf{r}_{t,l} + b) \tag{8.7}$$

where $\mathbf{r}_{t,l} \in \mathbb{R}^{768}$, t is a sentence position and l is a layer, n_t is the binary number label associated with the word at position t, σ is the sigmoid function, $\boldsymbol{\theta}$ is a real-valued column parameter vector and b is a bias term. In this case, we can define our variational family as $\mathcal{V} = \{p_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \mathbb{R}^{768}\}$.

8.3.2 Intervening on the Representations

We now discuss how we perform a causal intervention to prevent the analyzed model from using a given encoding. The goal is to damage the model and make it "forget" a property's information. This allows us to analyze whether that encoding actually influences the probed model's predictions—i.e., whether this encoding is indeed functional. To this end, we employ amnesic probing [Elazar, 2021].[2] In short, we first learn a linear diagnostic classifier, following eq. (8.7). If this linear classifier is able to extract information about number from representations, it means that it is possible for a linear hyperplane to separate intermediate representation in space to some degree (with singulars on the one side and plurals on the other). To

[2] In particular, this intervention consists in applying iterative null-space projection to the representations, originally proposed by Ravfogel et al. [Ravfogel, 2020]. We note that Ravfogel et al. [Ravfogel, 2022a; Ravfogel, 2022b] recently proposed two new methods to remove information from a set of representations.

damage representations, we wish to erase the information they bear about number. To do so, we compute the projector onto the kernel (or null) space of this linear transform $\boldsymbol{\theta}$, shown below:

$$\mathbf{W}_{\text{null}} = \mathbf{I} - \frac{\boldsymbol{\theta}\boldsymbol{\theta}^{\intercal}}{||\boldsymbol{\theta}||_2^2} \tag{8.8}$$

When applied to representations, this projector will map each vector onto the closest point, on the plane which separates singulars from plurals. This process therefore removes the information that our classifier was relying on. Applying this procedure once might not be sufficient, as the obtained damaged vectors could still bear some information about number encoded in another direction not captured by our first hyperplane. Indeed, the projection above only removes the information that was encoded in one direction of the intermediate space. As a comparison point, intermediate representations for the model under investigation in this study have 768 dimensions. It is therefore reasonable to expect that number information is present in several directions.

Hence, we iterate this process, and we store a set of parameter vectors $\boldsymbol{\theta}^{(k)}$ and their associated projectors $\mathbf{W}_{\text{null}}^{(k)}$ until we are unable to extract the property. The composition of these projectors makes it possible to remove all linearly extractable number information from the analyzed representations. We can then apply the resulting composition to the said representations to get a new set of vectors:

$$\mathbf{r}_{t,l}^{(k)} = \mathbf{W}_{\text{null}}^{(k)} \cdots \mathbf{W}_{\text{null}}^{(2)} \mathbf{W}_{\text{null}}^{(1)} \mathbf{r}_{t,l} \tag{8.9}$$

After learning the projectors, we can measure how erasing a layer's encoding impacts: (i) the subsequent layers, and (ii) our model's performance on the number agreement task. Removing a functional encoding of grammatical number should cause a performance drop on the number agreement task. Further, looking at both (i) and (ii) allows us to make a connection between the amount of information we can extract from our probed model's layers and its behavior. We are thus able to determine whether the encodings revealed by our diagnostic probes are valid from a usage-based perspective—are they actually used by the probed model on a task that requires them?[3]

[3]Our method differs from amnesic probing mostly in that all our analyses are based on a behavioral task which we know *a priori* to require the property we investigate.

8.4 Experimental Setup

Data. We perform our analysis on Linzen et al.'s (2016) number agreement dataset, which consists in sentences extracted from Wikipedia. In this dataset, each sentence has been labeled with the position of the cue and target, along with their grammatical number. We assume here that this dataset is representative of the number agreement task; this may not be true in general, however.

Model. In our experiments, we probe BERT [Devlin, 2019a].[4] Specifically, BERT is a bidirectional transformer model with 12 layers, trained using a masked language modeling objective. As BERT has been shown to perform well on this dataset [Goldberg, 2019], we already know that our probed model passes our first requirement; BERT does use number information in its predictions.

Distinguishing Nouns and Verbs. While number is a morpho-syntactic property common to nouns and verbs, we do not know *a priori* if BERT relies on a single subspace to encode number in their representations. Though it is possible for BERT to use the same encoding, it is equally plausible that each part of speech would get its own number encoding. This leads us to perform our analyses using independent sets of representations for nouns and verbs; as well as a mixed set which merges both of them. Further, verbs are masked when performing the number agreement task, so their representations differ from those of unmasked verbs. Ergo, we analyze both unmasked, and masked tokens at the target verb's position—which for simplicity we call verbs and masked verbs, respectively. This leaves us with four probed categories: nouns, verbs, masked verbs, and mixed.

[4] We focus on `bert-base-uncased`, as implemented in the transformers library [Wolf, 2020].

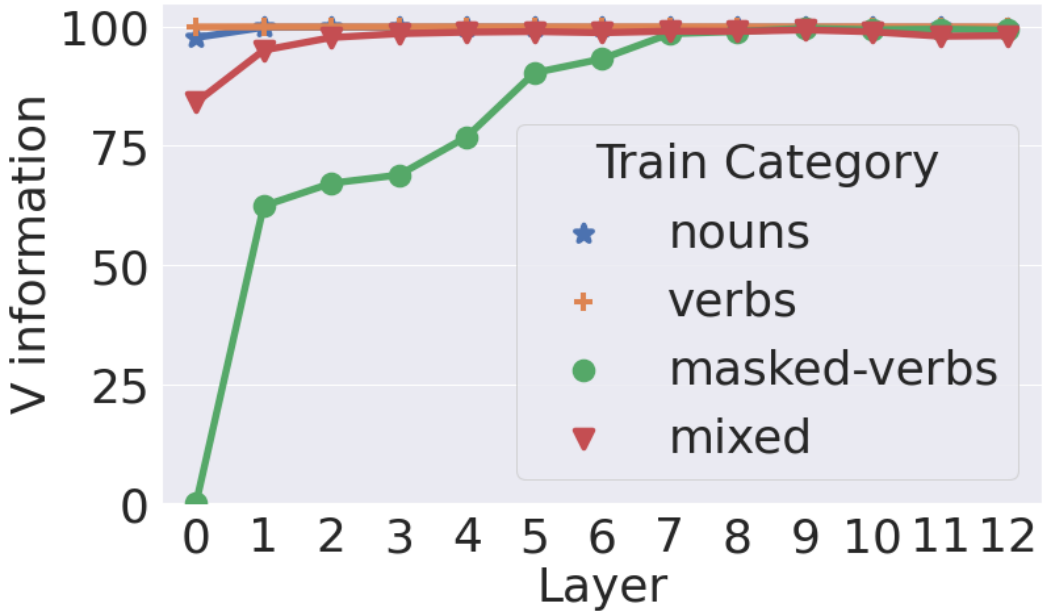


Figure 8.1: The amount of $\mathcal{V}$-information BERT representations hold about grammatical number, as estimated with linear diagnostic probes.

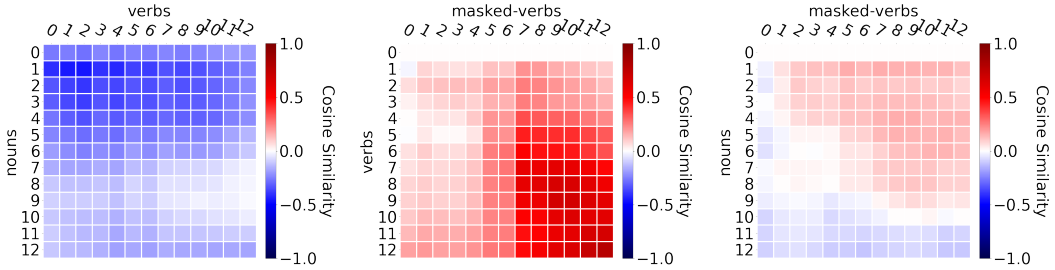


Figure 8.2: Cosine similarities between the learned parameter vectors of our diagnostic probes. The matrices display similarities between different layers, and across categories.

8.5 Experiments and Results

In our experiments, we focus on answering two questions: (i) How is number information encoded in BERT's representations? and (ii) How is number information transferred from a noun to its head verb for the model to use it on the behavioral task? We answer question (i) under both extractability and usage-based perspectives. In section 8.5.1, we present our sanity-check experiments that demonstrate that grammatical number is indeed linearly extractable from BERT's representations. In section 8.5.2 and section 8.5.3, we use our causal interventions: we identify BERT's *functional encodings* of number; and analyze whether these functional encodings are shared across parts of speech. Finally, in section 8.5.5 and section 8.5.6 we investigate question (ii), taking a closer look at the layers in which information is passed.

8.5.1 What do diagnostic probes say about number?

fig. 8.1 presents diagnostic probing results in all four of our analyzed settings.[5] *A priori*, we expect that verbs' and nouns' representations should already contain a large amount of $\mathcal{V}$-information about their grammatical number at the type-level. As expected, we see that the $\mathcal{V}$-information is near its maximum for both verbs and nouns in all layers; this means that nearly 100% of the uncertainty about grammatical number is eliminated given BERT's representations. Further, the mixed category results also reach a maximal $\mathcal{V}$-information, which indicates that it is possible to extract information linearly about both categories at the same time. On the other hand, the $\mathcal{V}$-information of masked verbs is 0 at the non-contextual layer and it progressively grows as we get to the upper layers.[6] As we go to BERT's deeper layers, the $\mathcal{V}$-information steadily rises, with nearly all of the original uncertainty eliminated in the mid layers. This suggests that masked verbs' representations acquire number information in the first 7 layers.

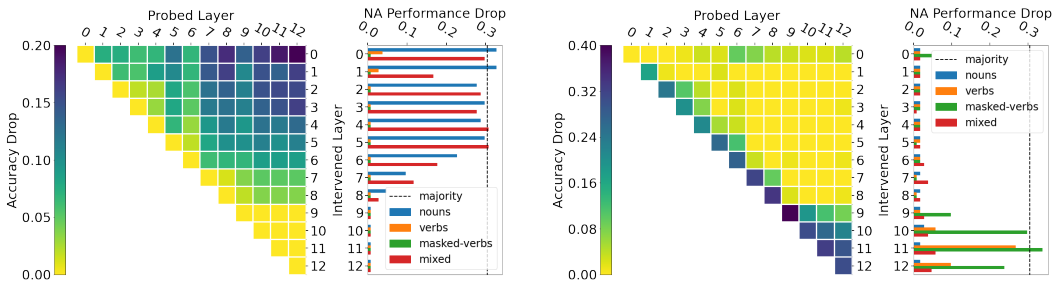
However, from these results alone we cannot confirm whether the encoding that nouns and verbs use for number is shared or disjoint. We thus inspect the encoding found by our diagnostic probes, evaluating the cosine similarity between their learned parameters $\boldsymbol{\theta}$ (ignoring the probes' bias terms b here). If there is a single shared encoding across categories, these cosine similarities should be high. If not, they should be roughly zero. fig. 8.2 (left) shows that nouns and verbs might encode number along different directions. Specifically, noun representations on the first 6 layers seem to have a rather opposite encoding from verbs, while the later layers are mostly orthogonal. Further, while masked verbs and verbs do not seem to share an encoding in the first few layers, they are strongly aligned from layer 6 on (fig. 8.2; center).

We now know that there are encodings from which we can extract number from nouns and verbs, and that these encodings are disjoint. However, we still do not know whether the encoding is spurious or functional.

Diagnostic Probing Cross-Evaluation In addition to comparing the angles of our diagnostic probes trained on different categories, we performed cross-evaluation of our trained diagnostic probes. In this setting, we trained probes on one category and tested them on the others. fig. 8.4 presents our cross-evaluation results. The performance of probes evaluated in one category, but trained on another, again

[5] We further present accuracy results in section 8.5.1.

[6] We note that, in fig. 8.1, layer 0 corresponds to the non-contextual representations (i.e. the word embeddings before being summed to BERT's position embeddings). Non-contextual layers thus contain no information about the number of a masked verb, as the mask token contains no information about its replaced verb's number.



(a) Information loss (measured at the target) after erasing nouns' number information at the cue position.

(b) NA performance drop after erasing number at the cue position.

(c) Information loss (measured at the target) after erasing masked verbs' number at the target position.

(d) NA performance drop after erasing number at the target position.

Figure 8.3: Effect of our causal interventions on information recovery in subsequent layers (triangular matrices) and on the number agreement task (bar charts). Information loss is measured at the target position by a diagnostic probe; we display the probing accuracy drop compared to when no intervention was performed. The legend in the bar charts indicates what category the amnesic projectors have been trained on. Majority represents the difference in performance between BERT and a trivial baseline which always guesses the majority label.

suggests that BERT encodes number differently across lexical categories. Interestingly, in the lower layer, the probe tested on nouns (top-left) guesses the wrong number systematically when trained on verbs, and vice-versa (top-right). This can be due to token ambiguity, as some singular nouns (e.g. "hit") are also plural verbs. This is further evidence that the encoding might be different for nouns and verbs, though this analysis still cannot tell us whether this is true from our usage-based perspective. Additionally, the mixed results (fig. 8.4; bottom-right), show it is possible to linearly separate both nouns and verbs with a single linear classifier trained on both categories, reaching perfect performance on all other categories, including masked-verbs (bottom-left).

8.5.2 Does the model use these encodings?

The patterns previously observed suggest there is a linear encoding, from which grammatical number can be extracted from BERT's representations. We, however, cannot determine whether these encodings are actually those used by the model to make predictions. We now answer this question taking our proposed usage-based perspective, studying the impact of linearly removing number information at

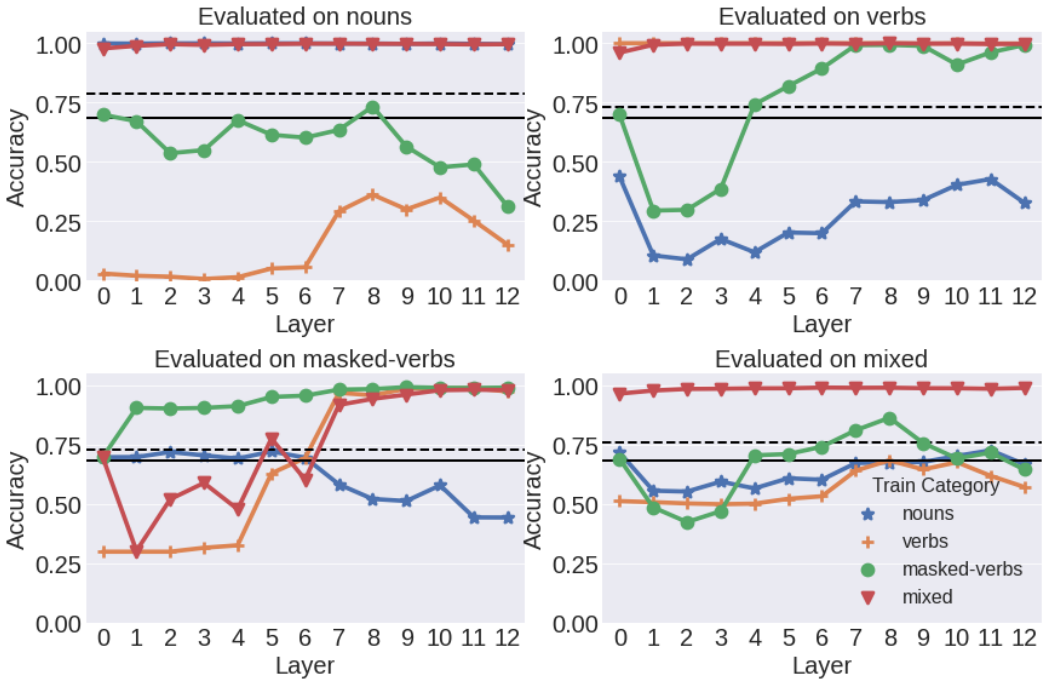


Figure 8.4: Probes cross-evaluation. Each plot corresponds to a test category, and colors correspond to the category used for training. Solid lines represent the percentage of majority-class (plural vs singular) tokens; dashed lines represent the percentage of majority-class tokens per lemma, averaged across lemmas.

both the cue and target positions.[7] We evaluate the model's change in behavior, as evaluated by its performance on the number agreement (NA) task.

fig. 8.3a and fig. 8.3c show the decrease in how much information is extractable at the target position after the interventions are applied. fig. 8.3b and fig. 8.3d show BERT's accuracy drops on the NA task (as measured at the output level). By comparing these results, we find a strong alignment between the information lost across layers and the damage caused to the performance on the task—irreversible information losses resulting from our intervention are mirrored by a performance decrease on the NA task. This alignment confirms that the model indeed uses the linear information erased by our probes. In other words, we have found the probed property's functional encoding.

8.5.3 Does BERT use the same encoding for verbs and nouns?

We now return to the question of whether nouns and verbs share a functional encoding of number, or whether BERT encodes number differently for them. To answer this question, we investigate the impact of removing a category's encoding from

[7]The number of dimensions removed by our amnesic projectors in each layer and category is presented in table. 8.1.

another category, e.g. applying an amnesic projector learned on verbs to a noun. In particular, we measure how these interventions decrease BERT's performance in our behavioral task. figs. 8.3b and 8.3d presents these results.

We observe that each category's projector has a different impact on performance depending on whether it is applied to the cue or the target. fig. 8.3b, for instance, shows that using the verb's, or masked verb's, projector to erase information at the cue's (i.e., the noun's) position does not hurt the model. It is similarly unimpactful (as shown in fig. 8.3d) to use the noun's projectors to erase a target's (i.e., the masked verb's) number information. Further, the projector learned on the mixed set of representations does affect the cue, but has little effect on the target. Together, these results confirm that BERT relies on rather distinct encodings of number information for nouns and verbs.[8]

These experiments allow us to make stronger claims about BERT's encoding of number information. First, the fact that our interventions have a direct impact on BERT's behavioral output confirms that the encoding we erase actually bears number information *as used by the model when making predictions*. Second, the observation from fig. 8.2—that number information could be encoded orthogonally for nouns and verbs—is confirmed from a usage-based perspective. Indeed, using amnesic probes trained on nouns has no impact when applied to masked verbs, and amnesic probes trained on verbs have no impact when applied to nouns. These fine-grained differences in encoding may affect larger-scale probing studies if one's goal is to understand the inner functioning of a model. Together, these results invite us to employ diagnostic probes more carefully, as the encoding found may not be actually used by the model.

8.5.4 Removing random directions from representations

Removing directions from intermediate spaces could harm the model's normal functioning independently from removing our targeted property. We thus run a control experiment proposed by Elazar et al. [Elazar, 2021], removing random directions at each layer (as opposed to the specific directions found by our amnesic probes). This experiment allows us to verify that the observed information loss and decrease in performance do not only result from removing too many directions. To do so, we remove an equal number of random directions at each layer. The results

[8] A potential criticism of amnesic probing is that it may remove more information than necessary. Cross-testing our amnesic probes, however, results in little effect on BERT's behavior. It is thus likely that they are not overly harming our model. Further, we also run a control experiment proposed by Elazar et al., removing random directions at each layer (instead of the ones found by our amnesic probes). These results are displayed in the appendix in table. 8.1.

are displayed in table. 8.1 and show that removing randomly chosen directions has little to no effect compared to our targeted causal interventions.

Layer	0	1	2	3	4	5	6	7	8	9	10	11	12
Masked Verbs													
Number of Directions	1	13	15	26	30	17	21	44	24	22	22	26	33
Loss in Layers	0.0	0.33	0.3	0.34	0.34	0.34	0.37	0.38	0.42	0.39	0.41	0.41	0.41
Loss in Layers (Random)	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NA Performance Drop	0.04	0.01	0.01	0.01	0.01	0.0	0.01	0.0	0.0	0.09	0.29	0.33	0.23
NA Performance Drop (Random)	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.01
Nouns													
Number of Directions	17	51	33	70	22	37	48	52	64	39	22	39	26
Loss in Layers	0.49	0.37	0.39	0.38	0.37	0.38	0.43	0.4	0.43	0.4	0.37	0.41	0.4
Loss in Layers (Random)	0.0	0.0	0.0	0.02	0.01	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0
NA Performance Drop	0.32	0.32	0.27	0.29	0.28	0.29	0.22	0.09	0.04	0.0	0.0	0.0	0.0
NA Performance Drop (Random)	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 8.1: Causal intervention results using both the default or random directions. For each category, we display the number of directions removed in each layer, the information loss resulting from amnesic interventions in each layer and the effect on the NA task. We also display the loss in layers and performance decrease on NA resulting from the removal of random directions as a control experiment.

8.5.5 Where does number erasure affect the model?

Once we have found which encoding the model uses, we can pinpoint at which layers the information is passed from the cue to the target. To that end, we observe how interventions applied in each layer affect performance. We know number information must be passed from the cue to the target's representations—otherwise the model cannot solve the task. Therefore, applying causal interventions to remove number information should harm the model's behavioral performance when applied to: (i) the cue's representations before the transfer occurs; (ii) the target's representations after the transfer occurred.

Interestingly, we observe that target interventions are only harmful after the 9^{th} layer; while noun interventions only hurt up to the 8^{th} layer (again, shown in fig. 8.3). This suggests that the cue passes its number information in the first 8 layers, and that the target stops acquiring number information in the last three layers. While we see a clear stop in the transfer of information after layer 8, fig. 8.3a shows that the previous layers' contribution decreases slowly up to that layer. We thus conclude that information is passed in the layers before layer 8; however, we concede that our analysis alone makes it difficult to pinpoint exactly which layers.

8.5.6 Where does attention pruning affect number transfer?

Finally, in our last experiments, we complement our analysis by performing attention removal to investigate how and where information is transmitted from the cue to the target position. This causal intervention first serves the purpose of identifying the layers where information is transmitted. Further, we wish to understand whether information is passed directly, or through intermediary tokens. To this end, we look at the effect on NA performance after: (i) cutting direct attention from the target to the cue at specific layers, (ii) cutting attention from all tokens to the cue (as information could be first passed to intermediate tokens, which the target could attend to in subsequent layers).[9] Specifically, we perform these interventions in ranges of layers (from layer i up to j).

Formally, let $\mathbf{A}^{l,h} \in \mathbb{R}^{T \times T}$ be a model's attention weights for a given layer $1 \leq l \leq 12$, a head $1 \leq h \leq 12$, and a sentence with length T.[10] Further, we define a binary mask matrix $\mathbf{M}^l \in \{0,1\}^{T \times T}$. We can now perform an intervention by masking the attention weights of all heads in a layer. Given a layer l:

$$\widehat{\mathbf{A}}^{l,h} = \mathbf{A}^{l,h} \circ \mathbf{M}^l, \quad 1 \leq h \leq 12 \tag{8.10}$$

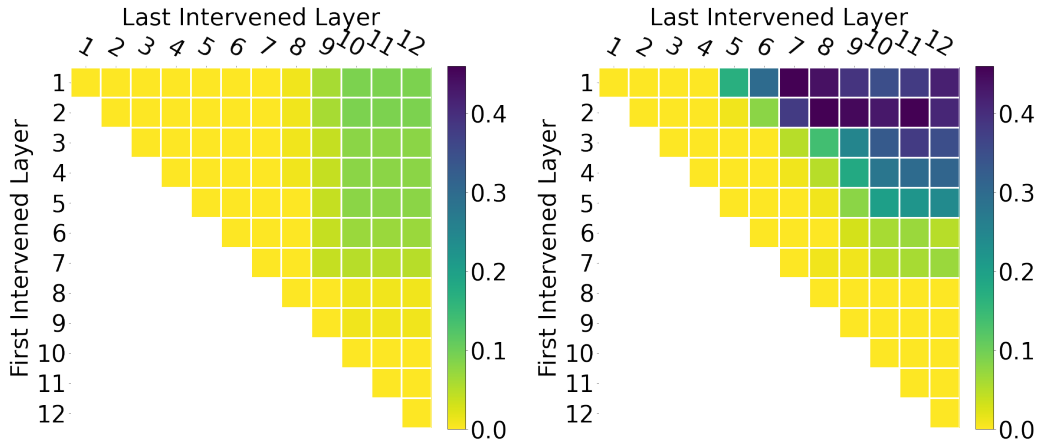
where $\circ$ represents an elementwise product between two matrices. Now assume a given sentence with cue position p_c, and with target position p_t. In our intervention (i), matrix $\mathbf{M}^l$ is set to all 1's except for $\mathbf{M}^l_{p_t,p_c} = 0$; the target's attention to the cue is thus set to zero. In intervention (ii), we set $\mathbf{M}^l_{:,p_c} = 0$ and other positions to 1, which removes all attention to the cue.

We report number agreement accuracy drops in fig. 8.5.

The diagonals from this figure show that removing attention from a single layer has basically no effect. Further, cutting attention from layers 6 to 10 suffices to observe near-maximal effect for direct attention. Interestingly, it is at those layers where we see a transition from it being more harmful to apply amnesic projectors to the cue or to the target (in section 8.5.5). However, while those layers play a role in carrying number information to the target position, the drop is relatively modest when cutting only direct attention ($\approx 10\%$). Cutting attention from all tokens to the cue, in turn, has a significant effect on performance (up to $\approx 40\%$), and is maximal for layers 2 to 8. This first suggests that, while other clues in the sentence could indicate the target verb's number (such as a noun's determiner), the noun itself is the core source of number information. Further, this shows the target can get in-

[9] Klafka et al. [Klafka, 2020], for instance, showed that number information of a given token was distributed to neighboring tokens in the upper layers

[10] Our analyzed model, BERT base, has 12 layers, and 12 attention heads in each layer.



(a) Removing attention from the target to the cue only

(b) Removing attention from all tokens to the cue

Figure 8.5: Number agreement task performance drops after performing attention removal. The attention cut is performed on a range of layers. Rows and columns, respectively, represent the first and last intervened layer.

formation from intermediate tokens, instead of number being passed exclusively through direct attention.[11]

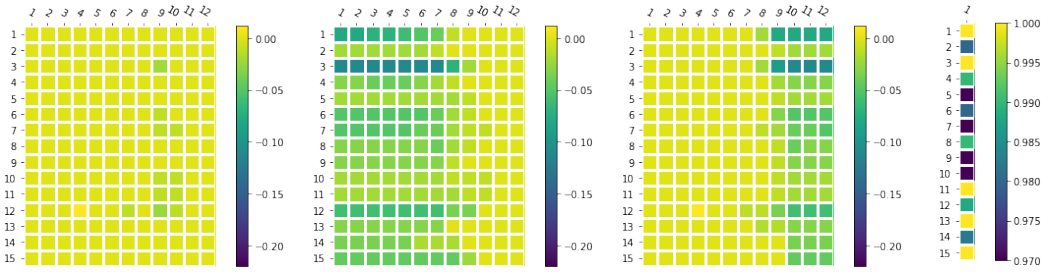
8.5.7 The effect of linear distance

We complement the previous analysis by testing whether the linear distance between the cue and the target influences the effect of attention removal. fig. 8.6a shows that cutting attention from one layer has negligible effect over performance regardless of distance, which is in line with results from the diagonals of fig. 8.5. When cutting attention from several subsequent layers (fig. 8.6b), we observe that performance drop depends on the linear position, and decreases when the model is not faced with short-range agreement. This is not surprising as many of the attention maps attend to surrounding tokens [Kovaleva, 2019]. Extensive analysis targeting individual attention heads (instead of cutting all attention from a given layer) is necessary to examine both their contribution to the model's successes, and their dependence on linear distance.

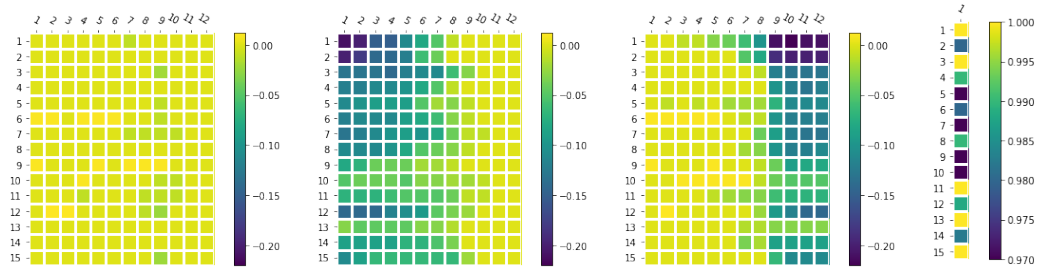
8.5.8 Wrapping up

Throughout this series of analyses, we shed light on the flow of number information in BERT when successfully processing number agreement. As the task involves guessing the correct number for the target at the decision layer, based on the cue's

[11] See section 8.5.7 for further experiments.



(a) Cutting attention from the target to the cue only



(b) Cutting attention from all tokens to the cue

Figure 8.6: Agreement task performance drops resulting from attention interventions, as a function of linear distance between the cue and the target. The rows represent distances (from 1 to 15) and columns represent the intervened layers. Three conditions are tested: cutting attention only at current layer (left), cutting attention starting from current layer up to the last one (middle) and from the first layer to current layer (right). The color map on the far right represent agreement scores without intervention for each linear distance.

number, given at the input layer, we hypothesized that this property was encoded in both positions across layers of the architecture. We additionally assumed that this encoding was linear, and perhaps different for both part-of-speech – i.e. encoded in different linear subspaces. By systematically testing the effect of eliminating this information from intermediate representations at these positions, we were able to find functional encodings of grammatical number, and to prove that they were indeed different across lexical categories, as information loss in layers mirror performance loss on the agreement task. By observing where number erasure applied at the source of information (the cue's representations) and the receiver (the target's), we concluded that this transfer occurred before layer 9. By further applying attention interventions, we were able to show that this transfer is distributed across layers and takes place from layers 2 to 8. We additionally showed that information is passed through intermediate tokens, and not through direct attention. While this mechanistic analysis allows us to pinpoint where number information is encoded and flows in the network, we are yet to understand the role of intermediate tokens in passing information about number from the cue to the target, and the location or nature of such tokens. Additionally, complementary analyses are needed to investigate how BERT abstracts away from attractors when applying this mechanism.

8.6 Discussion

Information Extractability and Usage Following the maxim that correlation is not causation, we carefully designed our analyses to give strong evidence for a property being encoded in a model's representations. We have shown that diagnostic probes uncover encodings that might not necessarily be useful to the model's predictions, as such methods are only correlational. Indeed, we show that BERT decodes grammatical number from orthogonal subspaces for nouns and verbs—even though simple linear classifiers can separate a population of mixed vectors. This in turn raises the question of whether complexity alone, a feature much discussed in the literature [Pimentel, 2020a; Voita, 2020], is enough to evaluate probes, as finding a simple encoding is not enough evidence that the encoding is actually useful to the model.

From Linguistic Properties to Encoding Using a pipeline similar to ours, [Ravfogel, 2021] recently investigated whether a model was solving the number agreement task in a manner that is linguistically plausible. In this paper, we have shown that even a relatively simple property's encoding (i.e., grammatical number's) can hide subtleties which only surface after carrying out a fine-grained analysis. In-

deed, despite the fact that number is a single morpho-syntactic property common to nouns and verbs, we show that BERT uses separate representations for each category. This fine-grained difference in representation informs us that one should be cautious when choosing a property to probe a given model. Indeed the model could be representing the latter in a subtler way than what a researcher would initially expect.

Understanding BERT's Inner Workings Throughout this work, our results allow us to identify how number is encoded by our model, and where it is transferred across token positions—as confirmed by behavioral observations. Our results point towards number information being transmitted from cue to target up to the 9^{th} layer. Our results also reveal that information transfer does not result from direct attention only, which confirms previous observations that information is distributed across neighboring tokens in the sentence [Klafka, 2020].

It is not easy to dissect the inner mechanisms which support large pre-trained models' impressive abilities. However, identifying how information is encoded and where it is transferred across layers reduces the scope of where to look for answers, as it sheds light on the algorithmic processes which produce their decisions. Further, with more reliable accounts of the encoding structures used by a model when decoding a property, we might be able to operationalize a larger set of probing questions. Given a better understanding of how BERT structures number information, for instance, we can now try to ask how it identifies the subject a verb should get it from, and get a more complete picture of the algorithmic level of information processing in such NLMs.[12]

[12]Wei et al.'s (2021) causal interventions on the training data, for instance, could be interesting for such an analysis.

8.7 Conclusion

In this chapter, we investigated whether grammatical number was encoded in BERT's layer and found an encoding of such property, which the model uses to perform number agreement. Using targeted causal interventions, we are able to track how number information is passed across layers, and transmitted from the cue to the target of the agreement relation.

By finding a functional encoding of grammatical number, we bridge the gap between the implementational level and the algorithmic level of the model's linguistic abilities. We hypothesized that on the algorithmic level, the model abstracted away from individual examples by encoding a linear representation of grammatical number. By finding the neural substrate of such representation, we validate that the model has knowledge of such property, and that it uses it at the algorithmic level to solve the number agreement task. This evidence shows the model to generalize beyond the memorization of lexical combinations. This finding illustrates perfectly the interactions between our levels of analysis as causal interventions applied at the implementational level can bring key pieces of information at the algorithmic level.

Additionally, we surprisingly find that the encoding of number is different for nouns and verbs, showing the model to hide subtleties which only surface when performing fine-grained analysis.

This work is encouraging as it represents a first step towards mapping the causal chain of information transfer inside the architecture. Despite the complexity of the network, our targeted analysis uncovered interpretable representations, and showed the model to truly generalize beyond individual examples. Future work could explore more complex properties in layers, and extend the scope of properties beyond the token-level.

While we find the neural substrate of representations plausibly used to carry information in an inference scheme, we do not uncover the entirety of the computations at the algorithmic level. Indeed, being able to assess that grammatical number information flows from the cue to the target in a linear encoding does not answer questions such as: how does my NLM represent the dependency relation, or how does it parse its input sentence to know where to read number information. It is puzzling to see that the model is able to transfer information in sentences sampled randomly, with varying linear distances and number of attractors. Investigating more carefully how attractors influence this causal mechanism is an important step towards getting a comprehensive understanding of how agreement is processed. This leaves space for future work to understand how the model is able to read the right bit of information and transfer it to the target.

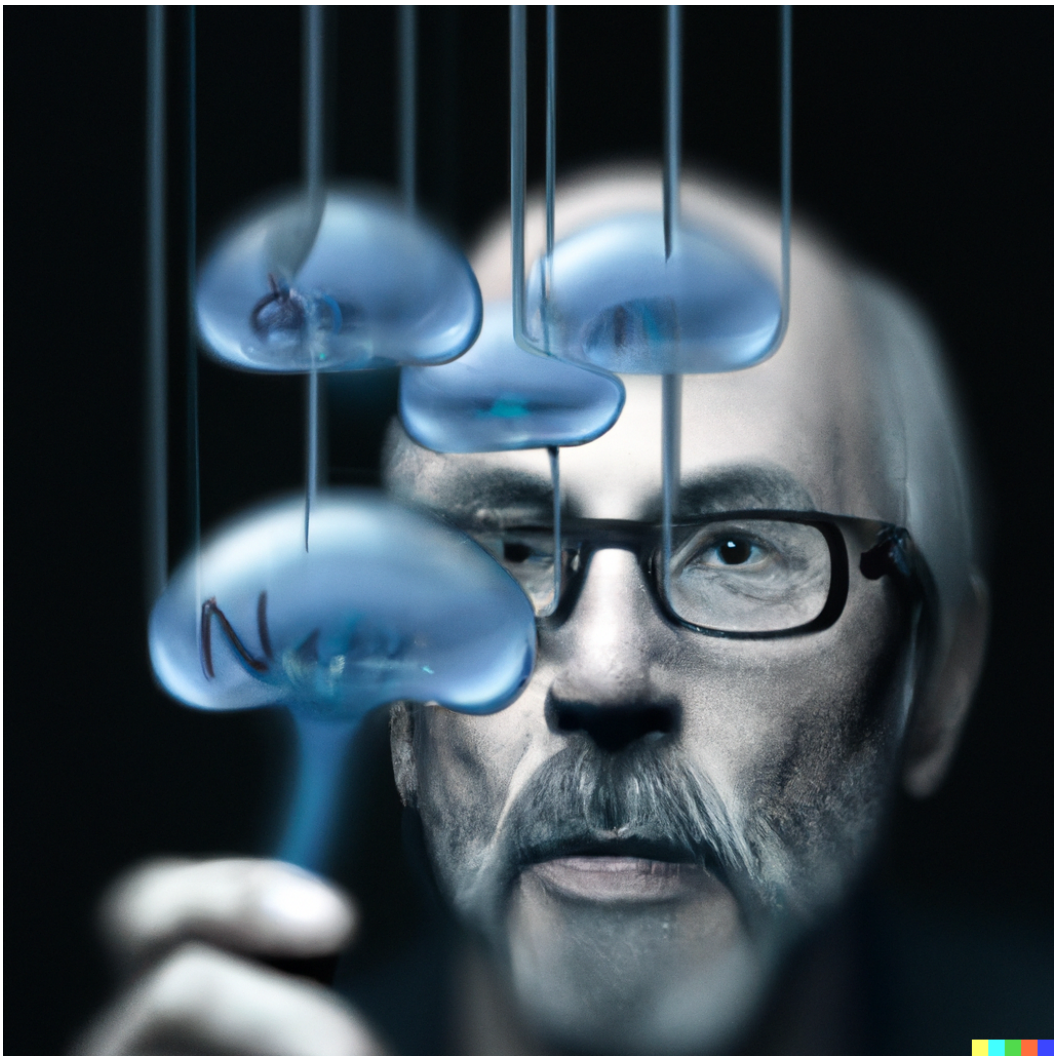


Figure 8.7: "A scientist seeking representations of linguistic abilities in an artificial neural network, digital art", K.L. x DALL·E 2

Conclusion and perspectives

9.1 Conclusions and contributions

In this thesis, we addressed various aspects of linguistic generalization in transformer neural language models. In doing so, we made two main types of contributions: methodological/epistemological contributions guiding future investigations of the questions addressed in this thesis, and empirical contributions shedding light on the linguistic abilities possessed by the models we examined.

On the methodological side, we noted an epistemic gap between different methodological paradigms reviewed in chapter 2, all aimed at assessing the capture of linguistic abilities in transformer-based NLMs. We further formulated a systemic view on the examination of a NLM's linguistic abilities, which can be applied *inter alia* to transformer-based models in chapter 3. In this view, there are three levels of analysis: the surface/behavioral level, the algorithmic level and the implementational level. This articulation reunites different stances on linguistic knowledge which appeared to have separate epistemic grounds. We also describe the relationship which lie between those different levels of analysis. In our behavioral experiments on number agreement, we show that our behavioral evaluation can lead us to formulate hypotheses on the algorithmic level – that processes underlying linguistic generalization on this task could be either semi-lexicalized, or relying on the shallow memorization of lexical patterns. These possibilities remained hypothetical at this stage however, as behavioral evidence alone cannot bear out algorithmic processes with certainty. In a subsequent series of experiments presented in chapter 4 and chapter 5, we built on previously observed limitation of behavioral tests to provide algorithmic accounts of linguistic abilities. Specifically, we showed a model could approximate expected responses on tests requiring sentence structure even without position information. This evidence led us to argue once more that behavioral evidence supporting hypotheses at the algorithmic level could lure us on the computational strategies supporting the surface ability. We chose to address algorithmic questions regarding the usage of information by deploying

causal methods in chapter 6, with a case study targeting the reliance on position information during the model's training. After demonstrating the capacity of such methods to yield solid conclusions, we further built on the power of causal methodologies, and proposed to couple them with targeted behavioral tests to gain better understanding on the strategies supporting linguistic abilities in. In chapter 7 We provided a framework to uncover components of the causal chain leading a model to produce certain behavior. In this framework, we connect the levels of analysis sketched in our systemic view on understanding NLMs as information processing systems, with the goal of understanding a targeted linguistic ability. Equipped with this framework, we proposed to look for the function of different components of the NLM in transmitting information to the decision layer on a linguistic task, and performed a case study experiment aimed at finding functional encodings of grammatical number in chapter 8. We were able to find such encodings, and to track the flow of information in the model. This stresses the importance of behavioral tests targeting fine-grained linguistic abilities, as they pave the way towards gaining understanding on isolated components of the model supporting linguistic knowledge, and how such play a role at the algorithmic level. Finally, the evidence from that experiment allowed us to conclude that the model is not a shallow memorizer on this task, which shows that intervening at the implementational level can yield strong algorithmic evidence, as it is the implementation which supports the algorithms and representations. Such evidence is necessary if we wish to gain a comprehensive understanding of the nature of NLMs' linguistic abilities. Another main contribution from this experiment is that we show the limitations of traditional diagnostic probing techniques in uncovering encodings representing linguistic knowledge that is truly meaningful to the model. Indeed, we find spurious encodings of grammatical number, common to both nouns and verbs, which the model doesn't seem to use on the number agreement task.

We also discussed how linguistic theories and our investigation can contribute to each other in chapter 3, which we demonstrate throughout the thesis. On the one hand, we borrow from phenomena observed by linguists to investigate the abilities of NLMs. On the other hand, our experiment on number agreement in chapter 4 demonstrates clearly the contribution that the investigation of NLMs can bring to linguists. We showed that the NLM under investigation showed error patterns similar to those of humans, questioning the disentanglement between syntactic and semantic processes hypothesized on this syntactic task. This illustrates how our findings can drive new investigations regarding how certain phenomena are processed by humans when a borderline case is identified. Further, our findings in chapter 8 can shed light on the way linguistic abilities can be supported at both the

implementational and algorithmic level, as we showed in our experiments.

Finally, the empirical portion of this thesis informs us on the abilities of the NLMs under investigation. We depart from the surface level of analysis and adopt a top-down approach to delve into the nature of the linguistic abilities possessed by a transformer-based NLM architecture. In the context of a syntactic task, number agreement, we were able to rule out hypotheses from linguistic theory regarding the separation of syntactic and semantic processes in our NLM in chapter 4. We showed that these were not disjoint on this task but intricated, contrarily to evidence from previous work. We hypothesized that the model could have acquired semi-lexicalized abilities on this task, or that it could be a good memorizer of meaningful lexical combinations. The next experiment in chapter 5 questioned our naive view on the model's processing of structure: it is not trivial that expected responses can be approximated using purely distributional information on these tasks, as it seems to be the case in this setting. We chose to investigate more carefully this question by targeting the causal importance of position information during the model's training, showing that transformer models do rely on word order, and foremost position encodings, for their training objective in chapter 6. This evidence is firm, as the experimental design shows a causal effect. Finally, our last series of experiments in chapter 8 allow us to find emergent linear encodings of grammatical number in the model. Those provide evidence that the model does abstract away from semantically-driven lexical combinations, which rules out that it is relying on memorization for the number agreement task.

Throughout this thesis, we bridged the gap which that we identified in the literature, between methodologies which produced accounts for linguistic knowledge that did not translate into each other. In doing so, we draw inspiration from the neuroscientific litterature and cognitive science to advance our understanding of neural language models. The systemic view that we provide on examining a NLM's linguistic knowledge might help each of these approaches benefit to the others, as our functionalist view on a NLM's components leads to connecting any investigation at the neuron or layer level to the higher algorithmic function it plays in capturing linguistic abilities. We contend that the quest will not be easy as understanding the deeper implementational level supporting linguistic knowledge and uncovering the entirety of the causal chain supporting the decisions of NLMs requires careful analysis of the architecture's components. The operations underlying the processing of linguistic structure are yet to discover, which might require making alternative hypotheses to models of sentence structure found in grammars as NLMs don't have any hierarchical bias. Equipped with the formalizations provided in this thesis, these goals can be pursued on solid grounds: with our functionalist approach, rep-

resentations and operations found in layers should play a role in making predictions and should be sought using causal methods.

While the empirical portion of this thesis addresses transformer-based neural langugae models, many of the contributions made in this thesis are beyond this specific type of architecture. In particular, the formalization of the core-questions is made as general as possible and to be applied to future state-of-the-art models. We were also cautious in making these formulations apply to both autoregressive and bidirectional (or masked) language models.

9.2 Perspectives and future work

In this thesis, we laid foundations for the investigation of linguistic knowledge in NLMs in which we connect the various levels of analysis of such question. We have been able to find a neural substrate at the implementational level, encoding an abstract representation of grammatical number at an algorithmic level, which is itself used to produce adequate predictions at the surface behavior level. In doing so however, we only uncovered a portion of the model's capacity to capture linguistic abstractions. We are still a long way from understanding how the model processes sentence structure, as we do not know how the model determines where it should read that bit of information. In this thesis, we sketched ways to address such questions. To understand how the model represents sentence structure, we could first understand which categories it represents in its layers, and how it integrates constraints that such categories exert on each other.

Uncovering the mechanisms by which a NLM applies the simplest of its heuristics in their entirety could be a starting point before addressing hard phenomena. To do so, we need to formulate mechanistic hypotheses on how information is processed in the network, not only what information is functionally encoded. The task might be tedious due to the variety of components which need to be analyzed. Another main problems with current NLMs is their strong reliance on non-linear functions, which make the functions they learn, and the processes through which they achieve generalisation, largely opaque. As a result, intermediate layers and transforms that most NLMs are built from are intrinsically hard to interpret. This causes much of their processing to be inaccessible to direct observation. Another fruitful area to explore in future work is searching for proficient NLMs which are additionally more easy to interpret.[13] One might argue that there is no point in over-

[13]Recently, [Lappin, 2022] recently proposed Unitary Recurrent Networks, as an architecture which relies only on orthogonal matrices for word embedding, and linear algebraic functions as intermediate transforms.

analyzing a specific architecture as deep knowledge of how it functions will not necessarily be applicable to the next state-of-the-art system. It is certainly a strong argument to keep in mind. On the other hand, if we are able to set up analysis methods that prove to grasp the complexities for a given architecture, we get closer to having a procedure to diagnose any architecture. In practice, many choices are obviously architecture-specific and not universally transferable. However, in this thesis, we tried to state the different problems by referring as little as possible to a specific neural model. We can surmise that causal analyses like ours are very hard to produce in the general case if we wish to target more complex properties, making it harder to find its encoding, especially if its scope is beyond the word-level. The problems raised in this thesis leave several areas open for exploration, such as comprehensive accounts for how NLMs could implement memorization, or a comprehensive understanding of the mechanisms which support the combination of information from tokens into an encoding of properties for greater parts, such as those presented in our typology of linguistic units, in the model's layers.

List of publications

Peer-reviewed articles (first author)

Karim Lasri, Tiago Pimentel, Thierry Poibeau, Alessandro Lenci, Ryan Cotterell. "*Probing for the Usage of Grammatical Number* ". Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (**oral presentation**).
DOI: 10.18653/v1/2022.acl-long.603

Karim Lasri, Thierry Poibeau, Alessandro Lenci. "*Does BERT really agree ? Fine-grained Analysis of Lexical Dependence on a Syntactic Task*". Findings of the Association for Computational Linguistics: ACL 2022 (**poster**).
DOI: 10.18653/v1/2022.findings-acl.181

Karim Lasri, Olga Seminck, Thierry Poibeau, Alessandro Lenci. "*Subject Verb Agreement Error Patterns in Meaningless Sentences: Humans vs. BERT*". Findings of the Association for Computational Linguistics: ACL 2022 (**poster**).

Karim Lasri, Thierry Poibeau, Alessandro Lenci. "*Word Order Matters when you Increase Masking*". Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (**oral presentation**).

Bibliography

[Abdou, 2022] Mostafa Abdou, Vinit Ravishankar, Artur Kulmizev, and Anders Søgaard. "Word Order Does Matter and Shuffled Language Models Know It". *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 6907–6919. DOI: 10.18653/v1/2022.acl-long.476 (cit. on pp. 106, 111).

[Adger, 2003] David Adger. *Core Syntax: A Minimalist Approach*. Oxford University Press, 2003 (cit. on p. 44).

[Adi, 2017] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. "Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks". *International Conference on Learning Representations*. 2017. eprint: 1608.04207 (cit. on pp. 2, 30, 31).

[Alain, 2016] Guillaume Alain and Yoshua Bengio. "Understanding intermediate layers using linear classifier probes". *arXiv preprint arXiv:1610.01644* (2016) (cit. on pp. 2, 32).

[Ambridge, 2016] Ben Ambridge, Amy Bidgood, Julian Pine, Caroline Rowland, and daniel freudenthal daniel. "Is Passive Syntax Semantically Constrained? Evidence From Adult Grammaticality Judgment and Comprehension Studies". *Cognitive Science* 40 (2016), pp. 1435–1459. DOI: 10.1111/cogs.12277 (cit. on p. 51).

[Anonymous, 2023] Anonymous. "Quantifying Memorization Across Neural Language Models". *Submitted to The Eleventh International Conference on Learning Representations*. 2023 (cit. on p. 5).

[Armendariz, 2020] Carlos Santos Armendariz, Matthew Purver, Matej Ulčar, Senja Pollak, Nikola Ljubešić, and Mark Granroth-Wilding. "CoSimLex: A Resource for Evaluating Graded Word Similarity in Context". English. *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, 2020, pp. 5878–5886 (cit. on p. 26).

[Arrivé, 1969] Michel Arrivé. "Les Éléments de syntaxe structurale, de L. Tesnière". *Langue française* 1.1 (1969), pp. 36–40. DOI: 10.3406/lfr.1969.5395 (cit. on p. 29).

[Augenstein, 2017] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. "SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications". *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 546–555. DOI: 10.18653/v1/S17-2091 (cit. on p. 26).

[Ba, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016 (cit. on p. 22).

[Bahdanau, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". *ArXiv* 1409 (2014) (cit. on p. 21).

[Baltin, 1982] Mark R. Baltin. "A Landing Site Theory of Movement Rules". *Linguistic Inquiry* 13.1 (1982), pp. 1–38 (cit. on p. 56).

[Bandy, 2021] Jack Bandy and Nicholas Vincent. "Addressing "Documentation Debt" in Machine Learning Research: A Retrospective Datasheet for BookCorpus". *CoRR* abs/2105.05241 (2021). arXiv: 2105.05241 (cit. on p. 93).

[Bar-Hillel, 1953] Yehoshua Bar-Hillel. "A Quasi-Arithmetical Notation for Syntactic Description". *Language* 29 (1953), p. 47 (cit. on p. 60).

[Baroni, 2019] Marco Baroni. "Linguistic generalization and compositionality in modern artificial neural networks". *Philosophical Transactions of the Royal Society B* 375(1791) (2019). arXiv: 1904.00157 (cit. on p. 75).

[Baroni, 2020] Marco Baroni. "Linguistic generalization and compositionality in modern artificial neural networks". *Philosophical Transactions of the Royal Society B: Biological Sciences* 375 (2020), p. 20190307. DOI: 10.1098/rstb.2019.0307 (cit. on p. 48).

[Baroni, 2021] Marco Baroni. "On the proper role of linguistically-oriented deep net analysis in linguistic theorizing". *ArXiv* abs/2106.08694 (2021) (cit. on p. 45).

[Belinkov, 2017a] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. "What do Neural Machine Translation Models Learn about Morphology?" *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 861–872. DOI: 10.18653/v1/P17-1080 (cit. on pp. 31, 32).

[Belinkov, 2017b] Yonatan Belinkov, Lluis Marquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. "Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks". *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, 2017, pp. 1–10 (cit. on p. 31).

[Bernardy, 2017] Jean-Phillipe Bernardy and Shalom Lappin. “Using Deep Neural Networks to Learn Syntactic Agreement”. *Linguistic Issues in Language Technology, Volume 15, 2017*. CSLI Publications, 2017 (cit. on pp. 69, 70, 85).

[Berwick, 2015] Robert C. Berwick and Noam Chomsky. *Why Only Us: Language and Evolution*. The MIT Press, 2015 (cit. on p. 44).

[Bhattamishra, 2020] Satwik Bhattamishra, Arkil Patel, and Navin Goyal. “On the Computational Power of Transformers and Its Implications in Sequence Modeling”. 2020, pp. 455–475. DOI: 10.18653/v1/2020.conll-1.37 (cit. on p. 23).

[Bird, 2009] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O’Reilly Media, Inc.", 2009 (cit. on p. 79).

[Blank, 2015] Idan Blank, Zuzanna Balewski, Kyle Mahowald, and Evelina Fedorenko. “Syntactic processing is distributed across the language system”. *NeuroImage* 127 (2015). DOI: 10.1016/j.neuroimage.2015.11.069 (cit. on p. 130).

[Blevins, 2018] Terra Blevins, Omer Levy, and Luke Zettlemoyer. “Deep RNNs Encode Soft Hierarchical Syntax”. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 14–19. DOI: 10.18653/v1/P18-2003 (cit. on p. 31).

[Bock, 1992] Kathryn Bock and J.Cooper Cutting. “Regulating mental energy: Performance units in language production”. *Journal of Memory and Language* 31.1 (1992), pp. 99–127. DOI: https://doi.org/10.1016/0749-596X(92)90007-K (cit. on p. 82).

[Bock, 1991] Kathryn Bock and Carol A Miller. “Broken agreement”. *Cognitive psychology* 23.1 (1991), pp. 45–93 (cit. on p. 78).

[Boeckx, 2006] Cedric Boeckx. *Linguistic Minimalism: Origins, Concepts, Methods, and Aims*. Oxford University Press UK, 2006 (cit. on p. 44).

[Bresnan, 1973] Joan W. Bresnan. “Syntax of the Comparative Clause Construction in English”. *Linguistic Inquiry* 4.3 (1973), pp. 275–343 (cit. on p. 56).

[Carlini, 2020] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, et al. “Extracting Training Data from Large Language Models”. *USENIX Security Symposium*. 2020 (cit. on p. 5).

[Cavnar, 1994] William B Cavnar, John M Trenkle, et al. “N-gram-based text categorization”. *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*. Vol. 161175. Citeseer. 1994 (cit. on p. 17).

[Chang, 2021] Tyler Chang, Yifan Xu, Weijian Xu, and Zhuowen Tu. "Convolutions and Self-Attention: Re-interpreting Relative Positions in Pre-trained Language Models". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, 2021, pp. 4322–4333. DOI: 10.18653/v1/2021.acl-long.333 (cit. on pp. 19, 111).

[Chen, 2021a] Pu-Chin Chen, Henry Tsai, Srinadh Bhojanapalli, Hyung Won Chung, Yin-Wen Chang, and Chun-Sung Ferng. "A Simple and Effective Positional Encoding for Transformers". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 2974–2988. DOI: 10.18653/v1/2021.emnlp-main.236 (cit. on pp. 19, 111).

[Chen, 2021b] Peng Chen. "PermuteFormer: Efficient Relative Position Encoding for Long Sequences". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 10606–10618. DOI: 10.18653/v1/2021.emnlp-main.828 (cit. on pp. 19, 111).

[Chen, 2017] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. "Enhanced LSTM for Natural Language Inference". *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1657–1668. DOI: 10.18653/v1/P17-1152 (cit. on p. 16).

[Chomsky, 1956a] N. Chomsky. "Three models for the description of language". *IRE Transactions on Information Theory* 2.3 (1956), pp. 113–124. DOI: 10.1109/TIT.1956.1056813 (cit. on p. 60).

[Chomsky, 1967] N. Chomsky. "A Review of B. F. Skinner's Verbal Behavior". *Readings in the Psychology of Language*. Ed. by Leon A. Jakobovits and Murray S. Miron. Englewood Cliffs, N.J.: Prentice-Hall, 1967 (cit. on p. 43).

[Chomsky, 1963] N. Chomsky and M.P. Schützenberger. "The Algebraic Theory of Context-Free Languages*". *Computer Programming and Formal Systems*. Ed. by P. Braffort and D. Hirschberg. Vol. 35. Studies in Logic and the Foundations of Mathematics. Elsevier, 1963, pp. 118–161. DOI: https://doi.org/10.1016/S0049-237X(08)72023-8 (cit. on p. 60).

[Chomsky, 1956b] Noam Chomsky. "Three models for the description of language". *IRE Transactions on information theory* 2.3 (1956), pp. 113–124 (cit. on p. 71).

[Chomsky, 1957] Noam Chomsky. *Syntactic Structures*. The Hague: Mouton and Co., 1957 (cit. on p. 43).

[Chomsky, 1959] Noam Chomsky. "On certain formal properties of grammars". *Information and Control* 2.2 (1959), pp. 137–167. DOI: https://doi.org/10.1016/S0019-9958(59)90362-6 (cit. on p. 60).

[Chomsky, 1965a] Noam Chomsky. *Aspects of the Theory of Syntax*. 50th ed. The MIT Press, 1965 (cit. on p. 29).

[Chomsky, 1965b] Noam Chomsky. *Aspects of the Theory of Syntax*. Cambridge: The MIT Press, 1965 (cit. on p. 52).

[Chomsky, 1971] Noam Chomsky. *Problems of knowledge and freedom: The Russell lectures*. Pantheon Books, 1971 (cit. on p. 71).

[Chomsky, 1976] Noam Chomsky et al. *Reflections on language*. Temple Smith London, 1976 (cit. on p. 71).

[Chomsky, 1980] Noam Chomsky. "Rules and representations". *Behavioral and Brain Sciences* 3.1 (1980), pp. 1–15. DOI: 10.1017/S0140525X00001515 (cit. on p. 43).

[Chomsky, 1986] Noam Chomsky. *Knowledge of Language. Its Nature, Origin, and Use*. Convergence. New York/Westport/London: Praeger, 1986 (cit. on pp. 44, 52).

[Chomsky, 1995] Noam Chomsky. *The Minimalist Program*. 20th ed. 1995 (cit. on p. 44).

[Chow, 2015] Wing Yee Chow, Cybelle Smith, Ellen Lau, and Colin Phillips. "A "bag-of-arguments" mechanism for initial verb predictions". *Language, Cognition and Neuroscience* 31 (2015), pp. 1–20. DOI: 10.1080/23273798.2015.1066832 (cit. on p. 56).

[Chung, 2014] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling". English (US). *NIPS 2014 Workshop on Deep Learning, December 2014*. 2014 (cit. on pp. 16, 23).

[Chung, 1995] Sandra Chung, William Ladusaw, and James Mccloskey. "Sluicing and Logical Form". *Natural Language Semantics* 3 (1995), pp. 239–282. DOI: 10.1007/BF01248819 (cit. on p. 56).

[Clark, 2011] Alexander Clark and Shalom Lappin. *Linguistic Nativism and the Poverty of the Stimulus*. 2011. DOI: 10.1002/9781444390568 (cit. on p. 45).

[Clark, 2019] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. "What Does BERT Look at? An Analysis of BERT's Attention". *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 276–286. DOI: 10.18653/v1/W19-4828 (cit. on p. 129).

[Clark, 2007] Stephen Clark and James R. Curran. "Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models". *Computational Linguistics* 33.4 (2007), pp. 493–552. DOI: 10.1162/coli.2007.33.4.493 (cit. on p. 44).

[Collins, 2005] Chris Collins. "A Smuggling Approach to the Passive in English". *Syntax* 8 (2005), pp. 81–120. DOI: 10.1111/j.1467-9612.2005.00076.x (cit. on p. 56).

[Commission, 2018] European Commission. *2018 reform of EU data protection rules*. 2018. URL: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf (visited on 05/16/2022) (cit. on p. 81).

[Conneau, 2018a] Alexis Conneau and Douwe Kiela. "SentEval: An Evaluation Toolkit for Universal Sentence Representations". *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), 2018 (cit. on pp. 18, 24).

[Conneau, 2018b] Alexis Conneau, German Kruszewski, Guillaume Lample, Loic Barrault, and Marco Baroni. "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties". *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 2126–2136. DOI: 10.18653/v1/P18-1198 (cit. on pp. 30–32).

[Corbett, 2003] G. Corbett. "Agreement: Terms and Boundaries". *The Role of Agreement in Natural Language: TLS 5 Proceedings*. 2003, pp. 109–122 (cit. on p. 68).

[Culicover, 1999] Peter W. Culicover and Ray Jackendoff. "The View from the Periphery: The English Comparative Correlative". *Linguistic Inquiry* 30.4 (1999), pp. 543–571 (cit. on p. 56).

[Dai, 2019] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context". *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 2978–2988. DOI: 10.18653/v1/P19-1285 (cit. on pp. 21, 110).

[Dauphin, 2016] Yann Dauphin, Angela Fan, Michael Auli, and David Grangier. "Language Modeling with Gated Convolutional Networks" (2016) (cit. on p. 17).

[Dayal, 1998] Veneeta Dayal. "Any as Inherently Modal". *Linguistics and Philosophy* 21 (1998), pp. 433–476 (cit. on p. 56).

[Delétang, 2022] Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Wenliang, Elliot Catt, et al. *Neural Networks and the Chomsky Hierarchy*. 2022. DOI: 10.48550/arXiv.2207.02098 (cit. on p. 24).

[Devlin, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805 (cit. on pp. 17, 94).

[Devlin, 2019a] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *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)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423 (cit. on pp. 1, 3, 19, 35, 55, 149).

[Devlin, 2019b] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". *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)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423 (cit. on pp. 21, 25, 109, 114).

[Dey, 2017] Rahul Dey and Fathi M. Salem. "Gate-variants of Gated Recurrent Unit (GRU) neural networks". *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)* (2017), pp. 1597–1600 (cit. on pp. 16, 23).

[Doddington, 2002] George Doddington. "Automatic Evaluation of Machine Translation Quality Using N-Gram Co-Occurrence Statistics". *Proceedings of the Second International Conference on Human Language Technology Research*. HLT '02. San Diego, California: Morgan Kaufmann Publishers Inc., 2002, pp. 138–145 (cit. on p. 17).

[Dong, 2014] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. "Learning a Deep Convolutional Network for Image Super-Resolution". *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 184–199 (cit. on p. 17).

[Dufter, 2021] Philipp Dufter, Martin Schmitt, and Hinrich Schütze. "Position Information in Transformers: An Overview". *CoRR* abs/2102.11090 (2021). arXiv: 2102.11090 (cit. on pp. 108, 114).

[Elazar, 2022] Yanai Elazar, Victoria Basmov*, Yoav Goldberg, and Reut Tsarfaty. "Text-based NP Enrichment". *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 764–784. DOI: 10.1162/tacl_a_00488. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00488/2037151/tacl_a_00488.pdf (cit. on p. 26).

[Elazar, 2021] Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. "Amnesic probing: Behavioral explanation with amnesic counterfactuals". *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 160–175 (cit. on pp. 2, 33, 134, 145, 147, 154).

[Erk, 2013] Katrin Erk, Diana McCarthy, and Nicholas Gaylord. "Measuring Word Meaning in Context". *Computational Linguistics* 39.3 (2013), pp. 511–554. DOI: 10.1162/COLI_a_00142. eprint: https://direct.mit.edu/coli/article-pdf/39/3/511/1801959/coli_a_00142.pdf (cit. on p. 26).

[Ettinger, 2020] Allyson Ettinger. "What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models". *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 34–48. DOI: 10.1162/tacl_a_00298 (cit. on pp. 33, 36, 56, 81).

[Ettinger, 2018] Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. "Assessing Composition in Sentence Vector Representations". *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 1790–1801 (cit. on p. 32).

[Ettinger, 2016] Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. "Probing for semantic evidence of composition by means of simple classification tasks". *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 134–139. DOI: 10.18653/v1/W16-2524 (cit. on pp. 31, 32).

[Fanselow, 2006] Gisbert Fanselow, Caroline Féry, Matthias Schlesewsky, and Ralf Vogel. *Gradience in Grammar: Generative Perspectives*. Oxford University Press, 2006. DOI: 10.1093/acprof:oso/9780199274796.001.0001 (cit. on p. 51).

[Federmeier, 1999] Kara D. Federmeier and Marta Kutas. "A Rose by Any Other Name: Long-Term Memory Structure and Sentence Processing". *Journal of Memory and Language* 41.4 (1999), pp. 469–495. DOI: https://doi.org/10.1006/jmla.1999.2660 (cit. on p. 56).

[Ferreira, 2005] Fernanda Ferreira. "Psycholinguistics, Formal Grammars, and Cognitive Science". *Linguistic Review - LINGUIST REV* 22 (2005), pp. 365–380. DOI: 10.1515/tlir.2005.22.2-4.365 (cit. on p. 50).

[Finlayson, 2021] Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. "Causal Analysis of Syntactic Agreement Mechanisms in Neural Language Models". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, 2021, pp. 1828–1843. DOI: 10.18653/v1/2021.acl-long.144 (cit. on pp. 76, 144).

[Fischler, 1983] Ira Fischler, Paul Alexander Bloom, Donald G. Childers, Salim Roukos, and Nathan W. Perry. "Brain potentials related to stages of sentence verification." *Psychophysiology* 20 4 (1983), pp. 400–9 (cit. on p. 56).

[Gazdar, 1981] Gerald Gazdar. “Unbounded Dependencies and Coordinate Structure”. *Linguistic Inquiry* 12.2 (1981), pp. 155–184 (cit. on p. 56).

[Geva, 2020] Mor Geva, Ankit Gupta, and Jonathan Berant. “Injecting Numerical Reasoning Skills into Language Models”. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 946–958. DOI: 10.18653/v1/2020.acl-main.89 (cit. on p. 144).

[Geva, 2021] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. “Transformer Feed-Forward Layers Are Key-Value Memories”. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 5484–5495. DOI: 10.18653/v1/2021.emnlp-main.446 (cit. on p. 129).

[Gibson, 2013] Edward Gibson and Evelina Fedorenko. “The need for quantitative methods in syntax and semantics research”. *Language and Cognitive Processes* 28.1-2 (2013), pp. 88–124. DOI: 10.1080/01690965.2010.515080. eprint: https://doi.org/10.1080/01690965.2010.515080 (cit. on p. 50).

[Giulianelli, 2018] Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. “Under the Hood: Using Diagnostic Classifiers to Investigate and Improve how Language Models Track Agreement Information”. *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 240–248. DOI: 10.18653/v1/W18-5426 (cit. on pp. 30, 144).

[Godfrey, 1992] John J. Godfrey, Edward C. Holliman, and Jane McDaniel. “SWITCHBOARD: Telephone Speech Corpus for Research and Development”. *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*. ICASSP’92. San Francisco, California: IEEE Computer Society, 1992, pp. 517–520 (cit. on p. 26).

[Goldberg, 2004] Adele Goldberg and Ray Jackendoff. “The English Resultative as a Family of Constructions”. *Language* 80 (2004). DOI: 10.1353/lan.2004.0129 (cit. on p. 56).

[Goldberg, 2019] Yoav Goldberg. “Assessing BERT’s Syntactic Abilities”. *CoRR* abs/1901.05287 (2019). arXiv: 1901.05287 (cit. on pp. 33, 69, 72, 76, 149).

[Gulordava, 2018] Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. “Colorless Green Recurrent Networks Dream Hierarchically”. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 1195–1205. DOI: 10.18653/v1/N18-1108 (cit. on pp. 70, 72, 76).

[Hall Maudslay, 2020] Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. "A Tale of a Probe and a Parser". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020 (cit. on p. 30).

[Hartsuiker, 2001] Robert J Hartsuiker, Inés Antón-Méndez, and Marije Van Zee. "Object attraction in subject-verb agreement construction". *Journal of Memory and Language* 45.4 (2001), pp. 546–572 (cit. on p. 78).

[Haviv, 2022] Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. *Transformer Language Models without Positional Encodings Still Learn Positional Information*. 2022. DOI: 10.48550/ARXIV.2203.16634 (cit. on pp. 106, 107, 111, 118).

[Hayes, 2000] Bruce Hayes. "Gradient Well-Formedness in Optimality Theory" (2000) (cit. on p. 52).

[He, 2020a] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. "DeBERTa: Decoding-enhanced BERT with Disentangled Attention". *CoRR* abs/2006.03654 (2020). arXiv: 2006.03654 (cit. on pp. 19, 111).

[He, 2020b] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. "DeBERTa: Decoding-enhanced BERT with Disentangled Attention". *CoRR* abs/2006.03654 (2020) (cit. on pp. 21, 55, 110).

[Hendrycks, 2016] Dan Hendrycks and Kevin Gimpel. "Gaussian Error Linear Units (GELUs)". 2016. arXiv: http://arxiv.org/abs/1606.08415v3 [cs.LG] (cit. on p. 23).

[Hessel, 2021] Jack Hessel and Alexandra Schofield. "How effective is BERT without word ordering? Implications for language understanding and data privacy". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online: Association for Computational Linguistics, 2021, pp. 204–211. DOI: 10.18653/v1/2021.acl-short.27 (cit. on p. 106).

[Hewitt, 2021] John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. "Conditional probing: measuring usable information beyond a baseline". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 1626–1639. DOI: 10.18653/v1/2021.emnlp-main.122 (cit. on p. 145).

[Hewitt, 2019a] John Hewitt and Percy Liang. "Designing and Interpreting Probes with Control Tasks". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 2733–2743. DOI: 10.18653/v1/D19-1275 (cit. on pp. 32, 36, 135).

[Hewitt, 2019b] John Hewitt and Christopher D. Manning. “A Structural Probe for Finding Syntax in Word Representations”. *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)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4129–4138. DOI: 10.18653/v1/N19-1419 (cit. on pp. 29, 30).

[Hjelmslev, 1957] Louis Hjelmslev. *Prolégomènes a une théorie du langage / Trad. U. Canger. La Structure fondamentale du langage / Trad. A. M. Leonard*. Arguments (Éditions de Minuit) ; 35. Paris: Edit. de Minuit, 1957 (cit. on p. 42).

[Hochreiter, 1997] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-term Memory”. *Neural computation* 9 (1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735 (cit. on pp. 16, 23).

[Hockenmaier, 2003] Julia Hockenmaier. “Data and models for statistical parsing with Combinatory Categorial Grammar” (2003) (cit. on p. 44).

[Hockenmaier, 2002] Julia Hockenmaier and Mark Steedman. “Generative Models for Statistical Parsing with Combinatory Categorial Grammar”. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002, pp. 335–342. DOI: 10.3115/1073083.1073139 (cit. on p. 44).

[Hoffman, 2013] Thomas Hoffman and Graeme Trousdale, eds. *The Oxford Handbook of Construction Grammar*. Oxford: Oxford University Press, 2013 (cit. on p. 76).

[Hu, 2020] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. “XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalisation”. *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 4411–4421 (cit. on p. 26).

[Huang, 2020] Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. “Improve Transformer Models with Better Relative Position Embeddings”. *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, 2020, pp. 3327–3335. DOI: 10.18653/v1/2020.findings-emnlp.298 (cit. on p. 109).

[Hulth, 2003] Anette Hulth. “Improved Automatic Keyword Extraction Given More Linguistic Knowledge”. *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. 2003, pp. 216–223 (cit. on p. 26).

[Jackendoff, 1971] Ray S. Jackendoff. “Gapping and Related Rules”. *Linguistic Inquiry* 2.1 (1971), pp. 21–35 (cit. on p. 56).

[Kadmon, 1993] Nirit Kadmon and Fred Landman. “Any”. *Linguistics and Philosophy* 16.4 (1993), pp. 353–422. DOI: 10.1007/BF00985272 (cit. on p. 56).

[Kalchbrenner, 2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. "A Convolutional Neural Network for Modelling Sentences". *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 655–665. DOI: 10.3115/v1/P14-1062 (cit. on p. 17).

[Kaplan, 2004] Ronald M. Kaplan. "Lexical Functional Grammar A Formal System for Grammatical Representation". 2004 (cit. on p. 29).

[Katharopoulos, 2020] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. "Transformers Are RNNs: Fast Autoregressive Transformers with Linear Attention". *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org, 2020 (cit. on p. 23).

[Katz, 1963] Jerrold J. Katz and Jerry A. Fodor. "The structure of a semantic theory". *The structure of a semantic theory* 39.2 (1963), pp. 170–210 (cit. on p. 91).

[Khandelwal, 2020] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. "Generalization through Memorization: Nearest Neighbor Language Models". *International Conference on Learning Representations*. 2020 (cit. on p. 5).

[Kim, 2010] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. "SemEval-2010 Task 5 : Automatic Keyphrase Extraction from Scientific Articles". *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, 2010, pp. 21–26 (cit. on p. 26).

[Kiros, 2014] Ryan Kiros, Ruslan Salakhutdinov, and Richard Zemel. "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models". *31st International Conference on Machine Learning, ICML 2014* 3 (2014) (cit. on p. 17).

[Klafka, 2020] Josef Klafka and Allyson Ettinger. "Spying on Your Neighbors: Fine-grained Probing of Contextual Embeddings for Information about Surrounding Words". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 4801–4811. DOI: 10.18653/v1/2020.acl-main.434 (cit. on pp. 31, 144, 156, 160).

[Korsky, 2019] Samuel Korsky and Robert Berwick. *On the Computational Power of RNNs*. 2019 (cit. on p. 24).

[Kovaleva, 2019] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. "Revealing the Dark Secrets of BERT". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 4365–4374. DOI: 10.18653/v1/D19-1445 (cit. on pp. 34, 129, 157).

[Kracht, 2006] Marcus Kracht and Geoffrey Pullum. “The Mathematics of Language”. *The Mathematical Intelligencer* 28 (2006), pp. 74–78. DOI: 10.1007/BF02987162 (cit. on p. 60).

[Krizhevsky, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012 (cit. on p. 17).

[Kuncoro, 2018a] Adhi Kuncoro. *The Perils Of Natural Behaviour Tests For Unnatural Models: The Case Of Number Agreement*. 2018 (cit. on p. 101).

[Kuncoro, 2018b] Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. “LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better”. 2018, pp. 1426–1436. DOI: 10.18653/v1/P18-1132 (cit. on p. 45).

[Lai, 2014] Alice Lai and Julia Hockenmaier. “Illinois-LH: A Denotational and Distributional Approach to Semantics”. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, 2014, pp. 329–334. DOI: 10.3115/v1/S14-2055 (cit. on p. 25).

[Lake, 2018] Brenden Lake and Marco Baroni. “Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks”. *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 2873–2882 (cit. on p. 18).

[Lakew, 2018] Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. “A Comparison of Transformer and Recurrent Neural Networks on Multilingual Neural Machine Translation”. *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 641–652 (cit. on p. 24).

[Lakretz, 2019] Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. “The emergence of number and syntax units in LSTM language models”. *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)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 11–20. DOI: 10.18653/v1/N19-1002 (cit. on pp. 33, 129, 144).

[Lambek, 1958] Joachim Lambek. “The Mathematics of Sentence Structure”. *The American Mathematical Monthly* 65.3 (1958), pp. 154–170. DOI: 10.1080/00029890.1958.11989160. eprint: https://doi.org/10.1080/00029890.1958.11989160 (cit. on p. 60).

[Lappin, 2021] Shalom Lappin. *Deep Learning and Linguistic Representation*. 2021. DOI: 10.1201/9781003127086 (cit. on pp. 17, 45).

[Lappin, 2022] Shalom Lappin and Jean-Philippe Bernardy. "A Neural Model for Compositional Word Embeddings and Sentence Processing". *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 12–22. DOI: 10.18653/v1/2022.cmcl-1.2 (cit. on p. 166).

[Lasri, 2022] Karim Lasri, Alessandro Lenci, and Thierry Poibeau. "Does BERT really agree? Fine-grained Analysis of Lexical Dependence on a Syntactic Task". *arXiv preprint arXiv:2204.06889* (2022) (cit. on pp. 79, 84, 85).

[Lau, 2016] Jey Lau, Alexander Clark, and Shalom Lappin. "Grammaticality, Acceptability, and Probability: A Probabilistic View of Linguistic Knowledge". *Cognitive science* 41 (2016). DOI: 10.1111/cogs.12414 (cit. on pp. 13, 50–52, 54).

[Lau, 2020] Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. "How Furiously Can Colorless Green Ideas Sleep? Sentence Acceptability in Context". *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 296–310. DOI: 10.1162/tacl_a_00315 (cit. on pp. 17, 84).

[Laurinavichyute, 2022] Anna Laurinavichyute and Titus von der Malsburg. "Semantic Attraction in Sentence Comprehension". *Cognitive Science* 46.2 (2022), e13086. DOI: https://doi.org/10.1111/cogs.13086. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.13086 (cit. on p. 76).

[Lazaridou, 2015] Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. "Combining Language and Vision with a Multimodal Skip-gram Model". *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, 2015, pp. 153–163. DOI: 10.3115/v1/N15-1016 (cit. on p. 17).

[Lecun, 1995] Yann Lecun and Y. Bengio. "Convolutional Networks for Images, Speech, and Time-Series". 1995 (cit. on p. 17).

[Leong, 2020] Chee Wee (Ben) Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. "A Report on the 2020 VUA and TOEFL Metaphor Detection Shared Task". *Proceedings of the Second Workshop on Figurative Language Processing*. Online: Association for Computational Linguistics, 2020, pp. 18–29. DOI: 10.18653/v1/2020.figlang-1.3 (cit. on p. 26).

[Linzen, 2019] Tal Linzen. "What can linguistics and deep learning contribute to each other? Response to Pater". *Language* 95 (2019), e108–e99 (cit. on p. 44).

[Linzen, 2021] Tal Linzen and Marco Baroni. "Syntactic Structure from Deep Learning". *Annual Review of Linguistics* 7.1 (2021), pp. 195–212. DOI: 10.1146/annurev-linguistics-032020-051035. eprint: https://doi.org/10.1146/annurev-linguistics-032020-051035 (cit. on p. 45).

[Linzen, 2016] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. "Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies". *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 521–535. DOI: 10.1162/tacl_a_00115 (cit. on pp. 33, 54, 56, 68, 69, 144, 149).

[Linzen, 2018] Tal Linzen and Brian Leonard. "Distinct patterns of syntactic agreement errors in recurrent networks and humans". *CoRR* abs/1807.06882 (2018). arXiv: 1807.06882 (cit. on p. 77).

[Liu, 2019a] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. "Linguistic Knowledge and Transferability of Contextual Representations". *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)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 1073–1094. DOI: 10.18653/v1/N19-1112 (cit. on p. 32).

[Liu, 2019b] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692 (cit. on p. 1).

[Liu, 2019c] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692 (cit. on pp. 17, 21, 55, 118).

[Liu, 2019d] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, et al. "RoBERTa: A robustly optimized BERT pretraining approach". *arXiv preprint arXiv:1907.11692* (2019) (cit. on p. 109).

[Loula, 2018] João Loula, Marco Baroni, and Brenden Lake. "Rearranging the Familiar: Testing Compositional Generalization in Recurrent Networks". *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 108–114. DOI: 10.18653/v1/W18-5413 (cit. on p. 18).

[Lovering, 2021] Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. "Predicting Inductive Biases of Pre-Trained Models". *International Conference on Learning Representations*. 2021 (cit. on pp. 33, 90).

[Marelli, 2014] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. "A SICK cure for the evaluation of compositional distributional semantic models". *Proceedings of the Ninth International Confer-*

ence on Language Resources and Evaluation (LREC'14). Reykjavik, Iceland: European Language Resources Association (ELRA), 2014, pp. 216–223 (cit. on pp. 25, 26).

[Mariño, 2006] José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, et al. "N-gram-based Machine Translation". *Computational Linguistics* 32.4 (2006), pp. 527–549. DOI: 10.1162/coli.2006.32.4.527. eprint: https://direct.mit.edu/coli/article-pdf/32/4/527/1798355/coli.2006.32.4.527.pdf (cit. on p. 17).

[Marr, 1982] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. The MIT Press, 1982. DOI: 10.7551/mitpress/9780262514620.001.0001 (cit. on pp. 46, 47).

[Marvin, 2018] Rebecca Marvin and Tal Linzen. "Targeted Syntactic Evaluation of Language Models". *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 1192–1202. DOI: 10.18653/v1/D18-1151 (cit. on pp. xvi, 54, 56, 69–73, 79, 82).

[McCann, 2018] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. "The Natural Language Decathlon: Multitask Learning as Question Answering". *arXiv preprint arXiv:1806.08730* (2018) (cit. on p. 26).

[McCoy, 2020] R. Thomas McCoy, Robert Frank, and Tal Linzen. "Does Syntax Need to Grow on Trees? Sources of Hierarchical Inductive Bias in Sequence-to-Sequence Networks". *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 125–140. DOI: 10.1162/tacl_a_00304 (cit. on p. 18).

[McCoy, 2018] R. Thomas McCoy, Tal Linzen, Ewan Dunbar, and Paul Smolensky. "RNNs Implicitly Implement Tensor Product Representations". *CoRR* abs/1812.08718 (2018). arXiv: 1812.08718 (cit. on p. 29).

[McCoy, 2019] Tom McCoy, Ellie Pavlick, and Tal Linzen. "Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference". *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 3428–3448. DOI: 10.18653/v1/P19-1334 (cit. on pp. 1, 25, 90, 124, 125).

[McMillan, 1988] Clayton McMillan and Paul Smolensky. *Analyzing a connectionist model as a system of soft rules.* Ed. by Psychology Press. 1988 (cit. on p. 48).

[Medsker, 2001] Larry R Medsker and LC Jain. "Recurrent neural networks". *Design and Applications* 5 (2001), pp. 64–67 (cit. on pp. 16, 23).

[Melamud, 2016] Oren Melamud, Jacob Goldberger, and Ido Dagan. "context2vec: Learning Generic Context Embedding with Bidirectional LSTM". *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 51–61. DOI: 10.18653/v1/K16-1006 (cit. on p. 17).

[Meng, 2017] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. "Deep Keyphrase Generation". *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 582–592. DOI: 10.18653/v1/P17-1054 (cit. on p. 26).

[Metheniti, 2020] Eleni Metheniti, Tim Van de Cruys, and Nabil Hathout. "How Relevant Are Selectional Preferences for Transformer-based Language Models?" *Proceedings of the 28th International Conference on Computational Linguistics*. 2020, pp. 1266–1278 (cit. on p. 91).

[Michel, 2019] Paul Michel, Omer Levy, and Graham Neubig. "Are Sixteen Heads Really Better than One?" *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019 (cit. on pp. 34, 35).

[Mikolov, 2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. "Recurrent neural network based language model." *Interspeech*. Vol. 2. 3. Makuhari. 2010, pp. 1045–1048 (cit. on pp. 16, 23).

[Miller, 1995] George A Miller. "WordNet: a lexical database for English". *Communications of the ACM* 38.11 (1995), pp. 39–41 (cit. on p. 79).

[Newman, 2021a] Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. "Refining Targeted Syntactic Evaluation of Language Models". *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Virtual: Association for Computational Linguistics, 2021 (cit. on pp. 33, 59).

[Newman, 2021b] Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. "Refining Targeted Syntactic Evaluation of Language Models". *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, 2021, pp. 3710–3723. DOI: 10.18653/v1/2021.naacl-main.290 (cit. on pp. 59, 70, 76, 90).

[OShea, 2015] Keiron O'Shea and Ryan Nash. "An Introduction to Convolutional Neural Networks". *ArXiv e-prints* (2015) (cit. on p. 17).

[Olstad, 2020] Anne Marte Haug Olstad, Isabella Fritz, and Giosuè Baggio. "Composition decomposed: Distinct neural mechanisms support processing of nouns in modification and predication contexts." *Journal of experimental psychology. Learning, memory, and cognition* (2020) (cit. on p. 48).

[Papadimitriou, 2021] Isabel Papadimitriou, Ethan A. Chi, Richard Futrell, and Kyle Mahowald. "Deep Subjecthood: Higher-Order Grammatical Features in Multilingual BERT". *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, 2021, pp. 2522–2532. DOI: 10.18653/v1/2021.eacl-main.215 (cit. on p. 32).

[Pascanu, 2012] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "Understanding the exploding gradient problem". *ArXiv* abs/1211.5063 (2012) (cit. on pp. 16, 23).

[Peirce, 2007] Jonathan W Peirce. "PsychoPy—psychophysics software in Python". *Journal of neuroscience methods* 162.1-2 (2007), pp. 8–13 (cit. on p. 81).

[Peters, 2018a] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, et al. "Deep Contextualized Word Representations". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202 (cit. on pp. 17, 31).

[Peters, 2018b] Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. "Dissecting Contextual Word Embeddings: Architecture and Representation". *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 1499–1509. DOI: 10.18653/v1/D18-1179 (cit. on p. 31).

[Pham, 2021] Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. "Out of Order: How important is the sequential order of words in a sentence in Natural Language Understanding tasks?" *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, 2021, pp. 1145–1160. DOI: 10.18653/v1/2021.findings-acl.98 (cit. on p. 106).

[Phillips, 2009] Colin Phillips. "Should we impeach armchair linguists?" *Japanese-Korean Linguistics* 17 (2009) (cit. on p. 50).

[Pimentel, 2021] Tiago Pimentel and Ryan Cotterell. "A Bayesian Framework for Information-Theoretic Probing". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 2869–2887. DOI: 10.18653/v1/2021.emnlp-main.229 (cit. on p. 32).

[Pimentel, 2020a] Tiago Pimentel, Naomi Saphra, Adina Williams, and Ryan Cotterell. "Pareto Probing: Trading Off Accuracy for Complexity". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 3138–3153. DOI: 10.18653/v1/2020.emnlp-main.254 (cit. on pp. 32, 159).

[Pimentel, 2020b] Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. "Information-Theoretic Probing for Linguistic Structure". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 4609–4622. DOI: 10.18653/v1/2020.acl-main.420 (cit. on pp. 32, 145).

[Pinker, 1988] Steven Pinker and Jacques Mehler, eds. *Connections and Symbols*. Cambridge, MA, USA: MIT Press, 1988 (cit. on p. 48).

[Press, 2021] Ofir Press, Noah A. Smith, and Mike Lewis. "Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation". *CoRR* abs/2108.12409 (2021). arXiv: 2108.12409 (cit. on pp. 19, 110, 111).

[Radford, 2018] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners" (2018) (cit. on pp. 17, 21).

[Raffel, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67 (cit. on pp. 1, 110).

[Rajpurkar, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "SQuAD: 100,000+ Questions for Machine Comprehension of Text". *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264 (cit. on pp. 18, 24).

[Ramesh, 2022] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. 2022 (cit. on p. 9).

[Ravfogel, 2020] Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. "Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 7237–7256. DOI: 10.18653/v1/2020.acl-main.647 (cit. on pp. 134, 147).

[Ravfogel, 2021] Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. "Counterfactual Interventions Reveal the Causal Effect of Relative Clause Representations on Agreement Prediction". *Proceedings of the 25th Conference on Computational Natural Language Learning*. Online: Association for Computational Linguistics, 2021, pp. 194–209. DOI: 10.18653/v1/2021.conll-1.15 (cit. on pp. 33, 159).

[Ravfogel, 2022a] Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan Cotterell. "Linear Adversarial Concept Erasure". *arXiv preprint arXiv:2201.12091* (2022) (cit. on p. 147).

[Ravfogel, 2022b] Shauli Ravfogel, Francisco Vargas, Yoav Goldberg, and Ryan Cotterell. "Adversarial Concept Erasure in Kernel Space". *arXiv preprint arXiv:2201.12191* (2022) (cit. on p. 147).

[Ribeiro, 2020] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. "Beyond Accuracy: Behavioral Testing of NLP Models with CheckList". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 4902–4912. DOI: 10.18653/v1/2020.acl-main.442 (cit. on p. 33).

[Riezler, 2002] Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. "Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques". *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002, pp. 271–278. DOI: 10.3115/1073083.1073129 (cit. on p. 44).

[Rosenblatt, 1958] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65 6 (1958), pp. 386–408 (cit. on p. 15).

[Salazar, 2020] Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. "Masked Language Model Scoring". *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 2699–2712. DOI: 10.18653/v1/2020.acl-main.240 (cit. on p. 20).

[Saussure, 1916] Ferdinand de Saussure. *Cours de linguistique générale*. Paris: Payot, 1916 (cit. on p. 42).

[Schlechtweg, 2021] Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. "DWUG: A large Resource of Diachronic Word Usage Graphs in Four Languages". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 7079–7091. DOI: 10.18653/v1/2021.emnlp-main.567 (cit. on p. 26).

[Shannon, 1948] Claude E. Shannon. “A mathematical theory of communication”. *The Bell system technical journal* 27.3 (1948), pp. 379–423 (cit. on p. 147).

[Shaw, 2018] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. “Self-Attention with Relative Position Representations”. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 464–468. DOI: 10.18653/v1/N18-2074 (cit. on p. 109).

[Shen, 2017] Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron Courville. “Neural Language Modeling by Jointly Learning Syntax and Lexicon” (2017) (cit. on p. 45).

[Shen, 2018] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. *Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks*. 2018 (cit. on p. 45).

[Shi, 2016] Xing Shi, Inkit Padhi, and Kevin Knight. “Does String-Based Neural MT Learn Source Syntax?” *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, 2016, pp. 1526–1534. DOI: 10.18653/v1/D16-1159 (cit. on p. 32).

[Shim, 2022] Kyuhong Shim and Wonyong Sung. *A Comparison of Transformer, Convolutional, and Recurrent Neural Networks on Phoneme Recognition*. 2022 (cit. on p. 24).

[Siegelmann, 1995] H.T. Siegelmann and E.D. Sontag. “On the Computational Power of Neural Nets”. *Journal of Computer and System Sciences* 50.1 (1995), pp. 132–150. DOI: https://doi.org/10.1006/jcss.1995.1013 (cit. on p. 24).

[Simonyan, 2015] K Simonyan and A Zisserman. “Very deep convolutional networks for large-scale image recognition”. Computational and Biological Learning Society, 2015, pp. 1–14 (cit. on p. 17).

[Sinha, 2021a] Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. “Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little”. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 2888–2913. DOI: 10.18653/v1/2021.emnlp-main.230 (cit. on pp. 106, 111).

[Sinha, 2021b] Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. “Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little”. *CoRR* abs/2104.06644 (2021). arXiv: 2104.06644 (cit. on p. 35).

[Sinha, 2021c] Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. "Masked language modeling and the distributional hypothesis: Order word matters pre-training for little". *arXiv preprint arXiv:2104.06644* (2021) (cit. on p. 32).

[Smolensky, 1990] Paul Smolensky. "Tensor product variable binding and the representation of symbolic structures in connectionist systems". *Artificial Intelligence* 46.1 (1990), pp. 159–216. DOI: https://doi.org/10.1016/0004-3702(90)90007-M (cit. on p. 30).

[Soutner, 2013] Daniel Soutner and Luděk Müller. "Application of LSTM Neural Networks in Language Modelling". Vol. 8082. 2013, pp. 105–112. DOI: 10.1007/978-3-642-40585-3_14 (cit. on p. 16).

[Speer, 2018] Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. *LuminosoInsight/wordfreq: v2.2*. 2018. DOI: 10.5281/zenodo.1443582 (cit. on p. 79).

[Sperduti, 1997] Alessandro Sperduti. "On the Computational Power of Recurrent Neural Networks for Structures". *Neural Networks* 10.3 (1997), pp. 395–400. DOI: https://doi.org/10.1016/S0893-6080(96)00105-0 (cit. on p. 24).

[Spithourakis, 2018] Georgios Spithourakis and Sebastian Riedel. "Numeracy for Language Models: Evaluating and Improving their Ability to Predict Numbers". *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 2104–2115. DOI: 10.18653/v1/P18-1196 (cit. on p. 144).

[Sprouse, 2013] Jon Sprouse and Diogo Almeida. "The empirical status of data in syntax: A reply to Gibson and Fedorenko". *Language and Cognitive Processes* 28 (2013), pp. 222–228. DOI: 10.1080/01690965.2012.703782 (cit. on p. 50).

[Steedman, 2000] Mark Steedman. *The Syntactic Process*. Cambridge, MA, USA: MIT Press, 2000 (cit. on p. 29).

[Su, 2021] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. "RoFormer: Enhanced Transformer with Rotary Position Embedding". *CoRR* abs/2104.09864 (2021). arXiv: 2104.09864 (cit. on pp. 19, 111).

[Sundermeyer, 2012] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. "LSTM Neural Networks for Language Modeling". 2012. DOI: 10.21437/Interspeech.2012-65 (cit. on p. 16).

[Tenney, 2019a] Ian Tenney, Dipanjan Das, and Ellie Pavlick. "BERT Rediscovers the Classical NLP Pipeline". *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 4593–4601. DOI: 10.18653/v1/P19-1452 (cit. on p. 32).

[Tenney, 2019b] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, et al. "What do you learn from context? Probing for sentence structure in contextualized word representations". *International Conference on Learning Representations*. 2019 (cit. on pp. 31, 32).

[Thompson, 1993] Henry S. Thompson, Anne Anderson, Ellen Gurman Bard, Gwyneth Doherty-Sneddon, Alison Newlands, and Cathy Sotillo. "The HCRC Map Task Corpus: Natural Dialogue for Speech Recognition". *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*. 1993 (cit. on p. 26).

[Torroba Hennigen, 2020] Lucas Torroba Hennigen, Adina Williams, and Ryan Cotterell. "Intrinsic Probing through Dimension Selection". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 197–216. DOI: 10.18653/v1/2020.emnlp-main.15 (cit. on pp. 134, 144).

[Tucker, 2021] Mycal Tucker, Peng Qian, and Roger Levy. "What if This Modified That? Syntactic Interventions with Counterfactual Embeddings". *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, 2021, pp. 862–875. DOI: 10.18653/v1/2021.findings-acl.76 (cit. on p. 33).

[Vaswani, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, et al. "Attention is All you Need". *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, et al. Vol. 30. Curran Associates, Inc., 2017 (cit. on pp. 4, 17, 19, 21, 108).

[Vig, 2020a] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, et al. "Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias". *CoRR* abs/2004.12265 (2020). arXiv: 2004.12265 (cit. on p. 33).

[Vig, 2020b] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, et al. "Investigating Gender Bias in Language Models Using Causal Mediation Analysis". *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 12388–12401 (cit. on pp. 33, 129, 134).

[Voita, 2020] Elena Voita and Ivan Titov. "Information-Theoretic Probing with Minimum Description Length". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 183–196. DOI: 10.18653/v1/2020.emnlp-main.14 (cit. on pp. 32, 135, 145, 159).

[Wallace, 2019] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. "Do NLP Models Know Numbers? Probing Numeracy in Embeddings". *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 5307–5315. DOI: 10.18653/v1/D19-1534 (cit. on p. 144).

[Wang, 2019] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, et al. "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems". *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019 (cit. on pp. 1, 18, 24, 26).

[Wang, 2018] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 353–355. DOI: 10.18653/v1/W18-5446 (cit. on pp. 1, 18, 24, 26, 35).

[Wang, 2016a] Dingquan Wang and Jason Eisner. "The Galactic Dependencies Treebanks: Getting More Data by Synthesizing New Languages". *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 491–505. DOI: 10.1162/tacl_a_00113 (cit. on p. 112).

[Wang, 2016b] Shuohang Wang and Jing Jiang. "Learning Natural Language Inference with LSTM". *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, 2016, pp. 1442–1451. DOI: 10.18653/v1/N16-1170 (cit. on p. 16).

[Warstadt, 2020a] Alex Warstadt and Samuel R. Bowman. "Can neural networks acquire a structural bias from raw linguistic data?" *Proceedings of the Cognitive Science Society (CogSci)*. 2020 (cit. on p. 33).

[Warstadt, 2020b] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, et al. "BLiMP: The Benchmark of Linguistic Minimal Pairs for English". *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 377–392. DOI: 10.1162/tacl_a_00321 (cit. on p. 54).

[Warstadt, 2019] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. "Neural Network Acceptability Judgments". *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 625–641. DOI: 10.1162/tacl_a_00290 (cit. on pp. 45, 56).

[Warstadt, 2020c] Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. "Learning Which Features Matter: RoBERTa Acquires a Preference for Linguistic Generalizations (Eventually)". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 217–235. DOI: 10.18653/v1/2020.emnlp-main.16 (cit. on pp. 33, 90).

[Wei, 2021a] Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. "Frequency Effects on Syntactic Rule Learning in Transformers". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 932–948 (cit. on p. 90).

[Wei, 2021b] Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. "Frequency Effects on Syntactic Rule Learning in Transformers". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021, pp. 932–948. DOI: 10.18653/v1/2021.emnlp-main.72 (cit. on p. 160).

[Wettig, 2022] Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. *Should You Mask 15% in Masked Language Modeling?* 2022. DOI: 10.48550/ARXIV.2202.08005 (cit. on p. 118).

[White, 2021a] Jennifer C. White and Ryan Cotterell. "Examining the Inductive Bias of Neural Language Models with Artificial Languages". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, 2021, pp. 454–463. DOI: 10.18653/v1/2021.acl-long.38 (cit. on pp. 112, 113).

[White, 2021b] Jennifer C. White, Tiago Pimentel, Naomi Saphra, and Ryan Cotterell. "A Non-Linear Structural Probe". *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, 2021, pp. 132–138. DOI: 10.18653/v1/2021.naacl-main.12 (cit. on pp. 29, 30).

[Williams, 2018] Adina Williams, Nikita Nangia, and Samuel Bowman. "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 1112–1122. DOI: 10.18653/v1/N18-1101 (cit. on p. 25).

[Williams, 1980] Edwin Williams. "Predication". *Linguistic Inquiry* 11.1 (1980), pp. 203–238 (cit. on p. 56).

[Wolf, 2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, et al. "Transformers: State-of-the-Art Natural Language Processing". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6 (cit. on pp. 1, 114, 149).

[Xu, 2020a] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. *A Theory of Usable Information Under Computational Constraints*. 2020 (cit. on p. 146).

[Xu, 2020b] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. "A Theory of Usable Information under Computational Constraints". *International Conference on Learning Representations*. 2020 (cit. on pp. 135, 146).

[Yang, 2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. "XLNet: Generalized Autoregressive Pretraining for Language Understanding". *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019 (cit. on p. 21).

[Yao, 2021] Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. "Self-Attention Networks Can Process Bounded Hierarchical Languages". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, 2021, pp. 3770–3785. DOI: 10.18653/v1/2021.acl-long.292 (cit. on p. 18).

[Yu, 2020a] Charles Yu, Ryan Sie, Nicolas Tedeschi, and Leon Bergen. "Word Frequency Does Not Predict Grammatical Knowledge in Language Models". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, pp. 4040–4054. DOI: 10.18653/v1/2020.emnlp-main.331 (cit. on p. 90).

[Yu, 2020b] Charles Yu, Ryan Sie, Nicolas Tedeschi, and Leon Bergen. "Word Frequency Does Not Predict Grammatical Knowledge in Language Models". *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 4040–4054 (cit. on p. 90).

[Zaccarella, 2015] Emiliano Zaccarella, Lars Meyer, Michiru Makuuchi, and Angela D. Friederici. "Building by Syntax: The Neural Basis of Minimal Linguistic Structures". *Cerebral Cortex* 27.1 (2015), pp. 411–421. DOI: 10.1093/cercor/bhv234. eprint: https://academic.oup.com/cercor/article-pdf/27/1/411/25159795/bhv234.pdf (cit. on p. 48).

[Zhang, 2017] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. “Understanding deep learning requires rethinking generalization”. *International Conference on Learning Representations*. 2017 (cit. on p. 5).

[Zhang, 2021] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. “Understanding Deep Learning (Still) Requires Rethinking Generalization”. *Commun. ACM* 64.3 (2021), pp. 107–115. DOI: 10.1145/3446776 (cit. on p. 5).

[Zhang, 2019] Hongming Zhang, Ding Hantian, and Yangqiu Song. “SP-10K: A Large-scale Evaluation Set for Selectional Preference Acquisition”. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 722–731 (cit. on p. 92).

[Zhang, 2018] Kelly Zhang and Samuel Bowman. “Language Modeling Teaches You More than Translation Does: Lessons Learned Through Auxiliary Syntactic Task Analysis”. *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 359–361. DOI: 10.18653/v1/W18-5448 (cit. on p. 32).

[Zhou, 2019] Junru Zhou and Hai Zhao. “Head-Driven Phrase Structure Grammar Parsing on Penn Treebank”. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 2396–2408. DOI: 10.18653/v1/P19-1230 (cit. on p. 44).

RÉSUMÉ

Les modèles de langage neuronaux basés sur des transformeurs sont couramment déployés pour effectuer diverses tâches de traitement automatique des langues, car ils produisent des représentations vectorielles de textes qui peuvent être utilisées dans le cadre d'un apprentissage supervisé. Après avoir été pré-entraînés comme modèles de langage génériques, ils atteignent des performances spectaculaires sur un large éventail de tâches, dont plusieurs nécessitent en principe des connaissances sur la structure des phrases. Ces modèles ne sont pas explicitement supervisés avec la moindre instruction grammaticale, ce qui suggère que ces connaissances émergent pendant la phase de pré-entraînement. La nature des capacités acquises est peu comprise, car ces modèles sont généralement utilisés comme des boîtes noires. Leurs décisions sont en outre difficiles à interpréter, en raison de leur grand nombre de paramètres (jusqu'à 10^{12} pour les architectures les plus récentes) et de la complexité des fonctions apprises. De ce constat a émergé un nombre important de travaux de recherche visant à mieux comprendre les capacités linguistiques de ces modèles. Bien que cette littérature soit abondante, les paradigmes épistémologiques sous-tendant les différentes méthodologies ne sont pas compatibles entre eux, ce qui souligne la nécessité de formuler plus clairement les questions portant sur l'acquisition des connaissances linguistiques. Tout au long de la thèse, nous tentons de combler le fossé épistémique entre ces facettes en formulant explicitement les relations qui lient les approches existantes. En particulier, nous adoptons trois niveaux d'analyse pour comprendre les modèles de langage neuronaux en tant que systèmes de traitement de l'information, du plus haut au plus profond : le niveau comportemental, le niveau algorithmique et le niveau de l'implémentation. La partie expérimentale de cette thèse introduit d'abord des tests comportementaux évaluant le niveau d'abstraction syntaxique, afin d'étudier la nature des informations traitées au niveau algorithmique - en particulier nous mettons en évidence l'intrication entre les processus syntaxiques et sémantiques. Nous montrons ensuite que les tests comportementaux sont limités pour nous renseigner sur la nature de l'information traitée par le modèle. En particulier, nous montrons que les prédictions sur une tâche dépendant de la structure des phrases peuvent être approximées sans faire usage d'information sur l'ordre des mots. Face à ce problème, nous soulignons la nécessité de réaliser des interventions causales et nous étudions l'utilisation de l'information positionnelle par un modèle de langage neuronal. Nous prouvons que le modèle s'appuie graduellement sur l'ordre des mots à mesure que le nombre de mots masqués au cours de l'entraînement augmente. Nous démontrons également le pouvoir explicatif des interventions causales pour évaluer l'utilisation d'information de manière ciblée et non équivoque. Nous montrons ensuite comment les interventions causales, couplées à des tests de comportement, peuvent nous renseigner sur les représentations linguistiques du modèle aux trois niveaux mentionnés précédemment. Le cadre introduit permet (i) de mettre en évidence le substrat neuronal responsable de la représentation et du transfert d'information linguistique, (ii) d'évaluer la présence de représentations et d'opérations au niveau algorithmique et (iii) de déterminer l'influence causale de ces dernières sur le comportement du modèle. Nous appliquons ensuite la méthodologie introduite pour mettre en lumière l'encodage et l'utilisation de l'information sur le nombre grammatical dans le cadre d'une tâche d'accord sujet-verbe. Ainsi, nous comblons le fossé entre les études sur les capacités d'abstraction linguistique focalisées sur les représentations du modèle et celles axées sur son comportement de surface.

MOTS CLÉS

Apprentissage Profond, Modèle de Langage, Connaissance Linguistique, Traitement Automatique des Langues, Généralisation

ABSTRACT

Neural language models are commonly deployed to perform diverse natural language processing tasks, as they produce contextual vector representations of words and sentences which can be used in any supervised learning setting. In recent years, transformer-based neural architectures have been widely adopted towards this end. After being pre-trained with a generic language modeling objective, they achieve spectacular performance on a wide array of downstream tasks, several of which should in principle require knowledge of sentence structure. As these models are not explicitly supervised with any grammatical instruction, this suggests that linguistic knowledge emerges during the pre-training stage. The nature of the linguistic abilities acquired during training is still scarcely understood, as these models are generally used as black boxes. Their decisions are hard to interpret, as they generally possess a great number of parameters (up to 10^{12} for the most recent architectures) and learn very complex functions. These observations led to the emergence of a growing body of research aimed at uncovering the linguistic abilities of such models. While this literature is very abundant, the epistemic grounds of the different methodologies are not translatable into each other, which underlines the need to formulate more clearly the questions addressing the capture of linguistic knowledge. To this end, we identify the different stances on the greater problem: in addition to downstream performance, evidence for a trained model's linguistic abilities can be sought in its components, representations and surface behavior. Throughout the thesis, we attempt bridging the epistemic gap between these facets by formulating explicitly the relations which lie between these different subproblems. In particular, we adopt three levels of analysis to understand neural language models as information processing systems, from the highest to the deepest: the behavioral level, the algorithmic level, and the implementational level. In our framework, our departing point to investigate linguistic abilities is surface linguistic generalization. The empirical portion of this thesis first presents behavioral tests targeting a syntactic ability, to investigate the nature of the information processed at the algorithmic level - in particular we provide evidence for the entanglement between syntactic and semantic processes. We then show that behavioral tests can be limited to inform us on the nature of the information processed by the model. In particular, we provide evidence that surface behavior on a structure-sensitive task can be approximated to a good extent without relying on word order. Faced with this observation, we make the case for targeted causal interventions, and investigate a neural language model's reliance on position information on the masked language modeling task. In doing so, we show that the model increasingly relies on word order as the number of masked tokens increases during training. We also demonstrate the power of causal interventions to assess the usage of targeted information. We then discuss how causal interventions, coupled with targeted behavior tests, can inform us on the model's linguistic abilities at the three levels mentioned previously. Indeed, the introduced framework allows us to (i) find the neural substrate responsible for representing or transferring linguistic information, (ii) assess the presence of certain representations or operations at the algorithmic level and (iii) determine the causal influence of these representations and operations over the model's behavior. We discuss how this analysis can be performed and apply the methodology introduced to shed light on the encoding and usage of grammatical number information on the subject-verb agreement task. In doing so, we bridge the gap between representation-oriented and behavior-oriented analyses of linguistic knowledge.

KEYWORDS

Deep Learning, Language Model, Linguistic Knowledge, Natural Language Processing, Generalization