Study of the abstraction capabilities of neural language models

Bingzhi Li

Université Paris Cité
École doctorale Sciences du Language (ED 622)
Laboratoire de Linguistique Formelle (LLF)

Study of the Abstraction Capabilities of Neural Language Models

Par Bingzhi LI

Thèse de doctorat de SCIENCES DU LANGUAGE

Dirigée par Benoît CRABBÉ
Et co-encadrée par Guillaume WISNIEWSKI

Présentée et soutenue publiquement le 28 novembre 2023

Devant un jury composé de:

Barbara HEMFORTH, DR, CNRS et Université Paris Cité, présidente du jury
Thierry POIBEAU, DR, CNRS, ENS-PSL et Université Sorbonne Nouvelle, rapporteur
François YVON, DR, CNRS et Sorbonne Université, rapporteur
Dieuwke HUPKES, PhD, Meta AI, examinatrice
Benoît CRABBÉ, PR, Université Paris Cité, directeur de thèse
Guillaume WISNIEWSKI, MCF, Université Paris Cité, co-encadrant de thèse

Titre : Étude des capacités abstractives de modèles de langue neuronaux

Résumé (court) : Les théories linguistiques traditionnelles postulent que la compétence linguistique humaine est fondée sur des propriétés structurelles innées et des représentations symboliques. Cependant, les modèles de langue à base de Transformeurs excellent dans diverses tâches de traitement automatique des langues (TAL) sans intégrer explicitement de tels prérequis linguistiques. Leur succès empirique remet en question ces hypothèses linguistiques établies et soulève des interrogations sur les mécanismes sous-jacents des modèles. Cependant, leur opacité et complexité, liées à un grand nombre de paramètres, rendent difficile la compréhension de leur fonctionnement interne. Cette thèse vise à éclaircir si les Transformeurs se basent essentiellement sur la reconnaissance de motifs superficiels pour représenter des structures syntaxiques, ou s'ils sont capables d'abstraire implicitement des règles plus générales. Deux objectifs principaux guident cette recherche : i) évaluer le potentiel du modèle de langue Transformeur autoregressif comme outil explicatif du traitement syntaxique humain ; ii) améliorer l'interprétabilité du modèle. Nous abordons ces objectifs en examinant les abstractions syntaxiques des modèles Transformeur sur deux niveaux : leur capacité à modéliser des structures hiérarchiques, et leur capacité à généraliser compositionnellement des structures observées. Nous introduisons un cadre d'analyse intégré comprenant trois niveaux interdépendants : évaluation comportementale à travers des ensembles de test de défis, analyse représentationnelle à l'aide de sondes linguistiques, et analyse fonctionnelle par interventions causales. Nous évaluons d'abord le modèle sur des tests syntaxiques afin de déterminer sa capacité à reproduire le comportement linguistique humain. Ensuite, nous utilisons des sondes linguistiques et des interventions causales pour mesurer l'adéquation des représentations internes du modèle avec les théories linguistiques établies. Nos résultats montrent que les Transformeurs parviennent à représenter des structures hiérarchiques pour une généralisation syntaxique nuancée. Cependant, au lieu de s'appuyer sur des règles compositionnelles systématiques, il semble qu'ils se basent davantage sur l'abstraction lexico-catégorielle et des analogies structurelles. Si cela leur permet de gérer une forme sophistiquée de productivité grammaticale pour des structures familières, ils rencontrent des difficultés avec des structures qui nécessitent une application systématique des règles compositionnelles. Cette étude met en évidence à la fois la promesse et les limitations potentielles des modèles Transformeur autoregressifs comme outils explicatifs pour le traitement syntaxique humain, et fournit un cadre méthodologique pour leur analyse et leur interprétabilité.

Mots-clés: Traitement automatique des langues, modèles de langue neuronaux, interprétabilité, généralisation, abstraction linguistique, representation syntaxique, structures hiérarchiques, compositionalité

Title: Study of the abstraction capabilities of neural language models

Abstract: Traditional linguistic theories have long posited that human language competence is founded on innate structural properties and symbolic representations. However, Transformer-based language models, which learn language representations from unannotated text, have excelled in various natural language processing (NLP) tasks without explicitly modeling such linguistic priors. Their empirical success challenges these long-standing linguistic assumptions and also raises questions about the models' underlying mechanisms for linguistic competence. However, the black-box nature and complexity of these models, due to their numerous parameters, make it difficult to understand their internal workings. While research in this area is growing, the extent of their linguistic abstraction capabilities remains an open question. This thesis seeks to determine whether Transformer models primarily rely on surface-level patterns for representing syntactic structures, or if they also implicitly capture more abstract rules. The study serves two main objectives: i) assessing the potential of an autoregressive Transformer language model as an explanatory tool for human syntactic processing; ii) enhancing the model's interpretability. To achieve these goals, we assess the syntactic abstractions in Transformer models on two levels: first, the ability to represent hierarchical structures, and second, the ability to compositionally generalize observed structures. We introduce an integrated linguistically-informed analysis framework that consists of three interrelated layers: behavioral assessment through challenge sets, representational probing using linguistic probes, and functional analysis through causal intervention. Our analysis starts with assessing the model's performance on syntactic challenge sets to see how closely it mirrors human language behavior. Following this, we use linguistic probes and causal interventions to assess how well the model's internal representations align with established linguistic theories. Our findings reveal that Transformers manage to represent hierarchical structures for nuanced syntactic generalization. However, instead of relying on systematic compositional rules, they seem to lean more towards lexico-categorical abstraction and structural analogies. While this allows them to handle a sophisticated form of grammatical productivity for familiar structures, they encounter challenges with structures that require a systematic application of compositional rules. This study highlights both the promise and potential limitations of autoregressive Transformer models as explanatory tools for human syntactic processing, and provides a methodological framework for its analysis and interpretability.

Keywords: natural language processing, neural language models, linguistic abstraction, interpretability, generalization, syntactic representation, hierarchical structures, compositionality

Acknowledgments

As I pen down the final page of this thesis, I am overwhelmed with gratitude for the transformative past three years. It has been a journey filled with challenges, doubts, and substantial learning in between. The growth I have experienced and the completion of this thesis would not have been possible without the continuous support, guidance, and encouragement from many.

First and foremost, my deepest gratitude goes to my advisors, Guillaume Wisniewski and Benoît Crabbé, to whom I owe nearly everything I know about research. I really appreciate the enlightening discussions during our weekly meetings, reading groups, and even over lunch breaks. Benoît, I profoundly admire your expertise in both linguistics and NLP. Thank you for your insightful and detailed explanations of scientific concepts, and for helping me to find my own research path. Guillaume, you introduced me to the academic world, guiding me step by step in research. I am grateful for your trust and sense of humor, which have carried me through many challenging moments. You ingrained in me the belief that everything is learnable, a lesson that has deeply shaped me into a better version of myself.

I am grateful to Tal Linzen and Najoung Kim for their invaluable mentorship during my visit to NYU. My appreciation extends to the NYU Computation and Psycholinguistics Lab team for very helpful discussion and to Alexander, Yuekun, and Lucia for their collaboration. I also thank Labex EFL and LLF for funding this opportunity.

Dear committee members, I appreciate the time and effort you have invested in reviewing my thesis and accompanying me during the final stage of my PhD.

To the friends and colleagues from my PhD journey, your help and support have been crucial to me. Special thanks to Nathanaël, Juliette, Antoine, Anna, David, Julie, Marie, Timothée, Huiyi, Clémentine, Yiming, Fang, Aixiu, Chuyuan, Maria, Zhanglin, Dorotea, Ensieh, Saida, Zulipiye. I extend my heartfelt gratitude to all the members of LLF for their invaluable support.

A warm thank you to my husband, Bo, a programmer, for your support, and patience through my numerous practice talks.

Lastly, I am grateful to my parents for instilling in me the courage and wisdom to pursue my passions. I still remember a moment from my childhood during the busy harvest season; as I stood before the seemingly endless rice fields awaiting harvest, I felt overwhelmed and scared. You looked at the sickle in my hands and reassured me, “Just start with the stalk of rice at your feet, and we will get through the entire field”. This simple phrase has stayed with me as I ventured beyond our mountain village, beyond our small county, into university, and out into the vast world, surmounting numerous challenges. I aspire that my future work will, in some way, connect back to the land that my parents cherish so dearly.

Contents

Abstract i

Acknowledgments iii

List of figures ix

List of tables xii

List of abbreviations xiii

1 Introduction 1

1.1 Research question and objectives 2

1.1.1 Feasibility as explanatory model for human language processing . 3

1.1.2 Improving model interpretability 6

1.2 Contributions 7

1.2.1 Assessing model capacity to represent syntactic structures 7

1.2.2 Assessing model capacity to generalize compositionally observed structures 8

1.2.3 Publications 8

1.3 Outline 9

I Background 11

2 Structure of language and neural language models 12

2.1 Structure in human language 13

2.2 Neural language models 15

2.2.1 Language modeling 15

2.2.2 Transformer-based neural language model 22

2.3 Analysis of linguistic structure in neural NLP models 28

2.3.1 Challenge sets 28

2.3.2 Probing classifiers 31

2.3.3 Causal intervention analysis 34

II Assessing model capacity to represent syntactic structures 36

3 Long-distance agreement in neural language models 37

4 A contrastive study of NLM's syntactic abstraction based on long-distance agreement 44
4.1 Introduction . . . 45
4.2 Revisiting number agreement tasks via a heuristic-based evaluation protocol 47
4.2.1 Syntactic phenomena . . . 48
4.2.2 Datasets construction . . . 50
4.2.3 Experimental setup . . . 53
4.2.4 Heuristic-based evaluation protocol . . . 56
4.2.5 Control experiments . . . 61
4.2.6 Conclusion . . . 69
4.3 Locating syntactic information in Transformer language model . . . 70
4.3.1 Distribution of syntactic agreement information across token positions 71
4.3.2 Probing internal representations components . . . 76
4.3.3 Conclusion . . . 78
4.4 Right for the right reason: Exploring mechanisms of agreement computations 79
4.4.1 The Causal Framework . . . 80
4.4.2 Causal experiments and results . . . 85
4.4.3 Conclusion . . . 88
4.5 Word order: the impact of positional encoding on NLM's syntactic abstraction capacity . . . 89
4.5.1 Positional embeddings in Autoregressive Transformer LM . . . 90
4.5.2 Positional embeddings in masked Transformer LM . . . 91
4.5.3 Conclusion . . . 93
4.6 Conclusion and discussion . . . 93

III Assessing model capacity to generalize compositionally observed structures 96

5 SLOG: A Structural Generalization Test for Semantic Parsing 97
5.1 Introduction . . . 99
5.2 Overview of SLOG benchmark . . . 103
5.2.1 Novel recursion depth . . . 105
5.2.2 Novel combination of modified phrases and grammatical roles . . . 107
5.2.3 Novel gap positions . . . 109
5.2.4 Novel *wh*-questions . . . 110
5.3 Dataset generation . . . 111
5.4 Experimental setup . . . 113

5.4.1 Models . . . 113
5.4.2 Evaluation metric . . . 115
5.5 Results . . . 115
5.5.1 Unobserved depth and length both affect depth generalization . . . 117
5.5.2 Unobserved long-distance dependencies make generalization difficult 118
5.5.3 Gap generalizations are challenging for all tested models . . . 119
5.6 Conclusion . . . 124

IV Conclusion 126

6 Conclusions and perspectives 127
6.1 Conclusions . . . 127
6.2 Future work . . . 130

Bibliography 132

Résumé 157

Appendix 162

List of Figures

2.1 Syntactic tree: a hierarchical representation of word organization in sentences 14

2.2 The sentence's meaning is derived compositionally from the meaning of its components, in line with its syntactic structure in Figure 2.1 14

2.3 Forward inference in a feed-forward neural language model with window size of two, at timestep $i + 1$. To predict the next word w_{i+1}, the model concatenates embeddings of the two preceding words, $\mathbf{e}_i$ and $\mathbf{e}_{i-1}$, multiplies them by $\mathbf{W}_{in}$, and applies an activation function to produce the hidden layer. This layer is then transformed by $\mathbf{W}_{out}$ and a softmax to estimate the probability of each word in its vocabulary being the next word w_{i+1}. . . . 18

2.4 Forward inference in an RNN language model at timestep $i + 1$. To predict the next word after the context "A cat on the mat", the model takes the embedding of the current word 'mat' and multiplies it by W_{in}. Concurrently, it multiplies the hidden layer of the previous timestep h_{i-1} by W_{rec}. These values are summed and passed through an activation function to produce the current hidden layer, h^i, which is then transformed by $\mathbf{W}_{out}$ and a softmax to produce a probability distribution over the vocabulary. 19

2.5 Components of a typical Transformer block: self-attention, feed-forward network, layer normalization, and residual connections. 23

2.6 Masked self-attention in autoregressive Transformer language models: each token is processed considering all the preceding tokens and itself, future tokens are excluded. 24

4.1 In (a), the number of the main verb (*miaulent*, in red) is determined by the head of the subject *chats*. In (b), the past participle in the relative clause (*adopté*, in blue) has to agree in gender and number with its object (also in blue) when the latter precedes the verb. 49

4.3 The test set excludes sentences with coordinate **cue** as shown in (1). But it includes syntactic phrases as **cue** cases like in (2), as the antecedent of the relative pronoun is unambiguous. 52

4.4 Average accuracy of LSTM (indicated by lighter color bars) and Transformer models on the *Nonce set*, represented by orange bars, and the *Original set*, indicated by blue bars. 63

4.5 Frequency ratio of target form to competing form. For instance, for the *S-V sing* condition, a ratio of 10^1 indicates that the target verb form (singular) occurs 10 times more frequently in the pretraining data than its competing form (plural). 65

4.6 Absolute frequency of **target** verbs in pre-training data, with medians displayed in white numbers 66
4.7 Comparison of models' accuracy in two agreement tasks using top-3 evaluation metric (orange bars) and *target verb* evaluation metric (blue bars). . . 68
4.8 For the O-PP agreement, the prefix is highlighted in blue, the context in yellow and the suffix in green. 72
4.9 Average probing accuracy at each position based on the number of the *cue*. The b_i (resp. a_i) position denotes the i-th token before (resp. after) the pattern. The position labeled as 'Noun' corresponds to a noun with the opposite number as the *cue* in the 1-attractor subset, and a noun with the same number as the *cue* in the 0-attractor subset. 75
4.10 Probing accuracy as a function of the count of dimensions (for 768-dimension token representations) with non-zero coefficients, obtained through feature selection using ℓ_1 regularized logistic regression for each position within *context*. The X-axis denotes the count of non-zero coefficient dimensions, and the Y-axis represents probing accuracy. Vertical dashed lines indicate the points at which the accuracy reaches 90%. 77
4.11 With the initial masked self-attention mechanism, the next token representation is computed as a weighted sum of all previous token representations. To assess the impact of "que" on the model's agreement behavior, the causal intervention involves cutting the direct attention from the **target** position to the token "que" (denoted by ✗), and then comparing the Transformer's prediction before and after this intervention. 81
4.12 Causal model showing dependencies between the attention weights A^l at layer l, the **target**'s contextualized representation r_t and Transformer's predicated agreement feature $\mathcal{A}$. 82
4.13 Target representation as a linear combination of all preceding token representations, weighted by their attention scores $A_t^l =< \alpha_p, \alpha_c, \alpha_q, \alpha_i >$ 83
4.14 Average causal effect of interventions on Transformer's NA task performance, quantified by drop in accuracy before and after different interventions, and further broken down based on prediction difficulty measured by the number of heuristics. The term *cue* here refers to the antecedent and its modifiers (determiners and adjectives) in O-PP agreement, and to the subject and its modifiers in S-V agreement. 86

4.15 Accuracy comparison of autoregressive Transformer LM on two NA tasks with and without positional embeddings. Detailed scores are reported in Appendix Table A.10. 91
4.16 Masked Transformer LM's accuracy on two NA tasks with and without positional embeddings. Detailed scores are reported in Appendix Table A.11 92

5.1 Examples of lexical generalization in COGS — (a), and structural generalization in COGS — (b) and in SLOG — (b, c). The SLOG task requires mapping the generalization examples to their logical forms; the corresponding logical forms are shown in Table 5.1. 101
5.2 Example of an AM dependency tree: (b) displays the supertags assigned to each token, while (a) presents the dependency tree connecting them. . . . 113
5.3 Accuracy on SLOG, with error bars indicating variations across five runs. We also show the best published results on COGS (indicated with †), as reported in Yao and Koller (2022). 116
5.4 Aggregate accuracy on SLOG by generalization category, with error bars denoting the variations across generalization cases within each category over five model runs. 116
5.5 AM dependency tree for a direct object *wh*-question. (a) displays the gold supertags and (b) shows the incorrect predicted supertags. 121
5.6 Example of gold AM dependency tree for *wh*-questions with long movement 123
5.7 Example of predicted AM dependency tree for *wh*-questions with long movement 123

A.1 Probing accuracy based on tokens PoS tags and their positions in the sentences, from left to right: *prefix*, *context*, *suffix* 177

List of Tables

2.1 A categorization of some representative studies using probing classifiers to investigate syntactic structures in NLMs, according to linguistic properties examined, classifier types, probed models, and baseline models. 33

4.1 Parameters and perplexities (average across five models) of neural language models examined in this section. 55

4.2 Average accuracy (%) for both agreement tasks across five models for each architecture, compared to baselines. 56

4.3 Examples from our evaluation set of subject-verb agreement, stratified by the count of surface heuristics predicting the *target*'s number, a proxy to the task difficulty. The target verbs and their subjects are in bold. The orange numbers in parentheses indicate the presence of different types of heuristics. 58

4.4 Accuracies(%) achieved by LSTM and Transformer models as a function of the agreement prediction difficulty. Transformer model $\mathcal{M}$ uses 16-layer decoders each with 16 heads, $\mathcal{M}_{shallow}$ has 2 layers each with 16 heads and $\mathcal{M}_{shared}$ is a variant of $\mathcal{M}$ using shared parameters across all 16 layers. . 59

4.5 Accuracy breakdown based on whether the **cue-target** pair was seen (occurrence > 0) or unseen (occurrence = 0) during pre-training. The baseline $\texttt{argmax}_v\texttt{(}\textbf{cue-target}\texttt{)}$ consistently predicts the more frequently observed pairs. If both the target and competing pairs were unseen, this baseline model randomly selects one pair. 61

4.6 Accuracy breakdown based on the grammatical number of the **target**. The baseline $\texttt{argmax}_v\texttt{(target)}$ consistently predicts the more frequently observed number of the **target**. 64

4.7 Probing results across different sentence parts (see Figure 4.1). The reported scores represent the average accuracy of all PoS-based classifiers for each sentence segment. 73

4.8 Comparison of log-probabilities for each token of example sentences processed by our Transformer LM, before and after the intervention on "que". Sentences contain either the plural form of the target verb *acceptés*, or its singular form *accepté*. $\mathcal{A}$-column: 1 indicates a predicted agreement feature matching the gold label, 0 indicates no match. 81

5.1 Examples of two distinct types of generalization: lexical generalization in COGS — (a), structural generalization in COGS—(b) and in SLOG — (b, c). The symbol $\rightsquigarrow$ indicates the task of translating an English sentence into its corresponding meaning representation. 104
5.2 A full list of SLOG generalization cases. Each sentence in the table corresponds to a (sentence, logical form) pair, as illustrated in Figure 5.1. $\oplus$ denotes the combination of two observed structures, which allows to interpret the target novel structure. Some cases cover multiple sub-case constructions: e.g. all ditransitive verbs include both double-object and prepositional constructions. Due to space limitations, only one example is provided for each case. The three cases marked with '✓' are already present in the COGS dataset. 106
5.3 Mean accuracy (%) on unseen deeper recursion cases within and beyond the range of training output lengths (maximum training output = 229 tokens). 117
5.4 Performance of PP modification generalization broken down by construction. Bold orange words denote long predicate-argument dependencies, while bold black words indicate short ones. 119

A.1 Hyperparameter configurations for each model and their corresponding average perplexity scores. † denotes pseudo-perplexity scores used for MLM evaluation (§4.5.2), not comparable with conventional perplexity scores. . 163
A.2 Verbs (at least 10 sentences) yielding the highest and lowest accuracy for the Transformer-LM. 'Target Occurrences' refers to the frequency of the target form in pretraining data. 'Ratio' signifies the frequency ratio of the target form to its competing form (i.e., with the opposite number) in the pretraining data. 168
A.3 Examples of sentences for the worst performing verbs and past participles in Table A.2, the words in bold indicate the **cue**-**target** pairs. 169
A.4 Accuracy (%) achieved by our models (averaged across 5 models for each architecture), compared to accuracies predicted by the 5 surface heuristics considered in this work on long-distance agreement tasks. 176
A.5 Average accuracy (%) of LSTM and Transformer models on the *Nonce set* versus the *Original set* by prediction difficulty. 176
A.6 Comparison of Transformer LM's accuracy in two agreement tasks using *top3* evaluation metric and target verb evaluation metric (§4.2.3). For a fair comparison, sentences were excluded where the top ten predicted words do not include any verbs, which account for 7.9% of sentences in S-V agreement and 0.3% in O-PP agreement. 177

A.7 Comparison of LSTM LM's accuracy in two agreement tasks using *top3* evaluation metric and target verb evaluation metric (§4.2.3). For a fair comparison, sentences were excluded where the top ten predicted words do not include any verbs, which account for 29.8% of sentences in S-V agreement and 45.7% in O-PP agreement. 178
A.8 Inter-agreement (%) between the target verb evaluation metric and the *top3* evaluation metric. 178
A.9 Average causal effect of interventions on Transformer's NA task performance, quantified by **drop** in accuracy before and after different interventions, and further broken down based on prediction difficulty measured by the number of heuristics. The term *cue* here refers to the antecedent and its modifiers (determiners and adjectives) in O-PP agreement, and to the subject and its modifiers in S-V agreement. 179
A.10 Autoregressive Transformer LM's accuracy on two NA tasks with and without positional embeddings. 179
A.11 Masked Transformer LM's accuracy on two NA tasks with and without positional embeddings. 180
A.12 Mean accuracy (%) using exact-match is shown in gray, accuracy using reformatted exact-match described in Section 5.4 is shown in black. AM-Parser's graph-based output yields identical scores for both metrics hence only a single column is reported. 182
A.13 Performance of RC modification generalization broken down by construction. 182
A.14 Mean accuracy (%) on SLOG using the variable-free logical form of Qiu et al. (2022a). 184

List of abbreviations

NLP natural language processing

TAL traitement automatique des langues

NNs neural networks

LMs language models

NLMs neural language models

LSTM Long Short-Term Memory

BERT Bidirectional Encoder Representations from Transformers

GPT Generative Pre-trained Transformers

RC relative clauses

NP noun phrases

VP verb phrases

PoS part of speech

FFNLM Feed-forward neural language model

RNN Recurrent neural network

Seq2Seq sequence-to-sequence

NA number agreement

PCFG probabilistic Context-Free grammar

SCFG Synchronous Context-Free Grammar

S-V agreement subject-verb agreement

O-PP agreement object-past participle agreement

ACE average causal effect

LF logical form

MR meaning representation

PP Prepositional phrase

CP complementizer phrase

Chapter 1

Introduction

Human beings possess a remarkable capacity for abstraction and generalization, especially evident in our linguistic competence. Traditional linguistic theories suggest that human linguistic competence is rooted in complex built-in structures and symbolic processing (Chomsky, 1965, 1986; Pinker and Prince, 1988). These theories emphasize that language has a hierarchical structure, where larger linguistic structures are recursively constructed from smaller components. This recursive construction enables us to generate a large, potentially infinite number of sentences from a limited set of input elements (Hauser et al., 2002). The principle of compositionality bridges the syntactic structures to semantic understanding. It posits that the meaning of a sentence depends on the meanings of its parts and how they are syntactically combined (Frege, 1948; Partee, 1984), underpinning the immense productivity of language and our ability to understand numerous sentences, even those never encountered before.

Recent artificial neural networks (NNs) have achieved human-comparable performance in many natural language processing (NLP) tasks, ranging from machine translation to reading comprehension (Bubeck et al., 2023). Impressively, Transformer-based language models can generate apparently coherent and human-like grammatical text, appearing to possess an effective grasp of linguistic structures (Brown et al., 2020; OpenAI, 2023). Unlike traditional NLP models that rely on supervised learning and symbolic representations like parse trees or logical formulas (Jurafsky, 2000), the Transformer architecture does not explicitly model hierarchical structures and does not manipulate symbols. Instead, Transformers use matrix operations and non-linear transformations, processing information simultaneously. This allows them to perform tasks such as sentence completion by predicting what comes in the next slot given the prefix of a sentence (Elman, 1990; Vaswani et al., 2017). During this process, they learn to encode words and sentences into vectors directly from raw text without any grammatical guidance. These vector-based representations of

language, known as word or sentence embeddings, are now central to most NLP tasks and have demonstrated remarkable efficacy (Devlin et al., 2019; González-Carvajal and Garrido-Merchán, 2020; Choi et al., 2021).

Essentially, a Transformer language model is a model that can predict words from a given set of context words. This approach starkly differs from the traditional view of human linguistic competence, believed to be based on innate structural properties and symbolic-based processing. How can these models, which do not inherently embody hierarchical structures or execute symbolic operations, achieve such proficiency? The complexity of these "black-box" models, with millions or even trillions of parameters, makes it challenging to understand their inner workings. Their empirical success not only challenges these long-standing linguistic assumptions for language processing, but also prompts questions about the mechanisms underlying models' linguistic proficiency. One key question, which is also the focus of this thesis, is whether Transformers implicitly construct a form of abstract hierarchical representation.

1.1 Research question and objectives

The core research question we pose in this context is: **How do neural language models (NLMs) represent syntactic structures: do they essentially leverage surface-level patterns to mimic human language, or do they also implicitly learn abstract rules?**

This question engages with a longstanding debate in computational linguistics about the essential role of hierarchical, rule-based structures in language processing. While Transformers have excelled in various NLP tasks, theoretical studies suggest that they are limited in their ability to represent context-free grammars that capture hierarchical structures (Bhattamishra et al., 2020; Hahn, 2020). The empirical evidence, however, offers a more complex view. On the one hand, evidence suggests that Transformers can capture complex syntactic structures (Goldberg, 2019; Wolf, 2019), even mimic tree-like structures (Jawahar et al., 2019) and acquire structural biases from raw linguistic data (Warstadt and Bowman, 2020). On the other hand, some studies propose that these models may navigate structure-sensitive tasks by leveraging statistical regularities or heuristics in the training data (Wei et al., 2021; Sinha et al., 2021; McCoy et al., 2019; Da Costa and Chaves, 2020). Additionally, the capacity for extensive memorization in these models is well recognized (Halevy et al., 2009; Zhang et al., 2021), leading some to describe Transformer language models (LMs) as "stochastic parrots" that primarily memorize and shallowly recombine observed examples (Bender et al., 2021).

Given this divergence in existing research, coupled with the inherently opaque nature of

Transformer models, it remains an open question whether they are capable of constructing abstract hierarchical representations and achieving a level of human-like syntactic generalization. The crux of the inquiry lies in understanding how these models arrive at their predictions. Do they abstract complex rules akin to human-like syntactic understanding, or do they exploit statistical regularities that just happen to align with human language structure? The distinction is subtle, but it has profound implications for both our understanding of these models and their practical applications. Our research aims to characterize the model's syntactic abstraction capacity by disentangling these potential strategies, providing nuanced insights into the underlying mechanisms. The study serves two main objectives: i) assessing the potential of an autoregressive Transformer language model as an explanatory tool for human syntactic processing; ii) improving the model's interpretability.

1.1.1 Feasibility as explanatory model for human language processing

From both linguistic and cognitive science standpoints, studying the abstraction capabilities of neural language models is essential to understand how these models mirror or diverge from human linguistic cognition. It offers insights into whether these architectures can be considered analogs to human cognitive processes or just powerful pattern recognizers. Inspired by the research of Hupkes (2020), which confirmed the utility of recurrent neural language models as explanatory tools for human linguistic processing, we extend their foundational concepts to explore Transformer-based models, the current state-of-the-art. The first objective of this dissertation is to assess whether Transformer-based language models can provide a feasible computational framework for human syntactic processes.

Research has long probed the functional architecture of language in the brain through diverse techniques ranging from neuroimaging to behavioral experiments and computational modeling. Modern tools such as EEG, with high temporal but low spatial resolution, and fMRI, with high spatial but low temporal resolution, present challenges in studying the neural dynamics of linguistic behavior and cognition. Behavioral experiments, often involving small sets of artificial stimuli designed to be manageable for participants, may not capture the full complexity and variability of natural language. Furthermore, human subjects bring variability in terms of attention, fatigue, and motivation, which can introduce confounds into the experimental results. Both methodologies primarily offer a correlational view of brain activity, and more direct interventions, like neurosurgery or electrical stimulation, raise serious ethical concerns and may not be feasible in many cases, limiting causal inferences.

In this context, neural networks present a compelling alternative for studying language

processing, addressing many limitations inherent in traditional approaches. They enable large-scale experiments that can encompass the vast complexity and variability of natural language. Unlike human participants, models are free from confounds like fatigue and attention lapses, ensuring a controlled environment and consistent results. Crucially, these models allow for experimental manipulations to deduce causality without any ethical issues.

The idea of using neural networks for modeling human language processing traces its roots back to the connectionist models of the 1980s (Rumelhart et al., 1986). Early models were often trained on limited datasets, covering only a narrow spectrum of linguistic phenomena (Elman, 1991). Additionally, their architectures were generally simple and lacked computational resources to process large datasets or execute intricate language tasks. These constraints led to skepticism about their capability to capture the full complexity of human language, from syntax to semantics and beyond (Pinker and Prince, 1988; Fodor and Pylyshyn, 1988; Marcus, 1998). In contrast, the emergence of Transformer-based language models has redefined the field. Rather than questioning if these models can capture human-like linguistic behavior, the focus has shifted to understanding how they achieve it. Their remarkable performance in a variety of NLP tasks underscores their potential as explanatory tools for human language processing.

To effectively serve as an explanatory framework for human language processing, a computational model must meet certain foundational criteria. These criteria, which we outline below, guide the model selection and experimental design of this dissertation.

Our research focuses on the **autoregressive Transformer language model** (Vaswani et al., 2017).[1] This choice is motivated by the model's language modeling objective (§2.2), which aligns closely with the incremental word prediction characteristic of human language processing (Hale, 2001; Lappin and Bernardy, 2022). Such autoregressive language models are trained on vast amounts of text data, optimizing their weights to predict the next token based on the preceding tokens in a sequence. During language acquisition, humans are exposed to a large amount of data, and incremental word prediction plays a critical role in human language processing (Landauer and Dumais, 1997; Hale, 2001; Kuperberg and Jaeger, 2016; Levy, 2008) and more generally in cognition (Bar, 2007; Clark, 2015). We believe that a model intended to shed light on human language processing should align with this critical aspect of language acquisition and usage.[2]

[1] Autoregressive language model is also referred to as causal language model, we consistently use "autoregressive" in this dissertation.

[2] Incremental prediction is of course not the only task that humans undertake during language acquisition, designing and integrating more human-like learning tasks into computational models remains an active and open area of research.

Behavioral similarity For a model to offer insight into human linguistic processing of a particular phenomenon, it must first demonstrate the ability to replicate human usage of that phenomenon. Given the impressive human-like textual outputs of recent autoregressive LMs (Radford et al., 2019; Bubeck et al., 2023), it is clear that they possess a sophisticated capacity for general linguistic imitation. It remains to be seen whether they can capture more nuanced, structure-dependent phenomena. In this dissertation, our exploration centers on two pivotal facets of human language: hierarchical structure and compositionality. We aim to determine the extent to which Transformer-based models can behaviorally mimic these fundamental linguistic characteristics.

Representational adequacy While achieving behavioral imitation of syntactic phenomena is an essential prerequisite, it is not enough. For a model to shed light on human syntactic processing, it must capture the intricacies of linguistic structures, going beyond surface patterns or rote memorization, areas where they are known to excel (McCoy et al., 2019; Kodner and Gupta, 2020). In this dissertation, we further probe the internal representation of Transformer-based models to assess whether they encode a form of abstract hierarchical structure and if these structures align with well-established linguistic theories.

Controllability and interpretability Lastly, for a model to serve as an explanatory tool for a given linguistic phenomenon, it is essential that we have a certain understanding of its underlying workings in implementing this phenomenon. This understanding should allow us to exert a degree of control over the model's behavior. Symbolic models inherently offer this transparency due to their explicit symbolic representations and the deterministic nature of the rules governing them. However, neural networks, particularly those as complex as Transformers, operate as "black boxes". This obscurity has historically been a primary critique against neural networks as cognitive models (McCloskey, 1991).

In light of these challenges, we aim to explore and develop techniques that can penetrate this obscurity, revealing the inner mechanisms that drive specific linguistic behaviors in these models. In addition, we seek to employ interpretability tools, such as causal interventions, to identify and potentially influence the network components responsible for particular linguistic behaviors. While the vast number of parameters in these models makes complete control and interpretation challenging, our efforts seek to offer a degree of control that enables targeted manipulation and understanding of the models' decision-making processes.

1.1.2 Improving model interpretability

From a machine learning perspective, understanding what models know and how they know it, especially in relation to linguistic structures, is a prerequisite to improve these systems towards more interpretable and robust models. For instance, probing for the linguistic structure in models has been shown to be important in understanding their ability to adapt to new, unseen data (Marasovic, 2018). Delving into the syntactic abstraction capacity of these models offers a dual advantage: it provides insight into the nature of the representations they form and the factors driving their success, while also exposing their inherent limitations, which could help guide the creation of more effective architectures (Lake et al., 2017; Marcus, 2018).

Although Transformer-based models have achieved, and in some tasks surpassed, human-level performance in various fields (Otter et al., 2020; Lertvittayakumjorn and Toni, 2021; Khurana et al., 2023), they often exhibit fragility and their failures are distinctly unhuman-like (Firestone, 2020). For example, they can be derailed by minor input perturbations that humans easily handle (Firestone, 2020; Wang et al., 2022); models capable of generating human-like coherent text struggle with parsing moderately complex recursive structures (Yao and Koller, 2022). This raises concerns about the robustness and generalization capacity of Transformers models to less frequent patterns. Recognizing these challenges, researchers have sought to enhance neural networks using human linguistic biases. For example, integrating hierarchical structures as inductive bias into neural models has demonstrated improved efficiency in learning phenomena sensitive to structure (Kuncoro et al., 2018b; Wilcox et al., 2019; Qian et al., 2021). Similarly, integrating compositional structure biases can significantly boost a model's out-of-distribution compositional generalization (Qiu et al., 2022a). These linguistically motivated enhancements, driven by insights into model limitations, aim to enable faster and more robust learning (Lake et al., 2017; Besold et al., 2017), marking a promising research avenue.

With this backdrop, the second objective of this dissertation is to develop a linguistically-informed analysis framework. This framework combines and enhances current interpretability techniques, positioning challenge sets, probing classifiers, and causal analysis within an integrated epistemological structure. Its primary goal is to elucidate the underlying mechanisms that drive the linguistic behaviors of models, particularly in relation to syntactic structures. Additionally, it seeks to identify potential limitations in models' syntactic processing. This framework can serve as an analytic tool to measure progress and guide future improvements in these domains.

1.2 Contributions

This dissertation seeks to improve the understanding of Transformer-based models and evaluate the potential of autoregressive Transformer language models as explanatory models for human syntactic processing. Our exploration spans two abstraction levels within NLMs:

1.2.1 Assessing model capacity to represent syntactic structures

To better understand how Transformer language models represent hierarchical linguistic structures, we examine two superficially similar long-distance relationships — subject-verb and object-past participle agreements. Our contributions in this domain are multifaceted:

Heuristic-based evaluation protocol To distinguish between deep structural patterns and surface-level heuristics inherent in language, we introduce a novel heuristic-based evaluation protocol (§4.2.4). Our protocol uses a tiered evaluation system that emphasizes results from the most abstract cases. This protocol lays the groundwork for the development of model interpretation techniques in this dissertation and can also be instrumental in guiding linguistic experiments probing human syntactic capabilities.

Linguistically-informed analysis framework We introduced an integrated analysis framework that merges and expands upon recent interpretability techniques with a linguistic lens. At its heart, this framework examines two syntactic phenomena that, while outwardly similar, possess distinct linguistic modeling. We probe whether Transformer-based models can form distinct representations aligned with established linguistic analysis. Spanning behavioral assessments, representational probing, and functional analysis of inner mechanisms, our framework offers a comprehensive template for testing linguistic or cognitive hypotheses with computational models.

Dataset creation We created two challenge sets from naturalistic corpora: one for subject-verb agreement across relative clauses (27,582 sentences) and another for object-past participle agreement (68,794 sentences) in French. Instead of the common English-centric, template-generated methods in existing literature (Marvin and Linzen, 2018; Warstadt et al., 2019, ; i.a.), our sets emphasize the intricacies and diversity of natural language, ensuring ecological validity.

Pretrained model development We pre-trained three word-based French neural language models: an LSTM network, an autoregressive Transformer, and a masked Transformer LM, tailored for linguistic experiments. In contrast to the widespread use of sub-word-based models such as BERT (Devlin et al., 2019) or OpenAI's GPT2 (Radford et al., 2019), our

word-based models simplify linguistic experiments by avoiding sub-word intricacies.[3] Our pre-trained models can be readily used for other linguistic experiments.

1.2.2 Assessing model capacity to generalize compositionally observed structures

Our investigation not only evaluates NLMs' ability to represent syntactic structures but also delves into their capacity for compositional structural generalization. We examine whether Transformer-based models rely on syntactic generalization that aligns with human inductive biases to interpret new, unseen linguistic patterns effectively.

Many existing benchmarks, such as SCAN (Lake and Baroni, 2018) and COGS (Kim and Linzen, 2020), predominantly address *lexical generalization* — interpreting novel combinations of known lexical items and known linguistic structures, *structural generalization* tasks, where a model needs to combine known structures into a novel structure, are often underrepresented. To provide a more comprehensive perspective on the syntactic generalization capabilities of NLMs, we extend COGS to a compositional challenge set targeting structural generalization, covering syntactic elements like recursion and filler-gap dependencies.

Using our challenge set, we assess various Transformers models as well as a symbolic parser. Our findings underscore the specific limitations inherent to each architecture, highlighting areas of potential improvement. Furthermore, the findings provide a nuanced perspective on Transformers' abilities to make grammar-based generalizations. While these models can approximate compositional behavior to some degree, they don't seem to rely on the kind of syntactic generalization rooted in systematic compositional rules.

1.2.3 Publications

The research presented in this dissertation includes contributions that have been previously published and presented at conferences:

- Section 4.2 elaborates on an article published at EMNLP (Li et al., 2021).
- Section 4.3 builds on articles published at ACL (Li et al., 2022a) and TALN (Li et al., 2022b).
- Section 4.4 expands upon an article published in the journal TACL (Li et al., 2023b)

[3] Previous studies often used sub-word-based pretrained models, which necessitate constraints such as filtering evaluation sentences based on target-word appearances as single word pieces (Goldberg, 2019) or computing joint probabilities of sub-word sequences (Wolf, 2019).

- Section 4.5 presents entirely new, unpublished research.
- Chapter 5 is an expanded version of an article accepted by EMNLP (Li et al., 2023a).

For transparency and further research, all datasets we created, models we pretrained, and all associated code from this dissertation have been made publicly available in the repositories provided below:

- `https://gitlab.huma-num.fr/bli/syntactic-ability-nlm`
- `https://gitlab.huma-num.fr/bli/syntactic-info-distribution`
- `https://github.com/bingzhilee/contrastive_analysis`
- `https://github.com/bingzhilee/SLOG`

1.3 Outline

This dissertation consists of two main parts, each examining a different aspect of syntactic abstraction within neural language models. Chapter 4 assesses the extent to which the Transformer language model captures hierarchical structures, and Chapter 5 explores the capability of such models to compositionally generalize observed structures. Before the main parts, Chapter 2 provides essential background knowledge and highlights recent methodologies in the literature for analyzing the representation of linguistic structures in NLMs. Additionally, Chapter 3 offers a review of key studies that use long-distance agreement tasks as a tool for gauging NLM's syntactic capabilities.

Part 1 Chapter 4 introduces a three-tier epistemological framework for a contrastive study of Transformer LM's ability to represent syntactic structures. Our experiments specifically probe the model's capacity to represent long-distance relationships, indicative of hierarchical structure understanding. We use long-distance subject-verb agreement and object past-participle agreement in French as case studies and employ ecological training and evaluation data. This framework unfolds in a sequential manner, where each level of analysis builds upon the findings of the previous one.

- **Behavioral level** (Challenge sets): The foundational layer of our contrastive study is rooted in behavioral assessment. In section 4.2, we introduce a novel heuristic-based evaluation protocol to mitigate task-related confounds. Using this protocol, we assess the Transformer LM's syntactic awareness through number agreement tasks. The robust performance of the model indicates its ability to capture substantial syntactic information, moving beyond mere surface heuristics. This underscores its ability to meet the behavioral-level prerequisite for genuine syntactic generalization.

- **Representational level** (linguistic probes): Building upon the empirical findings, the next phase shifts to the model's internal representations. If the model excels in syntactic tests, it stands to reason that it encodes this syntactic information within its internal representations. The section 4.3 seeks to uncover where this syntactic information is encoded, moving from mere performance outcomes to the intricacies of internal encoding.
- **Functional level** (Causal intervention): The final level delves into causality. Simply detecting syntactic information in a model's representations does not guarantee that the model actively uses it. In section 4.4 and section 4.5, we use causal interventions to determine which components with relevant structural information actively influence the model's behavior during syntactic tasks.

Part 2 In Chapter 5, we develop a challenge test designed to evaluate compositional structural generalization. Using this test, we evaluate various Transformer-based models and a structure-informed parsing model. This test, crafted using a template-based synthetic data approach, operates within a semantic parsing framework. The models are tasked with translating English sentences into logic-based meaning representations. There is a systematic shift between training and evaluation sets: the constructions in the evaluation set differ structurally from the training set, but can be interpreted by rearranging components present within the training data. For example, in the training set, relative clauses (RC) modify noun phrases (NP) only in object positions, as in "Jimmy saw **the cat that the man held**". The test challenges models with RCs modifying NPs in the subject position, like "**The cat that Emma saw** ran". This study extends beyond the scope of Chapter 4, which focused on representing hierarchical structures. Instead, it examines the genuine syntactic generalization akin to what symbolic compositional rules would support.

Part I

Background

Chapter 2

Structure of language and neural language models

2.1 Structure in human language 13
2.2 Neural language models 15
2.2.1 Language modeling 15
2.2.2 Transformer-based neural language model 22
2.3 Analysis of linguistic structure in neural NLP models 28
2.3.1 Challenge sets 28
2.3.2 Probing classifiers 31
2.3.3 Causal intervention analysis 34

This chapter provides the background knowledge necessary for understanding the core discussions of this dissertation. Section 2.1 presents key linguistic assumptions regarding human language structures. Section 2.2 introduces neural language models, with a special focus on the autoregressive Transformer language model — the main investigation object of this thesis. Lastly, Section 2.3 offers a review of research methodologies that analyze the representation of linguistic structures in neural NLP models.

2.1 Structure in human language

Human language, in its essence, is a structured system that enables communication and the expression of thoughts and meaning, showcasing our unparalleled cognitive capabilities. Traditional linguistic theories, tracing back to seminal work by Chomsky and his peers, have long argued that the foundation of our linguistic abilities lies in the presence of intricate, innate structures within the human brain (Chomsky, 1965, 1986).

Central to these theories is the emphasis on the hierarchical nature of language. Rather than viewing sentences as mere strings of words, traditional linguistics posits that sentences have an underlying structure, where larger linguistic structures are recursively built from smaller components. This hierarchical arrangement allows for the nesting of linguistic elements within one another, leading to the creation of complex sentences and intricate meanings. According to this view, humans are born with a Universal Grammar — a set of fundamental principles that govern the structure of all human languages. The recursive nature of this symbolic system, using finite means (words and rules), enables the generation of an infinite number of expressions, showcasing the remarkable productivity of language (Chomsky, 1965; Hauser et al., 2002).

Linguistics often represents the hierarchical and recursive nature of language using discrete, symbolic representations like categorical labels and tree-like hierarchical structures (Berwick and Chomsky, 2016). Words are categorized based on their roles and functions in sentences, referred to as their grammatical or syntactic categories, commonly known as part of speech (PoS) tags in NLP (e.g., `noun`, `verb`). This hierarchical organization of words in sentences can be visualized as syntactic trees (Figure 2.1), where nodes represent syntactic categories (e.g., noun phrases (NP) or verb phrases (VP)), or individual words. Edges connect these nodes, with their left or right positioning indicating ordered relationships. Through this representation, the tree captures the hierarchical relationships between words and phrases.[1]

[1]Note that the provided syntactic tree is merely illustrative. In actual linguistic practice, the labels and structures vary according to the specific linguistic formalism being used.

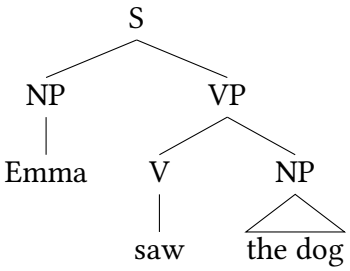


Figure 2.1: Syntactic tree: a hierarchical representation of word organization in sentences

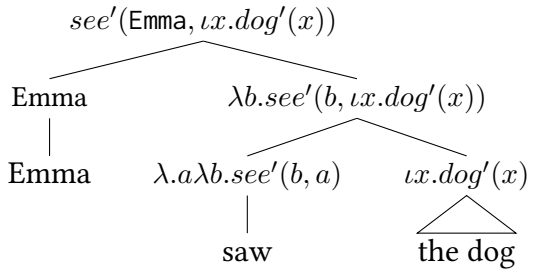


Figure 2.2: The sentence's meaning is derived compositionally from the meaning of its components, in line with its syntactic structure in Figure 2.1

Linguists largely agree that the principle of compositionality underpins the productivity of language, with a common focus on semantic compositionality. As articulated by Frege and later expanded by Partee, the principle of compositionality posits that the meaning of a linguistic expression is derived from the meanings of its individual components and their syntactic arrangement (Frege, 1948; Partee, 1984). This principle is crucial as it explains our ability to understand and produce a vast array of sentences, even those we have never previously encountered. Using tools from symbolic logic, formal semantics maps syntactic structures — how words are arranged — to semantic structures (their associated meanings). This approach allows us to clearly illustrate, as illustrated in Figure 2.2, how the meaning of a sentence is derived from the meanings of its constituent parts.

In essence, traditional linguistic theories advocate for a symbolic representation of language. They argue that our linguistic competence does not solely emerge from statistical learning or exposure to language data. Instead, it is rooted in complex, innate structures that guide our understanding and production of language (Chomsky, 1965; Hauser et al., 2002). This perspective has been foundational in shaping our understanding of human language and continues to influence linguistic research and debate.

Traditional NLP models have deep roots in linguistic theory. A typical NLP pipeline incorporates a range of intermediate linguistic modules, each designed to extract specific linguistic knowledge from human-annotated data, such as POS tagging, syntactic parsing, and semantic role labeling. These modules help transform textual information into structured,

symbolic forms that can be readily processed and analyzed. Before the rise of deep learning, these symbolic representations served a crucial role in NLP models. They were either directly incorporated into models as input data — obtained through linguistic feature engineering — or indirectly learned through models that extract linguistic patterns from corpora annotated by human linguists. Despite their interpretability benefits, these traditional models heavily depend on time-consuming feature engineering and extensive linguistic annotation, which limit their scalability and their capacity to generalize across different languages and domains. Furthermore, given the vast complexity and variability inherent to natural language, these models often struggle to effectively handle language processing based solely on small amounts of manually annotated data.

2.2 Neural language models

Neural language models (NLMs), central to the deep learning tsunami in natural language processing, fundamentally shift the paradigm from manual feature engineering to directly learning language representations from raw text data (Manning, 2015). In this section, we delve into a comprehensive overview of neural language models. Section 2.2.1 introduces the concept of language modeling — the task that enables NLMs to incorporate their understanding of different facets of language into their internal representations. Following this, Section 2.2.2 offers an in-depth look at the Transformer-based neural language models, in particular the autoregressive language model, a sub-type of NLMs and the main focus of this dissertation. In this subsection, we outline the Transformer architecture and describe the mathematics and intuitions behind its various components, with a particular focus on the self-attention mechanism.

2.2.1 Language modeling

For a language denoted as $\mathcal{L}$, a language model is a probability distribution over $\mathcal{L}$ such that the sum of probabilities for all possible sentences equals 1. This probability distribution estimates the likelihood of words or word sequences appearing in $\mathcal{L}$. For example, it could predict that “The cat on the mat meowed” is much more likely to appear in a text than “The cat on the mat shouted”.

A language model computes the probability of a sequence as the joint probability of each word, which can be decomposed into the product of conditional probabilities using the chain rule of probability:

$$\begin{aligned}\mathbb{P}(s) = \mathbb{P}(\text{The cat on the mat meowed}) = \mathbb{P}(\text{The}) \cdot \mathbb{P}(\text{cat} \mid \text{The}) \cdot \mathbb{P}(\text{on} \mid \text{The cat}) \cdot \\ \mathbb{P}(\text{the} \mid \text{The cat on}) \cdot \mathbb{P}(\text{mat} \mid \text{The cat on the}) \cdot \\ \mathbb{P}(\text{meowed} \mid \text{The cat on the mat})\end{aligned}$$

where $\mathbb{P}(\text{meowed} \mid \text{The cat on the mat})$ represents the conditional probability of the word "meowed" given the context of the previous words "The cat on the mat". More generally, for a sequence of words $s = w_1, w_2, ..., w_t$, its probability is calculated as:

$$\begin{aligned}\mathbb{P}(s) = \mathbb{P}(w_1, w_2, ..., w_t) &= \mathbb{P}(w_1) \cdot \mathbb{P}(w_2|w_1) \cdot \mathbb{P}(w_3|w_1, w_2) \cdot ... \cdot \mathbb{P}(w_t|w_1, ..., w_{t-1}) \\ &= \prod_{i=1}^{t} \mathbb{P}(w_i|w_{1:i-1}) \end{aligned} \tag{2.1}$$

In computational linguistics, this conditional probability $\mathbb{P}(w_i|w_1 \dots w_{i-1})$ is particularly useful as it allows to predict the next word w_i based on its preceding context $w_1 \dots w_{i-1}$ in the sentence.

Language modeling involves training language models on large text corpora to approximate the probability distribution of sequences in a language, using the next word prediction task. Through this training process, the model learns statistical patterns and word relationships, enabling it to predict word sequences effectively. This capacity is crucial for various NLP tasks. For example, in text generation, a chatbot can produce contextually relevant and coherent responses by predicting the most probable next words based on the given input.

N-gram language models Traditional language models estimate probabilities by counting word and word sequence frequencies in a training corpus. Predicting the next word w_i based on its entire preceding context $w_1, ..., w_{i-1}$ is challenging due to the exponential growth of possible word combinations. Long sequences suffer from the curse of dimensionality, leading to sparse representations. To address this, early models like n-grams introduce the Markov assumption. It simplifies the task by assuming that the probability of a word depends only on the last $n-1$ words, instead of the entire context. For instance, in a bigram model (where $n = 2$), w_i is assumed to depend only on the previous word w_{i-1}. So, instead of computing $\mathbb{P}(\text{meowed} \mid \text{The cat on the mat})$, a bigram model approximates it as $\mathbb{P}(\text{meowed} \mid \text{mat})$. This estimated probability can then be computed as the count of the bigram $C(w_{i-1}w_i)$ over the sum of all the bigrams starting with w_{i-1} in the training data:

$$\mathbb{P}(w_i|w_1, ..., w_{i-1}) \approx \mathbb{P}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{\sum_w C(w_{i-1}w)} \tag{2.2}$$

For the general case of an n-gram language model, the estimated probability of the entire S

is given by:

$$\mathbb{P}(S) = \mathbb{P}(w_1, w_2, ..., w_t) \approx \prod_{i=1}^{t} \mathbb{P}(w_i | w_{i-(n-1)}, ..., w_{i-1}) \tag{2.3}$$

N-gram models marked a shift from rule-based to statistical methods, foundational for early speech recognition and machine translation (Jelinek, 1998; Brown et al., 1993). However, they exhibit significant limitations. Most prominently, their parameters increase exponentially with n-gram order and face data sparsity challenges. They cannot handle unseen n-grams and struggle with long-distance word dependencies due to their fixed context window.

The limitations of the N-gram model prompted a shift to neural language models, which use artificial neural networks (i.e., learnable functions) for language modeling. Instead of relying solely on word counts, they learn continuous representations, which are high-dimensional vectors (known as word embeddings), for words and their contexts. This allows them to generalize to unseen word sequences. Moreover, they allow for the processing of longer sequences, thus enabling them to consider larger context windows.

Artificial neural networks (NN) are computational models originally inspired by biological neural networks (McCulloch and Pitts, 1943; Rosenblatt, 1958). As shown in Figure 2.3, the basic building block of an NN is the artificial neuron, referred to as a "node" or "unit". These neurons are grouped into layers: the input layer receives data features, the hidden layers learn patterns, and the output layer produces predictions. Neurons in one layer connect to those in the next through weighted connections. Each neuron processes its input by applying a weighted sum followed by a nonlinear activation function. It then sends this processed output to the subsequent neurons. During training, the weights are adjusted using an error backpropagation algorithm to minimize the difference between the network's output and the actual targets (Rumelhart et al., 1985). Mathematically, a neuron's operation is:

$$y = f\left(\sum_i w_i x_i + b\right) \tag{2.4}$$

where x_i represents inputs, w_i are weights, b is a bias term, and f is a non-linear activation function like the sigmoid or relu. The power of NNs comes from their ability to approximate almost any function given enough neurons and layers.

Feed-forward neural language model (FFNLM) The FFNLM applies a feed-forward neural network, a subtype of NN with unidirectional information flow (i.e., without looping back or cycles), for language modeling (Bengio et al., 2003). Similar to n-gram models, it employs a Markov assumption by considering a fixed number of previous words as context

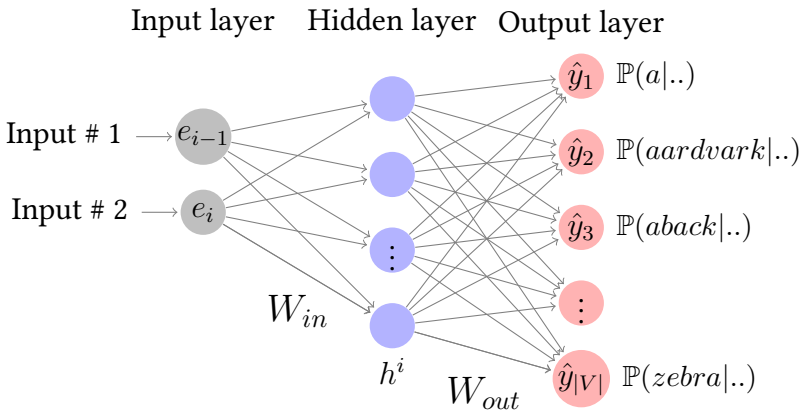


Figure 2.3: Forward inference in a feed-forward neural language model with window size of two, at timestep $i+1$. To predict the next word w_{i+1}, the model concatenates embeddings of the two preceding words, $\mathbf{e}_i$ and $\mathbf{e}_{i-1}$, multiplies them by $\mathbf{W}_{in}$, and applies an activation function to produce the hidden layer. This layer is then transformed by $\mathbf{W}_{out}$ and a softmax to estimate the probability of each word in its vocabulary being the next word w_{i+1}.

to predict the next word. In this model, words are first transformed into word embeddings. The embeddings from preceding k words are then fed into a feed-forward neural network, which in turn produces a probability distribution over possible next words. For instance, Figure 2.3 illustrates how to predict the next word w_{i+i} in an FFNLM with a context window size of $k = 2$. Formally, the defining equations at timestep $i+1$ are:

$$P(w_{i+1}|w_{i-1}w_i) = \textsc{softmax}(\mathbf{W}_{out}\mathbf{h}^i + \mathbf{b}_{out})$$
$$\mathbf{h}^i = g(\mathbf{W}_{in}\begin{bmatrix}\mathbf{e}_i \\ \mathbf{e}_{i-1}\end{bmatrix} + \mathbf{b}_{in}) \tag{2.5}$$

where $W_{in} \in \mathbb{R}^{d_h \times 2d}$, $W_{out} \in \mathbb{R}^{|V| \times d_h}$, with d_h denoting the hidden layer size, d the embedding size, $|V|$ the vocabulary size and g refers to the activation function.

However, despite being an advance over n-gram models, FFNLMs still have a fixed context size, which restricts their ability to capture long-range dependencies in text.

RNN language model Recurrent neural network (RNN) language models (Elman, 1990; Mikolov et al., 2010) overcome the limitations of the Markov assumption through recurrence. Unlike the feed-forward model, an RNN iteratively updates its hidden layers to capture information about the previous steps in the sequence. It processes the text one element at a time, predicting the next word based on the current word and the previous hidden state. Theoretically, this allows the model to retain information from the sentence's start to the present word, eliminating fixed context size constraints. As shown in Figure 2.4, while the

forward inference of RNN is very similar to that of feed-forward NLM, the key distinction lies in the RNN's capacity to maintain and use a memory of prior timesteps through its hidden states. Formally, the defining equations at timestep $i+1$ are:

$$\begin{aligned} \mathbb{P}(w_{i+1}|w_1 \dots w_i) &= \textsc{softmax}(\mathbf{W}_{out}\mathbf{h}^i + \mathbf{b}_{out}) \\ \mathbf{h}^i &= g(\mathbf{W}_{in}\mathbf{e}_i + \mathbf{b}_{in} + \mathbf{W}_{rec}\mathbf{h}^{i-1}) \end{aligned} \tag{2.6}$$

Here, $\mathbf{h}^i$ is an update of $\mathbf{h}^{i-1}$, integrating the information from the current word, $\mathbf{e}_i$. Essentially, $\mathbf{h}^i$ can be considered as a vector encoding the information from the starting word e_1 up to the current word e_i.

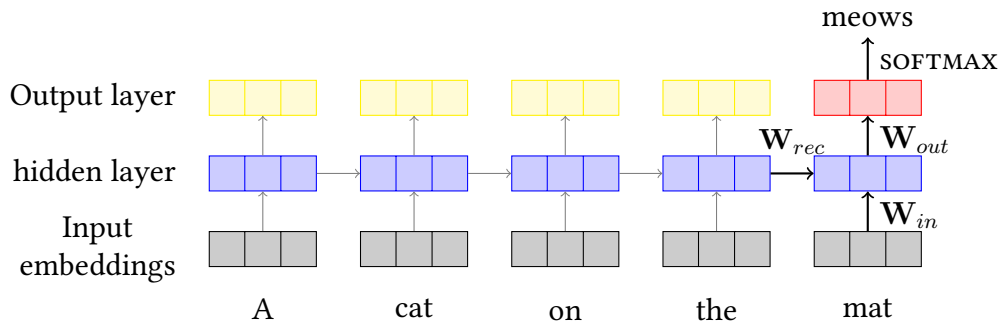


Figure 2.4: Forward inference in an RNN language model at timestep $i+1$. To predict the next word after the context "A cat on the mat", the model takes the embedding of the current word 'mat' and multiplies it by W_{in}. Concurrently, it multiplies the hidden layer of the previous timestep h_{i-1} by W_{rec}. These values are summed and passed through an activation function to produce the current hidden layer, h^i, which is then transformed by $\mathbf{W}_{out}$ and a softmax to produce a probability distribution over the vocabulary.

RNNs, with their recurrent nature, can capture longer dependencies but face challenges with extended sequences due to the vanishing or exploding gradient issue. The Long Short-Term Memory (LSTM) network, an advanced RNN variant, tackles the gradient problems through its sophisticated gated architecture. This structure helps the model decide when to keep or discard information across longer sequences, enhancing its ability to handle long-distance dependencies. Even with these improvements, LSTM still struggles with very long sequences.

Transformer-based language models Introduced by (Vaswani et al., 2017), the Transformer architecture has revolutionized NLP. At the heart of the Transformer architecture is the self-attention mechanism, detailed in Subsection 2.2.2, which theoretically allows each word to relate directly with every other word, irrespective of distance. The original model consists of an encoder-decoder structure, where the encoder is trained to convert input sequences into contextualized representations, while the decoder generates task-specific

output sequences, using the previous output for context at each step. This design, often referred to as sequence-to-sequence (Seq2Seq), is intended for Seq2Seq tasks, such as machine translation and text summarization.

Due to its parallel processing capability and the ability to capture long-range dependencies effectively, the Transformer architecture has been adapted for language modeling (Radford et al., 2018, 2019; Devlin et al., 2019) and has become the foundation for modern NLMs. Language modeling in Transformer models can be broadly categorized into two types: autoregressive and masked.

Autoregressive Transformer language models, such as the Generative Pre-trained Transformers (GPT) series (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2023) and more recent LLaMa (Touvron et al., 2023), predict the next word in a sequence based on all preceding words. They typically use only the decoder component of the original Transformer and are widely applied in natural language generation tasks. The main project of this dissertation (§4) focuses on a GPT2-like autoregressive Transformer LM, and we test LLaMa in our SLOG project (§5).

In contrast, masked Transformer language models such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), FlauBERT (Le et al., 2019), generate token representations by considering both prior and subsequent tokens in a sequence. They are pre-trained using a **masked language modeling objective**,[2] where a certain percentage of input tokens are masked and the model is trained to predict these masked tokens from their surrounding context. These models are typically built using the encoder layers of the original Transformer. While they don't directly model the probability of input sequences as autoregressive models do, making them less suited for generation tasks, they excel at generating contextually rich token representations, widely used in natural language understanding tasks.

Blending the strengths of both approaches, Seq2Seq language models like T5 (Raffel et al., 2020) frame all NLP tasks as text-to-text problems. During pre-training, various spans of text are masked (similar to BERT), and then the model is trained to predict full sequences autoregressively. This dual nature makes them versatile and suitable for a wide range of NLP tasks, from text generation to classification and translation. We test the structural generalization ability of T5 in Chapter 5.

Transformer-based pre-trained language models have become foundational elements in modern NLP systems due to their ability to learn generic transferable linguistic representations from vast unlabeled text corpora (Kalyan et al., 2021). Such contextualized

[2]Also called "denoising" objectives (Taylor, 1953)

representations can be used directly as inputs for task-specific text processing models. The transfer-learning paradigm, which includes pre-training followed by various fine-tuning methodologies (Devlin et al., 2019; Pruksachatkun et al., 2020; Liu et al., 2019b), further equips these models with task-specific knowledge. More recently, prompt tuning has emerged as an efficient technique that exploits these models to generate context-aware outputs from given prompts (Liu et al., 2021b). Over the past couple of years, Transformer-based language models offer groundbreaking improvements in NLP capabilities, models like GPT-4 can generate coherent human-like text (Bubeck et al., 2023). Moreover, they have inspired architectures in other domains, becoming a cornerstone in the expanding generative AI, which also includes image generation (Ramesh et al., 2022), and code generation (Chen et al., 2021).

On the theoretical front, these contextualized representations have also become a focus of research on language and human language processing (Baroni, 2020; Linzen and Baroni, 2021; Baroni, 2022; McCoy et al., 2018; Liu et al., 2019a; Yedetore et al., 2023, ; i.a.). Researchers have leveraged these representations to investigate how these models process and understand language, what kind of linguistic knowledge they implicitly learn, and how their understanding aligns with or diverges from our knowledge of human language processing. The insights gained from such investigations can help inform both the design of more effective computational models and the theoretical understanding of human language.

Evaluation of language models Perplexity is a probability-based metric to evaluate the quality of language model on its training task — predicting next word based on previous words in a sequence. While a language model's performance can always be assessed through improvements in downstream applications, such extrinsic evaluation can be computationally expensive and affected by various task-specific factors. Intrinsic evaluation like perplexity provides a direct and more efficient way to measure the potential improvement of LMs, aiding in model development and comparison.

Formally, for a language model trained on a corpus and tested on a held-out test set, $\mathcal{D}_t = w_1, w_2, ..., w_N$, the model's perplexity PPL on this test set is defined as the inverse probability of the test set, normalized by the number of words N:

$$\begin{aligned} \mathbb{P}(\mathcal{D}_t) &= P(w_1, w_2, ..., w_N)^{-\frac{1}{N}} \\ &= \left(\prod_{i=1}^{N} \frac{1}{P(w_i | w_1, \ldots, w_{i-1})} \right)^{\frac{1}{N}} \end{aligned} \tag{2.7}$$

In practice, working with log probabilities is often preferred as sequence probabilities can

become extremely small through multiplication. As language models are typically trained to maximize the log likelihood of the corpus, this definition of perplexity corresponds to exponentiating the average negative log-likelihood:

$$P(W) = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 P(w_i|w_1,\ldots,w_{i-1})} \tag{2.8}$$

Conceptually, perplexity can be viewed as the **weighted average branching factor** of a language. It represents the average number of plausible subsequent words that can follow any given word sequence. Thus, the perplexity of a language model on a test set is the average number of equally probable word predictions that the model makes for each actual word in the test set. A lower perplexity score indicates a better language model.

2.2.2 Transformer-based neural language model

In the previous section, we have seen that various neural networks can be used for language modeling; in this section, we delve into the Transformer, the current state of the art architecture for language modeling, with a specific focus on the autoregressive variant.

Overview of the Transformer architecture Transformer-based models consist of a series of identical Transformer blocks stacked together. As shown in Figure 2.5, a standard Transformer block has two main components: a self-attention layer, followed by a fully-connected feed-forward network. Each of these two sub-layers includes a residual connection and is followed by a layer normalization operation.

The self-attention layer, a key innovation of the Transformer architecture, enables the model to focus on different parts of the input when predicting the output for a particular position. This is accomplished by computing a weighted sum of the input sequence, where the weights are determined by the relevance of input elements related to the current position.

The feed-forward network (§2.2.1) applies a position-wise transformation to the attention outputs and allows the model to learn complex patterns. This transformation is the same at each position and includes two linear transformations with a relu activation function in between.

Residual connections play a crucial role in information preservation. These connections allow information from earlier layers to be passed unaltered to later layers, addressing issues such as vanishing gradients (He et al., 2016). Specifically, in Transformer blocks, the output of each sub-layer (self-attention and feed-forward network) is added to its input before being normalized. Layer normalization (Ba et al., 2016) is applied to these summed vectors to

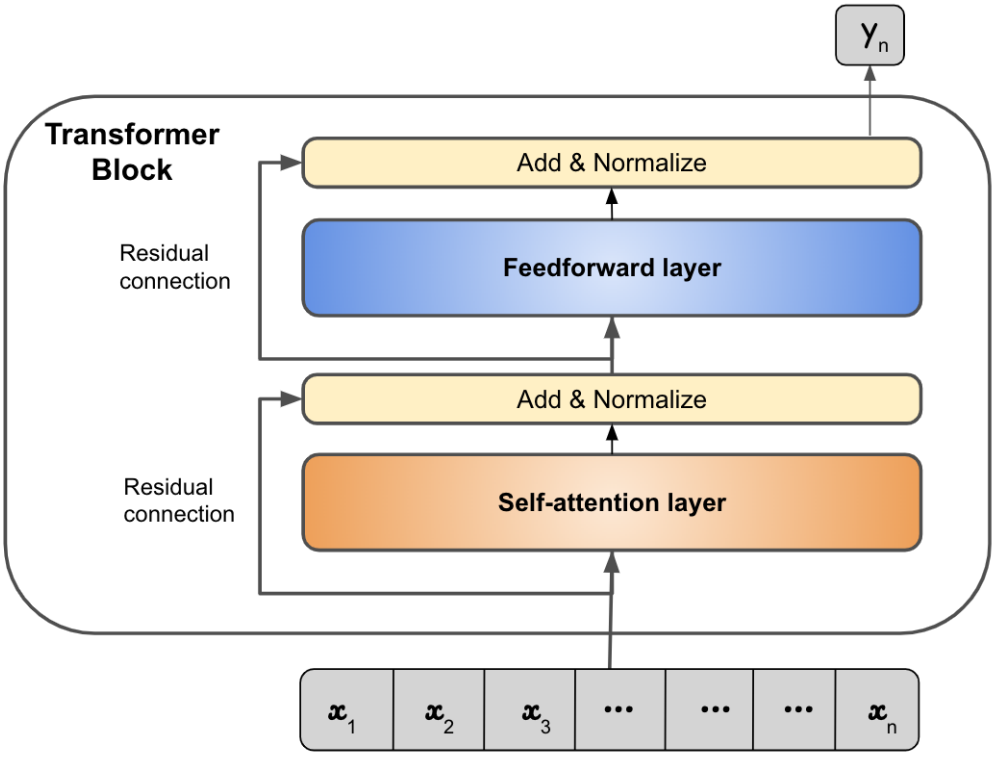


Figure 2.5: Components of a typical Transformer block: self-attention, feed-forward network, layer normalization, and residual connections.

maintain the hidden layer values within an optimal range for gradient-based training. This process involves transforming the inputs to achieve a mean of 0 and a standard deviation of 1 across each layer.

The original Transformer consists of both encoder and decoder stacks, with each stack containing six identical Transformer layers. Early adaptations of Transformer-based models typically stack $l \in \{6, 10, 12, 16, 24\}$ such layers, while the rise of large language models such as GPT-3 has led to architectures with nearly a hundred layers or more (Brown et al., 2020; OpenAI, 2023). In addition to these Transformer blocks, the model includes an input and output embedding layer. The input embedding layer converts each word in the input sequence to a high-dimensional vector, which is then combined with its position encoding to incorporate word order within the sequence. The output layer, on the other hand, is a linear layer followed by a softmax function to produce a probability distribution over the vocabulary for the next word prediction.

A standard formulation of the full Transformer stack is as follows:

$$
\begin{aligned}
&\text{For each layer } l = 1 \text{ to } L: \\
&\qquad \text{Self-Attention: } \mathbf{Z}_l^{(1)} = \text{SelfAttn}(\mathbf{Z}_{l-1}) + \mathbf{Z}_{l-1} \\
&\qquad \text{Normalization: } \mathbf{Z}_l^{(2)} = \text{LayerNorm}(\mathbf{Z}_l^{(1)}) \\
&\qquad \text{Feed Forward: } \mathbf{Z}_l^{(3)} = \text{FFN}(\mathbf{Z}_l^{(2)}) + \mathbf{Z}_l^{(2)} \\
&\qquad \text{Normalization: } \mathbf{Z}_l = \text{LayerNorm}(\mathbf{Z}_l^{(3)})
\end{aligned} \tag{2.9}
$$

where l is the layer number, L is the total number of layers, $\mathbf{Z}_l^{(1)}$ is the output of the self-attention mechanism at layer l, $\mathbf{Z}_l^{(2)}$ is the normalized self-attention output, $\mathbf{Z}_l^{(3)}$ is the output of the feed-forward network at layer l, and $\mathbf{Z}_l$ is the final output of layer l after normalization.

Self-attention mechanism The core principle behind the attention-based approach is its ability to assess the relevance of different elements within a sequence in relation to a target element. Let's consider an example sentence, "The cat on the mat ate a fish". As shown in Figure 2.6, while predicting the next word after "ate", self-attention draws comparisons between the current word "ate" and all preceding words, including itself. Each pair of words is then assigned a relevance score, which can be calculated as a dot product. To compute the final representation for "ate", the mechanism takes each word's vector representation seen so far, weighs it by the corresponding relevance score, and sums these weighted representations. As a result, words that are more relevant to "ate" will contribute more to its final representation.

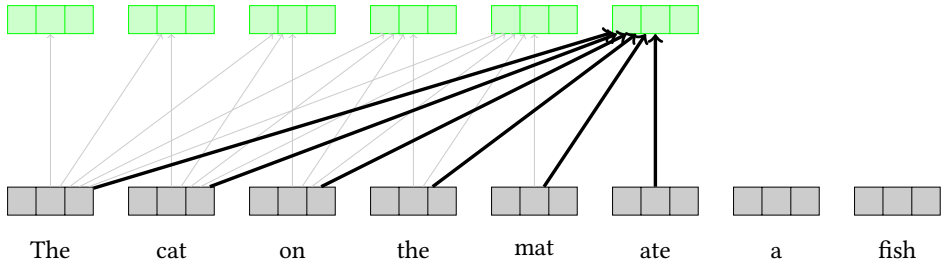


Figure 2.6: Masked self-attention in autoregressive Transformer language models: each token is processed considering all the preceding tokens and itself, future tokens are excluded.

Transformers introduce a more sophisticated way of representing how each word contributes to the understanding of other words within a sequence. The attention process discerns three distinct roles: the **query** (Q), the **key** (K), and the **value** (V). The query corresponds to the current focus of attention, the key represents the preceding input item being compared to the attention focus, and the value is employed to compute the output for

the current position. To capture these three roles, the self-attention mechanism uses three weight matrices $\mathbf{W}^q$, $\mathbf{W}^k$, $\mathbf{W}^v$, which are learned during the training. They transform each input vector to represent its specific role as a query, key, or value. Given these projections, the score between a current word x_i, and a token in the preceding context, x_j, is computed as the dot product of their respective **query** and **key** vectors — $q_i \cdot k_j$. To achieve more stable gradients, this score is normalized by dividing it by the square root of the key vectors' dimension. Given a query and the set of keys, $\{ k_1, k_2, ..., k_i \}$, these individual scores are then passed through a softmax function to obtain the attention distribution:

$$\alpha_1 \dots \alpha_i = \textsc{softmax}\left(\frac{q_i \cdot k_1}{\sqrt{d_k}}, \dots, \frac{q_i \cdot k_i}{\sqrt{d_k}}\right) \tag{2.10}$$

This attention distribution is then used to weigh the respective **value** vectors of the tokens. The result is a weighted sum of all the **value** vectors, which serves as the output of the self-attention mechanism for the token under consideration. This process is formally expressed as:

$$y_i = \sum_{j \leq i} \alpha_j \cdot V_j \tag{2.11}$$

Since each output y_i is computed independently, this entire attention process can be parallelized using matrix multiplication by considering all the N tokens of the input sequence as a single matrix $X \in \mathbb{R}^{N \times d}$. The entire self-attention process for a sequence of N tokens is computed as:

$$\text{SelfAttention}\left(\mathbf{Q}, \mathbf{K}, \mathbf{V}\right) = \textsc{softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{2.12}$$

Masked Transformer LMs directly use this self-attention computation. However, the autoregressive Transformer has to maintain the autoregressive property, where a token's prediction relies only on preceding tokens and not future ones. To achieve this, a causal attention mask is applied. The causal mask matrix is formally defined as:

$$\textsc{mask}_{ij} = \begin{cases} -\infty & \text{if } j > i \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

Here, the upper triangle (future positions related to the current token), is filled with negative infinity and the lower triangle has zeros. Incorporating this causal mask, the output of a

single self-attention layer becomes:

$$\text{SelfAttention}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\textsc{softmax}\left(\frac{\mathbf{Q}\cdot\mathbf{K}^{T}}{\sqrt{d_k}}+\textsc{mask}\right)\mathbf{V} \tag{2.14}$$

Here, adding a very large negative number (from the mask) to the future positions, ensures that when the softmax function is applied in the next step, these positions will have an attention score of nearly 0. This forces the self-attention mechanism to attend only to its previous words and itself, thereby preventing information flow from any future words. This causal attention mask, as suggested by Haviv et al. (2022) may implicitly introduce positional information into the self-attention layer.

Words within a sentence are interconnected in multiple ways. Consider the sentence "The cat on the mat ate a fish", the verb "ate" has a subject-verb syntactic dependency with "cat" and also shares a semantic relationship, where "cat" is the agent of the action. To capture these different aspects of the syntactic, semantic, and even discourse relationships simultaneously, the Transformer employs multi-head attention. Specifically, each of the h attention heads in a self-attention layer uses its unique learned set of weight matrices: $\mathbf{W}_h^K$,$\mathbf{W}_h^Q$ and $\mathbf{W}_h^V$, to determine the respective **query**, **key**, and **value** vectors. Consequently, the output of the multi-head layer with h heads consists of h distinct vectors, each representing a different facet of the token's contextual relationships. For instance, one head might focus on learning grammatical structures, while another might specialize in capturing thematic relationships. These head-specific outputs are then concatenated and linearly transformed via $\mathbf{W}^O$, to produce the final output for each token.

In mathematical terms, for an attention head i, the output, $\textsc{head}_i$, for a given sequence of N tokens, $X\in\mathbb{R}^{N\times d}$, is computed as follows:

$$\begin{aligned}\textsc{head}_i&=\text{SelfAttention}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)\\ \mathbf{Q}=\mathbf{X}\mathbf{W}_i^Q;\mathbf{K}&=\mathbf{X}\mathbf{W}_i^K;\mathbf{V}=\mathbf{X}\mathbf{W}_i^V\end{aligned} \tag{2.15}$$

where $\mathbf{W}_i^Q\in\mathbb{R}^{d\times d_k}$, $\mathbf{W}_i^K\in\mathbb{R}^{d\times d_k}$, $\mathbf{W}_i^V\in\mathbb{R}^{d\times d_v}$ with d denoting the dimensionality of both the input to and output from the model, d_k for the key and query embedding dimensions and d_v for the value embedding dimension. Outputs from each head are concatenated and linearly transformed, producing the final output of a multi-head attention layer:

$$\text{MultiHeadAttention}\left(X\right)=\left(\textsc{head}_1\oplus\textsc{head}_2,\ldots,\oplus\textsc{head}_h\right)\mathbf{W}^O \tag{2.16}$$

where $\mathbf{W}^O\in\mathbb{R}^{hd_v\times d}$, and h is the total number of attention heads, $\oplus$ denotes the concatenation operation.

Position embeddings Unlike RNN, which inherently handles word order information by processing input sequences one element at a time, the Transformer architecture is inherently agnostic to the order of tokens by considering all tokens in the input sequence simultaneously. However, the order of words is crucial to the semantics and syntax of a sequence and is often crucial in many tasks, such as language modeling and sequence-to-sequence translation. To overcome this limitation and inject some sense of position or order into the model, position embeddings were introduced.

One straightforward solution is to directly add positional embeddings to the input embeddings. Just as the model learns an embedding for a word like "cat", it can also learn a specific embedding for its position in a sequence such as "The cat on the mat", identifying it as the second word. In the original Transformer architecture, positional embeddings are generated using fixed sinusoidal functions. These functions convert integer positions into real-valued vectors, creating a unique positional embedding for each position. Specifically, each dimension of the positional embedding receives a value from a sine or cosine function of a different frequency. Formally, for position p and dimension i, the values are defined as:

$$\begin{aligned} PE_{(p,2i)} &= \sin\left(\frac{p}{10000^{\frac{2i}{d}}}\right) \\ PE_{(p,2i+1)} &= \cos\left(\frac{p}{10000^{\frac{2i}{d}}}\right) \end{aligned} \tag{2.17}$$

where d is the dimension of the embeddings. These fixed positional embeddings are then added to the standard word embeddings, giving the model a sense of each token's position in the sequence. This position encoding scheme has been extended to learned instead of fixed positional embedding in subsequent models such as BERT (Devlin et al., 2019), Reformer (Kitaev et al., 2020), RoBERTa (Liu et al., 2019c), etc.

While absolute position embeddings provide a sense of sequence order, they don't directly capture the relative distances between tokens. For instance, by modifying "the cat ate a fish" to "Yesterday, the cat ate a fish", the absolute positions change but not the core meaning. What matters for meaning is the relative position between "cat" and "fish", regardless of their absolute position in the sequence. To better capture such relational dynamics, relative position embeddings are introduced (Shaw et al., 2018; Dai et al., 2019). These embeddings shift the focus from the absolute position of a token in a sequence to the relative distances or positional differences between pairs of tokens.

Numerous subsequent models have proposed alternative position encoding schemes. For instance, some approaches integrate position information into the attention matrix instead of the input (Dai et al., 2019; Raffel et al., 2020). Others represent positions structurally based

on the distances on a sentence's parse tree representation (Wang et al., 2019; Shiv and Quirk, 2019). Improving position representations is an ongoing research focus. The study by Dufter et al. (2022) provides a comprehensive review of position encoding within the Transformer architecture.

2.3 Analysis of linguistic structure in neural NLP models

The study of linguistic structures in computational models has a long history, dating back to the work of Elman (1990, 1991) and Tabor (1994). Their pioneering research provided early evidence of the potential for neural networks to learn and embody abstract syntactic structures from non-annotated language data. Transitioning from these early insights to the modern era, the scale and complexity of current models like Transformers have significantly increased. As discussed in previous section, they generate output in the form of complex probability distributions over a large vocabulary of words or sub-words. This, in combination with their high-dimensional representation for inputs and millions of parameterized weights for operations, makes the interpretation of these models challenging.

In recent years, a myriad of analysis methods have been developed to better understand the inner mechanics of NLMs. Many studies suggest that these models have learned a substantial amount of syntactic knowledge that resembles human understanding, while others question the degree to which these models develop abstract structural representations of language. Although recent large language models demonstrate an apparently human-like ability to generate fluent and grammatically correct text (Bubeck et al., 2023), there is yet no consensus on whether these Transformer-based models truly understand and incorporate the linguistic structure.

In this section, we will explore three core methods for interpreting and analyzing the representation of linguistic structure in neural NLP models and also discuss their associated limitations.

2.3.1 Challenge sets

Challenge sets, also known as test suites, have a long-standing tradition in NLP, tracing back to work like Lehmann et al. (1996). These carefully curated sets include a wide range of linguistic phenomena, often targeting specific syntactic, semantic, or pragmatic properties (King and Falkedal, 1990; Sennrich, 2017; Isabelle et al., 2017; Naik et al., 2018, ; i.a.). While they were initially employed primarily for evaluating machine translation systems, the evolution and success of neural language models have broadened their application.

Largely inspired by the experimental paradigms in psycholinguistics, challenge sets have become one of the important methodologies for investigating the fine-grained linguistic knowledge embodied within NLMs. This approach attempts to answer questions like: How well do neural language models capture linguistic principles, and to what extent do they exhibit human-like grammatical competence?

In psycholinguistics literature, a paradigmatic test for human syntactic capacity comes from agreement phenomena (Bock and Cutting, 1992; Bock and Miller, 1991; Bock et al., 2001). For instance, subject-verb agreement in English as illustrated in (1): the form of the verb "are" is determined by its syntactic subject "keys", irrespective of the linear distance between them or the presence of the intervening noun, "cabinet", which carries a different grammatical number than the subject, and is often referred to as agreement *attractor* (Bock and Miller, 1991). Such long-distance agreement phenomenon exemplifies the hierarchical organization of language rather than a simple linear structure (Everaert et al., 2015).

(1) The old rusty **keys** to the cabinet **are** on the table.

Linzen et al. (2016) pioneered the use of subject-verb agreement to assess the syntactic sensitivity of modern NLM. They collected 1.35 million English sentences with present-tense verbs from an auto-parsed Wikipedia corpus and annotated each with the main verb's grammatical number. The model's syntactic ability was then evaluated through a number agreement (NA) prediction task. In this task, an LSTM took as input the sentence prefixes like the one in (2), and was trained to predict the grammatical number of the subsequent verb, either *Singular* or *Plural*.

(2) The old rusty keys to the cabinet __

Linzen and colleagues tested an LSTM with 50 hidden units and found that the model demonstrated near perfect overall accuracy on unseen sentence prefixes. Even in the most challenging cases with four attractors like (3)[3], the accuracy of the number prediction was still 82%.

(3) Yet the **ratio** of men who survive to the women and children who survive in these events **is** not clear.

From these results, the authors concluded that LSTM models, when provided with explicit supervision, can capture significant grammatical structures, enabling them to reasonably approximate structure-sensitive dependencies.

Building on this experimental approach, Gulordava et al. (2018) further showed that such long-distance agreement is learnable for an LSTM trained on language modeling (i.e.,

[3]Agreement attractors are highlighted with an underline. The subject and target verb are marked in bold.

without explicit supervision). Subsequent research has delved deeper into understanding the capability of NLMs to abstractly represent sentence structures during agreement resolution. This exploration spans various dimensions: different languages (Ravfogel et al., 2018; Gulordava et al., 2018; Lakretz et al., 2021b), diverse models (Bernardy and Lappin, 2017; Goldberg, 2019), and potential confounding factors such as lexical co-occurrences (Gulordava et al., 2018; Lasri et al., 2022a) or surface-level heuristics (Kuncoro et al., 2018a). Our research contributes to this body of work, with a focus on French agreement phenomena and autoregressive Transformer LM (Li et al., 2021). The majority of these studies converge on the positive finding that neural language models are capable of learning a considerable amount of non-trivial structure information from the (unannotated) training data. More detailed discussions on related work that approaches long-distance agreement tasks can be found in Chapter 3.

Another significant line of research has sought to expand this experimental approach beyond agreement phenomena to encompass a wider array of syntactic phenomena, such as anaphora, licensing, argument structure alternation, and filler-gap dependencies (Marvin and Linzen, 2018; Kann et al., 2019; Warstadt et al., 2019; Wilcox et al., 2018; Hu et al., 2020, ; i.a.). These studies typically create precisely drafted templates to generate challenge sets featuring specific linguistic phenomena, and then evaluate a neural network's grammaticality judgement on minimally differing sentence pairs based on grammaticality. Evaluations are conducted either through binary acceptability classification, similar to the number agreement prediction task proposed by Linzen et al. (2016), or by comparing the probabilities that a language model assigns to whole sentences. More recently, Warstadt et al. (2020) introduced BLiMP, a benchmark of linguistic minimal pairs covering a wide range of English grammatical phenomena. Generally, in these studies, NLMs' performance varies significantly across linguistic phenomena. While the models demonstrate robust knowledge of some syntactic phenomena, such as local subject-verb agreement, ellipsis, and control/raising, they struggle with more subtle semantic and complex syntactic phenomena, including licensing and extraction islands.

Formal languages Analyzing NLMs' ability to handle linguistic structures is complex due to the intertwining of syntactic, semantic, and statistical regularities in human languages. To precisely focus on syntax-processing, researchers also employ formal languages in challenge sets. Typically, a study using formal languages designs a formal grammar to generate a corpus of sentences. A language model (§2.2) is then trained on this corpus, and the evaluation focuses on the model's capability to recognize sequences from the training set and to generalize these learnings to unseen sequences.

Some studies focus on formal languages that correspond to specific classes in the Chomsky hierarchy, investigating which language classes can be theoretically or empirically learned by NLMs. Early studies have shown that certain regular (Giles et al., 1992) and context-free (Elman, 1991) languages can be learned by different RNN models. Subsequent research found that, with proper parametrization, LSTM networks could learn context-sensitive languages, such as $a^n b^n c^n$, and generalize to longer sequences (Gers and Schmidhuber, 2001; Weiss et al., 2018; Suzgun et al., 2019). In contrast, Transformer models have demonstrated, in theoretical studies, more limited capacities compared to LSTMs when handling regular languages and context-free languages (Bhattamishra et al., 2020; Hahn, 2020). However, empirical findings like (Ebrahimi et al., 2020) have shown that Transformers can learn $Dyck_k$ languages from finite samples, matching the performance of LSTMs.

Others craft formal grammars that mirror specific structures present in natural language. For instance, Lakretz et al. (2021a) used a probabilistic Context-Free grammar (PCFG) to investigate RNN's ability to handle recursively nested subject-verb agreements, Hupkes et al. (2020) used a set of PCFGs to assess NLMs' capacity in processing hierarchical compositional structure. Notably, Sennhauser and Berwick (2018) evaluated LSTMs using bracket prediction tasks as a measure of understanding linguistic hierarchical structures. While their findings confirmed that LSTMs can learn context-free grammar, they also observed that models' good performance stemmed more from efficiently handling nuisance variables rather than truly learning the underlying context-free rules. Hahn (2020) has theoretically demonstrated that Transformer-based models struggle with bracket closing and iterated negation tasks, both computations are considered to be essential to hierarchical structure.

Limitations Challenge sets shed light on models' fine-grained linguistic capabilities by assessing their responses to specific inputs. However, this approach offers limited insight into the internal representations that the model has learned. Confounding factors, such as the inability to distinguish genuine syntactic comprehension from superficial pattern recognition like frequency-based heuristics, can make their results hard to interpret. To get a more comprehensive picture of NLMs' syntactic abilities, these tests should be supplemented with other methods, such as probing tasks or interpretability techniques that can provide insights into the models' internal workings.

2.3.2 Probing classifiers

The probing classifier approach, also known as auxiliary prediction tasks (Adi et al., 2016), diagnostic classifiers (Veldhoen et al., 2016), or linguistic probes (Zhu and Rudzicz, 2020),

is widely used to analyze the linguistic capabilities of neural NLP models. At its core, this approach involves training a classifier — a "probe" — on a model's internal representation to predict specific linguistic properties. Success in this prediction indicates that the model has encoded the relevant linguistic features. The basic premise is that if a model captures a particular linguistic property, this information should be extractable from its internal representation (Hupkes et al., 2018). This approach thus seeks to address the question: What linguistic properties are encoded in a model's internal representations, and where are they located within the model?

Formally, we define a model under investigation as a function, NN : $x \rightsquigarrow r$, that generates a representation, r, for an input element. A probing dataset, denoted as $\mathcal{D} = \{r^{(i)}, z^{(i)}\}$, pairs each representation with its associated linguistic property. The probing classifier can then be defined as a function $\mathcal{C}$ that maps the model's representation to a linguistic property of interest:

$$\mathcal{C} : r \rightsquigarrow z \tag{2.18}$$

In an early application of this approach, Shi et al. (2016) probed the syntactic information in neural machine translation. They extracted the hidden states of an RNN encoder and used them to train a logistic regression classifier, predicting labels related to morpho-syntax, such as PoS tags, constituent labels (e.g., NP, VP), voice, and tense. Their results, showing high probing accuracy relative to baseline measures, led them to conclude that the RNN captures significant syntactic information at both the word and sentence levels. Furthermore, they used probing classifiers to identify where syntactic information was stored across layers, observing that local features were often encoded in lower layers, while more abstract, global information was found in upper layers.

This probing methodology has since expanded to investigate other syntactic facets in RNN models, such as surface sentence structure (Adi et al., 2016),[4] parse tree depth (Conneau et al., 2018), syntactic agreement (Giulianelli et al., 2018) and even semantic properties (Ettinger et al., 2016). This methodology has also been extensively applied to Transformer-based models (Tenney et al., 2019; Liu et al., 2019a; Jawahar et al., 2019; Klafka and Ettinger, 2020, ; i.a.). Collectively, these investigations have yielded promising results, consistently indicating that neural NLP models trained on vast data do encode a wide array of linguistic properties within their internal representations. An interesting extension of this methodology is the structural probe introduced by Hewitt and Manning (2019). This probe, distinct yet related to the probing classifier, identified a linear transformation that could extract syntactic parse tree structures from word representation spaces in models like ELMo and BERT, but not

[4]Surface sentence structure refers to sentence length, word identities and word order in Adi et al. (2016).

from simpler baseline representations. (See Table 2.1 for a categorization of representative work using the probing classifiers.)

Linguistic properties	Probing classifiers	Probed models	Baseline models	Papers
PoS, tense,voice, constituents	Logistic regression	LSTM encoder	Phrase/syntax-based system	Shi et al. (2016)
Surface sentence structure	MLP	LSTM encoder	CBOW	Adi et al. (2016)
Surface structure, parse tree depth, top constituents	MLP	BiLSTM, ConvNet	Unigram, Human	Conneau et al. (2018)
Syntactic agreement	Linear	LSTM LM	–	Giulianelli et al. (2018)
PoS, dependency edge	Linear &MLP	ELMo	Control tasks	Hewitt and Liang (2019)
8 core NLP labeling tasks	MLP	CoVe, ELMo, GPT, BERT	Lexical baselines, randomized ELMo, word-level CNN	Tenney et al. (2019)
Entire parse tree	Linear	ELMo, BERT	Non-contextual models	Hewitt and Manning (2019)

Table 2.1: A categorization of some representative studies using probing classifiers to investigate syntactic structures in NLMs, according to linguistic properties examined, classifier types, probed models, and baseline models.

On the other hand, recent studies also highlight potential pitfalls in the probing classifier approach, emphasizing that learned properties should be interpreted in comparison to control baselines. This can be achieved through techniques such as training probes on randomized representations (Conneau et al., 2018; Tenney et al., 2019), using control functions (Maudslay et al., 2020), or implementing control tasks (Hewitt and Liang, 2019; Ravichander et al., 2021). Specifically, control tasks are designed in a way that they can only be solved if the probe memorizes the task. Based on this, Hewitt and Liang (2019) introduced the concept of *selectivity*, which is defined as the performance gap between a probing task and its control counterpart. Using this metric to guide probe selection, they found that, while linear probes are highly selective, nonlinear probes are generally less so. The effectiveness of such probes with respect to its complexity remains a topic of discussion (Maudslay et al., 2020; Ravichander et al., 2021), Belinkov (2022) provides a comprehensive review on probing methods.

Limitations A main limitation of probing classifiers is that they only reveal correlations between linguistic properties and a network's inner representations, but do not necessarily indicate causality. Since these probes operate independently from the model's original task, they do not provide any insight on whether the information discovered by the probe influences the model's predictions. Only a few studies we have seen so far, like the one by Giulianelli et al. (2018), address this limitation; we will further explore and categorize such efforts in the following subsection, focusing on the causal analysis approach.

2.3.3 Causal intervention analysis

While linguistic probes are instrumental in revealing what linguistic properties might be encoded within neural models, they often fail to establish a causal relationship between these properties and the probed model's prediction. Causal intervention analysis fills this gap: it assesses the direct influence of specific model components on predictions by manipulating parts of the model and tracking resultant output changes. In this way, we can answer the causal question: which information is actually being used by neural models? Causal analysis is commonly paired with behavioral tests or probing tasks, providing a comprehensive framework for both uncovering and validating the model's linguistic behaviors.

Causal interventions in neural models vary based on where they are applied within the model. Broadly, these interventions can be grouped into three categories:

- Input-level interventions (Zmigrod et al., 2019; Vig et al., 2020; Amini et al., 2023)
- layer-level interventions (Giulianelli et al., 2018; Elazar et al., 2021; Vig et al., 2020; Ravfogel et al., 2021; Feder et al., 2021, ; i.a.)
- neuron unit-level interventions (Bau et al., 2018; Lakretz et al., 2019; Vig et al., 2020; Mueller et al., 2022)

In one of the pioneering works, Giulianelli et al. (2018) combined causal intervention with probing classifiers to explore an NLM's syntactic capabilities. They showed that by intervening on an NLM's internal representations — guided by the gradients from a probing classifier targeting the subject's plurality — the model's predictions in the subject-verb agreement task could be altered. Thus, the authors concluded that probing classifiers can identify features that are actually used by the model. Later, the study by Elazar et al. (2021) presents a nuanced view. They explored the effects of erasing specific linguistic information from BERT's representation layers on language modeling. Using the iterative null space projection method (INLP; Ravfogel et al. (2020)), they systematically erased

linguistic information, such as Part-of-Speech and syntactic dependencies, from BERT's internal representations. The INLP process involves training (linear) probing classifiers to detect these linguistic properties and iteratively erasing the associated features until the representations are no longer predictive of the target property. When comparing the language modeling performance before and after such interventions, they observed that the removal of certain properties, such as phrase boundaries, which had high probing performance, didn't significantly impact language modeling performance. This led them to a conclusion contrasting with Giulianelli et al. (2018): probing classifiers may not always detect information that the model actively uses for its predictions.

Further expanding the scope, Vig et al. (2020) explored gender bias in pre-trained Transformer LMs through a comprehensive causal intervention analysis. They manipulated the grammatical gender in the input, attention weights, and individual neurons to measure their causal impacts on the model's behavior. They found that gender bias predominantly resides in a small part of the network and this bias can be traced back to both direct input influences and indirect pathways via individual neurons and attention heads. The implications of this study extend beyond gender bias, offering a structural-behavioral framework for broader research aimed at interpreting and understanding the inner workings of neural NLP models.

Limitations Causal intervention analysis presents a unique perspective for establishing causality in interpreting deep NLP models, thus addressing certain limitations of challenge tests and probing classifiers. However, its implementation can be computationally expensive, especially in complex scenarios like neural-level interventions, and establishing clear cause-and-effect relationships within expansive networks is intricate. These complexities limit its practical application, particularly with state-of-the-art models.

Part II

Assessing model capacity to represent syntactic structures

Chapter 3

Long-distance agreement in neural language models

After surveying the landscape of various approaches and diverse conclusions on the linguistic capacities of neural NLP models, we now move into a focused review of one widely used approach: long-distance agreement tasks. This approach provides a compelling lens through which to investigate the ability of neural models to capture syntactic structure. At its core, syntactic agreement is a fundamental aspect of syntax, where certain sentence elements must align in features like number, gender, or person. Long-distance dependencies inherently require an understanding of how components in a sentence relate across spans of text, and crucially, morphological cues such as number and gender explicitly denote these long-term dependencies, offering a clear means to assess whether models effectively establish these connections.

In this chapter, we present several key studies that used the long-distance agreement paradigm — especially subject-verb agreement — to evaluate the ability of neural language models to capture syntactic information.

Subject-verb agreement processing, a well-established paradigm in psycholinguistics, is commonly used to study human syntactic ability. Studies in this domain suggest that humans rely on hierarchical structures to ensure syntactic coherence (Bock and Cutting, 1992; Bock and Miller, 1991; Bock et al., 2001). The work of Elman (1991), one of the first to analyze the syntactic capacity of neural networks, used the resolution of subject-verb agreement to demonstrate that a simple recurrent network is capable of encoding relevant grammatical relations and hierarchical structures in its distributed representation. This experimental approach, revitalized by the seminal work of Linzen et al. (2016) (detailed in §2.3.1), has since been used in a tremendous number of works to explore the capacity of

neural networks to capture abstract information about linguistic structures (Wilcox et al., 2018; Gulordava et al., 2018; Giulianelli et al., 2018; Jumelet et al., 2019; Lasri et al., 2022b, ; i.a.).

Naturalistic data Using this paradigm, early studies (Linzen et al., 2016; Bernardy and Lappin, 2017) showed that RNN models could handle the subject-verb number agreement task when given explicit supervision. Lately, Gulordava et al. (2018), expanding on Linzen et al. (2016), revealed that an LSTM language model, pre-trained only to predict the next word in an unannotated corpus, could effectively handle long-distance agreement in an unsupervised manner.

Specifically, Gulordava and colleagues trained an LSTM language model on a corpus from Wikipedia with 100M tokens. This model was then tested on its ability to handle long-distance number agreements using sentences extracted from Universal Dependency treebanks. The evaluation method involves presenting the pre-trained model with sentence prefixes up to the target verb, and then comparing the probabilities that the model assigned to the singular and plural form of the target verb. For instance, in example (4), if the model predicts a higher probability for "are" over "is", it is deemed to have made the correct prediction for that sentence. Consequently, the overall accuracy for the agreement prediction task is calculated as the percentage of test instances in which the verb form with the higher probability is indeed the correct one.

(4) The old rusty keys to the cabinet _
$\mathbb{P}(\textbf{are}|\text{prefix}) > \mathbb{P}(\textbf{is}|\text{prefix}) \Rightarrow$ predict "are"

Using this method, Gulordava et al. (2018) showed that LSTM achieved high accuracy in various constructions in the four languages tested: English, Italian, Hebrew, and Russian. In the case of Italian, the authors also conducted experiments with human subjects. The performance of the LSTM language model was at pair with human performance.

Furthermore, Gulordava and colleagues introduced a control setting to ensure that the model did not use collocational information to determine the correct verb form. For instance, in the sentence "The cats on the mat meow loudly", a language model may prefer the correct agreement by encoding information about what typically meows (cats) and what does not (mat), without relying on the target abstract structural rule. Such a confounding factor could overstate model success and raise questions about whether surface statistical patterns rather than the intended abstract syntactic information are driving performance. Chomsky (1957) claimed that grammaticality should be considered as a pure matter of syntax and structure, independent from semantic meaning or significance. Therefore, a

sentence like “Colorless green ideas sleep furiously”, despite being nonsensical, remains grammatically well-formed. If a model can capture the syntactic structure exemplified by agreement phenomena in naturalistic datasets, it should also be able to learn the syntactic constraints of nonsensical sentences. Inspired by this concept, Gulordava and colleagues also evaluated LSTMs on grammatically well-formed yet semantically implausible test instances, with the same number agreement prediction task. Specifically, a nonsensical evaluation set was created by replacing each content word of the original corpus-extracted sentence with a random word sharing the same PoS and morphological features:

(5) ORIGINAL: The old rusty keys to the cabinet (are/∗is) ...
NONCE: The colorless green ideas to the door (are/∗is)... (paraphrasing Chomsky)

Their results showed that LSTM model’s performance on nonsensical sentences was only slightly lower than on original ones; in Italian, this difference was just 6.6%, a similar performance drop was observed in human subjects. This highlights model’s ability to predict agreement in the absence of lexical or semantic cues and thus rules out the possibility that the LSTM decisions relied solely on surface information.

Formal languages Another way to isolate genuine syntactic processing from semantic information is to use formal languages. Lakretz et al. (2021a) investigated RNN’s ability to handle recursively nested subject-verb agreements, using artificial data generated by a PCFG. To illustrate, consider the example:

(6) a2 a1 **n3[sg]** a5 a3 **n1[pl]** a2 a2 **v5[pl]** a4 a1 **v4[sg]** a2 a5

Here, tokens starting with ‘a’,‘n’ and‘v’ represent adjective-, noun- and verb-like tokens, respectively. Tokens marked with number information, highlighted in bold, ensure that nouns and verbs at each nested depth (in this case, depth$= 2$) exhibit number agreement. The surrounding adjective-like tokens control dependency length, spanning two units in length on either side. Lakretz and colleagues created such training datasets, varying in terms of nested tree depths and dependency lengths. They then assessed RNN language models, trained with a language modeling objective, on subject-verb agreement tasks in controlled, incrementally challenging scenarios. Findings from this study indicated that while RNN language models could generalize to longer dependencies, they struggled with deeper tree structures.

Synthetic data Given the sparsity of complex syntactic sentences in treebanks (Gulordava et al., 2018) and the limited scope of the formal language approach, which often explores

specific syntactic facets in artificial settings, the use of synthetic data becomes an appealing alternative. Marvin and Linzen (2018) developed a template-based syntactic evaluation dataset, which features pairs of sentences, identical in all respects except for their grammaticality as shown in (7), targeting diverse structures-sensitive phenomena. In exploring the subject-verb agreement, their work delved into well-controlled challenging scenarios, where intervening elements such as prepositional phrases, relative clauses, or verb phrase coordination, separate the target subject and verb. For evaluation, instead of solely comparing the probability an LM assigns to a pair of words, they assessed the probabilities of entire sentences, determining if the model favored the grammatical over the ungrammatical sentence.

(7) $\mathbb{P}(\text{sentence a.}) \overset{?}{>} \mathbb{P}(\text{sentence b.})$

a. The farmer that the parents love swims.

b. *The farmer that the parents love swim.

This evaluation method extends to scenarios where multiple words may contribute to ungrammaticality, such as negative-polarity items. Their findings highlighted that while RNN language models excelled at local subject-verb agreements (i.e., no attractor), they exhibited sensitivity to specific lexical items and faced difficulties with rarer patterns, such as agreement across an object relative clause. Subsequent studies broadened the scope to include other phenomena considered by linguists to be sensitive to hierarchical structures, such as argument structure alternation, and filler-gap dependencies, as detailed in Section 2.3.1.

Abstract representations On the other hand, several studies have pointed out the limitations of relying solely on the agreement prediction approach to assess the representation of abstract syntactic structures by neural models. For instance, Kuncoro et al. (2018a) found that artificial neural networks may exploit spurious correlation in agreement tests without actually acquiring the desired syntactic competence: In the test set from Linzen et al. (2016), the agreement controller is the first noun in 80% of sentences with multiple *attractors*. This means that a simplistic heuristic, like agreeing with the first noun, can handle most of the complex agreement cases.

In addition, Newman et al. (2021) raised concerns regarding the hand-crafted minimal pair setting commonly used in agreement prediction tasks. While evaluating models based on their agreement outputs does not provide insights on their internal representations, this minimal pair setting further limits this approach to systematically capture a model's syntactic behavior. For instance, when given the prefix “The keys to the cabinet", the commonly used

metric compares only one verb pair's probabilities: is/are, as illustrated in (8a). However, this evaluation does not account for the model's overall probability distribution across vocabulary. So, even if a model correctly predicts "are" for the *be* pair, it could err in other contexts, such as favoring "exists" over "exist", as shown in (8b), when not restricted to choose specific verb forms. To assess the broader syntactic understanding of a model, complementary methodologies going beyond behavioral tests are required.

(8) The keys to the cabinet __

a. $\mathbb{P}(\textbf{are}|\text{prefix}) > \mathbb{P}(\textbf{is}|\text{prefix}) \Rightarrow$ predict "are", plural form

b. $\mathbb{P}(\textbf{exists}|\text{prefix}) > \mathbb{P}(\textbf{exist}|\text{prefix}) \Rightarrow$ predict "exists", singular form

Delving deeper, another research strand has focused on exploring models internal representations and inner workings. Giulianelli et al. (2018) conducted one of the first studies to investigate mechanisms tracking subject-verb agreement in LSTMs. After replicating the number agreement experiments of Gulordava et al. (2018), they used probing classifiers (§2.3.2) to analyze where and how LSTMs represented the agreement information: Classifiers were trained to predict the number information of the target subject ('singular' or 'plural') from LSTM's internal representations for all tokens in a sentence. The results revealed that in sentences where the LSTM accurately predicted the verb, the classifiers could retrieve the agreement information with high accuracy. Intriguingly, in cases where the LSTM chose an ungrammatical verb, the error in number encoding occurred early on, long before the verb's appearance. Furthermore, the study used the gradients of the classifiers to rectify the model's internal states at the timestep when the classifier first detected incorrect number encoding. After this single intervention, the model showed a significant improvement in its number agreement predictions, indicating that such encoded information detected by probing classifiers directly influenced the LSTM model's predictions.

Lakretz et al. (2019) investigated the neuron-level mechanisms within the RNN model of Gulordava et al. (2018), examining how the model processed long-distance agreement. Using neuron-level ablations, where specific neuron activations were set to 0, they assessed the impact of individual neurons on the model's syntactic performance. Within the model, only two units were identified as responsible for encoding grammatical number for long-distance dependencies; deactivating these units caused the network's performance approach chance level. These two long-range "number units" were intricately connected to a distinct set of "syntax units" that encoded the syntactic structures. One such syntax unit was specialized in tracking the main subject-verb dependency, indicating when to store or erase number information within the long-range number units. On top of these structure-aware units, a set of short-range number units was identified, which determined agreement based on linear-

distance — the most recent noun. This interaction created a sparse mechanism, consisting of only three units for long-range agreement, which enables the model to carry the main subject's grammatical number over long distances. However, such a sparse mechanism makes nested long-range dependencies challenging. For example, in (9), after recording the outer dependency and number (keys, plural), the model lacked available long-range units. Thus, the agreement in the embedded clause agreement (man–holds) had to rely on short-range units, which can be misled by attractors. This deep dive into the RNN's agreement mechanism provides a foundation for comparative studies between the model's syntactic behavior and human cognition. It provides actionable hypotheses that can be tested to better understand human syntactic processing (Lakretz et al., 2020, 2021b).

(9) The **keys**$_1$ [that the **man**$_2$ near the cabinets **holds**$_2$] **are**$_1$ rusty.

Shift to Transformer-based models The studies reviewed so far in this chapter have focused on RNN language models. As I began my thesis, Transformer-based models began to redefine the state-of-the-art in NLP and other fields, leading to a noticeable shift in the community's focus towards them. Consequently, in the realm of interpretability and explainability, a plethora of research has emerged to evaluate the linguistic capabilities of Transformer models. Among these investigations, the long-distance agreement task remains a popular tool to probe the structure-sensitive generalization capabilities of these Transformer models.

After BERT and GPT's impressive syntactic capabilities were confirmed by replicating the agreement experiments of Linzen et al. (2016) and Gulordava et al. (2018) in studies like Goldberg (2019) and Wolf (2019), later research aimed to uncover the mechanisms behind these models' proficiency in handling long-distance dependencies. For instance, many studies, ours included, applied causal intervention analysis on Transformer models to uncover their strategies for resolving long-distance agreements (Finlayson et al., 2021; Lasri et al., 2022b; Li et al., 2022a). Others explored how subject-verb agreement resolution in Transformers was influenced by factors independent of structure, examining frequency effects as in Wei et al. (2021), lexical information as in Lasri et al. (2022a) and surface heuristics as in our work Li et al. (2021).

More recent research has expanded beyond English, with numerous studies assessing Transformer models on non-English linguistic structures, leading to varied conclusions about their syntactic capabilities. For example, Guarasci et al. (2023) evaluated BERT's ability to learn Italian syntax, and de Dios-Flores et al. (2023) probed BERT's understanding of control dependencies in Spanish and Galician, highlighting model's difficulty with non-adjacent

dependencies. Meanwhile, many non-English challenge sets have also been introduced. For instance Wilkens et al. (2023) developed a dataset for Brazilian Portuguese that included various grammatical structures, in particular agreement phenomena. Someya and Oseki (2023) introduced the Japanese benchmark of linguistic minimal pairs, covering 11 intricate linguistic phenomena, and highlighted challenges in verbal agreement and binding. An et al. (2023) crafted a French synthetic benchmark for subject-verb agreement, modeling it after a visual pattern detection task inspired by Raven (1941).

In this chapter, I have outlined several key studies[1] that employ the long-distance agreement task to assess the ability of neural language models to capture syntactic information. Their methodologies and diverse conclusions form the premise of the following chapter, in which we combine challenge sets, probing classifiers, and causal interventions to investigate the mechanisms tracking long-distance agreement in an autoregressive Transformer language model.

[1]It is worth noting that this overview is not exhaustive, recently, this field has seen numerous follow-up studies and related work that I have not been able to cite or detail extensively.

Chapter 4

A contrastive study of NLM's syntactic abstraction based on long-distance agreement

4.1 Introduction . . . 45
4.2 Revisiting number agreement tasks via a heuristic-based evaluation protocol . . . 47
4.2.1 Syntactic phenomena . . . 48
4.2.2 Datasets construction . . . 50
4.2.3 Experimental setup . . . 53
4.2.4 Heuristic-based evaluation protocol . . . 56
4.2.5 Control experiments . . . 61
4.2.6 Conclusion . . . 69
4.3 Locating syntactic information in Transformer language model . . . 70
4.3.1 Distribution of syntactic agreement information across token positions . . . 71
4.3.2 Probing internal representations components . . . 76
4.3.3 Conclusion . . . 78
4.4 Right for the right reason: Exploring mechanisms of agreement computations . . . 79
4.4.1 The Causal Framework . . . 80
4.4.2 Causal experiments and results . . . 85
4.4.3 Conclusion . . . 88
4.5 Word order: the impact of positional encoding on NLM's syntactic abstraction capacity . . . 89
4.5.1 Positional embeddings in Autoregressive Transformer LM 90
4.5.2 Positional embeddings in masked Transformer LM . . . 91
4.5.3 Conclusion . . . 93
4.6 Conclusion and discussion . . . 93

This chapter, forming the centerpiece of this dissertation, builds directly upon the studies reviewed in Chapter 3. We expand previous research by conducting a contrastive analysis of a Transformer model through two carefully crafted long-distance agreement tasks in French. We aim to investigate how well the Transformer can handle these two strucutre-sensitive phenomena, and whether its performance stems from its ability to build an abstract, high-level (maybe hierarchical) sentence representation (Giulianelli et al., 2018; Lakretz et al., 2019) or merely because it captures surface statistical regularities, as suggested by previous studies (Sennhauser and Berwick, 2018; Chaves, 2020; Li and Wisniewski, 2021). To effectively evaluate the model's syntactic capacity, we first introduce a novel heuristic-based evaluation protocol, which enables us to probe the model's ability to handle agreement tasks beyond superficial heuristics. We then use probing approaches, paired with causal analysis, to identify the location of the syntactic information within the model and determine how the model actually uses this information for agreement resolution.

Chapter outline This chapter is structured as follows: Section 4.1 introduces the research question and relevant background concepts, as well as the two agreement tasks that are central to our study. In Section 4.2, we revisit the number agreement tasks through a heuristic-based evaluation protocol to address potential confounding factors. These refined tasks and the evaluation protocol serve as the foundation for all subsequent experiments outlined in this chapter. In Section 4.3, we investigate the specific location of syntactic information within the autoregressive Transformer language model. Following this, Section 4.4 analyses how the model uses this encoded syntactic information to process long-distance agreement phenomena. In Section 4.5, we explore the relationship between the model's ability to abstract syntactic structure and the sequential word order information presented in the input sequences. The chapter concludes with Section 4.6, where we recapitulate our findings and discuss their implications.

4.1 Introduction

Transformers-based language models (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020) have reshaped NLP with their unparalleled performance across a wide range of language tasks. Their empirical success, coupled with the findings of previous studies (see Section 2.3), indicates that these models potentially have acquired a certain level of abstraction in understanding language structure. Since Linzen et al. (2016), the long distance agreement task has been a paradigmatic test for assessing the ability of NLMs to uncover syntactic information from raw texts: a model able to predict the long-distance agreement dependency, has to, to some extent, develop an abstract representation of the syntactic

structure and encode it in its internal representations.

In this study, we investigate how Transformer language models process and represent syntactic structure through long-distance agreements tasks. The essential research question we aim to explore is: When resolving long-distance agreements, to what extent do models abstract their representations from surface pattern recognition, and are they able to develop meaningful, syntactically driven representations of linguistic structure?

By addressing this question, we can evaluate model's representational adequacy for modeling syntactic structures and develop linguistically-informed analysis tools to enhance our understanding and control over these models. Such insights are crucial for evaluating these models as potential explanatory explanatory models for human language processing. Moreover, delving into the linguistic abstraction of these models can provide insight into the properties that contribute to the success of NLMs but also identify their limitations, which could help guide the creation of more effective architectures. For instance, previous studies find that modeling explicitly hierarchical structure as an inductive bias of RNN models helps them learn structure-sensitive phenomena more effectively (Kuncoro et al., 2018b; Wilcox et al., 2019). Despite the remarkable empirical success of Transformer-based models, they can be fragile, especially when faced with noisy or adversarial inputs (Wang et al., 2022). Integrating human linguistic priors into these models might provide added robustness and optimize learning efficiency (Lake et al., 2017; Besold et al., 2017).

To explore this question, we focus on two types of number agreement phenomena in French, both feature morphological markings:

(10) Les **chat·s** [que Noûr aime bien]$_{RC}$ **jou·ent** dans le jardin.
The_Pl cats_**Pl** [that Noûr likes_Sg a_lot]$_{RC}$ play_**Pl** in the garden.

(11) Les **chat·s** [que Noûr a **adopté·s**]$_{RC}$ sont mignons.
The_Pl cats_**Pl** [that Noûr has adopted_**Pl**]$_{RC}$ are_Pl cute_Pl

Example (10) demonstrates a subject-verb agreement between the noun "chats" and the main verb "jouent" across a relative clause, while (11) showcases an object-past participle agreement between the same noun "chats" and the past participle "adoptés". At first glance, (10) and (11) may appear to represent identical agreements between a noun and a verbal form separated by a few words. Yet from a linguistic perspective they are substantially different: while the former involves the subject controlling the main verb's number, the latter involves anaphora resolution and movement—operations that are fundamentally different from the phrase structure embedding in the subject-verb agreement (see §4.2.1 for more detailed description).

It is unclear whether and how a Transformer language model can identify these abstract representations based merely on the words sequence. The present work aims to contrast how Transformer handles these two kinds of agreement. Specifically, we seek to determine whether the Transformer LM encodes the **same** abstract structure in its internal representations to capture the information required for agreement resolutions, or if it instead encodes an abstract structure that reflects the **distinction** made in the theoretical modeling of these two agreements. This contrast will shed new light on our understanding of the internal workings of Transformer models.

This chapter offers two key contributions. First, we expand the existing syntactic evaluation paradigm by conducting a contrastive analysis of a Transformer model's ability to abstractly represent two superficially similar syntactic phenomena in French: long-distance subject-verb agreement and a less studied phenomenon, object-past participle agreement. Second, we introduce an integrated linguistically-informed analysis framework that can serve as a template for empirically testing linguistic or cognitive theories with computational models.

As an initial step, we introduce a novel heuristic-based evaluation protocol to revisit conventional number agreement tasks. This helps to discern whether the model relies on structural patterns or surface-level heuristics. Our findings indicate that Transformer models excel at both agreement tasks, successfully abstracting away from potential lexical or heuristic confounds. Subsequently, we use probing classifiers and causal intervention on self-attention to examine **where** the Transformer model encodes syntactic information internally and **how** the model uses it in agreement resolution tasks. The results reveal that for both phenomena, even though the long-distance agreement information is mainly encoded locally across the tokens between the two agreeing elements, Transformer model deploys distinct, linguistically motivated strategies to process each phenomenon. Lastly, through ablation studies, we explore the role of positional embeddings in the Transformer's architecture.

4.2 Revisiting number agreement tasks via a heuristic-based evaluation protocol

As discussed in Chapter 3, many recent studies have demonstrated that unsupervised sentence representations generated by neural language models encode syntactic information, as evidenced by their success in predicting long-distance agreements. However, conventional behavioral assessments, which focus solely on output, fall short of determining whether a model's success in these tasks arises from genuine syntactic understanding or from exploiting

superficial patterns in the data.

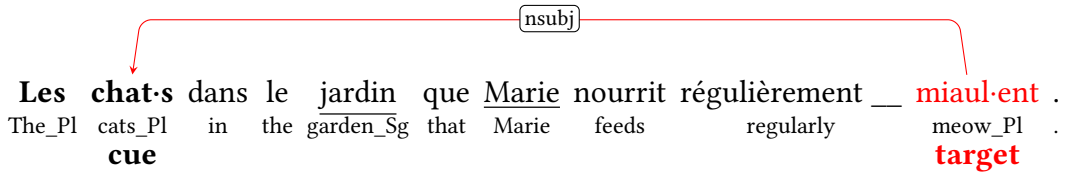
To address this issue, we introduce a heuristic-based evaluation protocol tailored for agreement tasks. This protocol enables us to identify cases where the correct answer cannot be inferred through simple surface-level heuristics. If the model still performs well under these conditions, it would strongly suggest that it has indeed acquired a level of non-superficial syntactic competence. We further complement this with control experiments aimed at assessing other confounding factors that might influence the model's predictions. This multi-faceted evaluation strategy lays the groundwork for the subsequent development and assessment of different interpretation techniques, as we will explore in Sections 4.3 and 4.4.

4.2.1 Syntactic phenomena

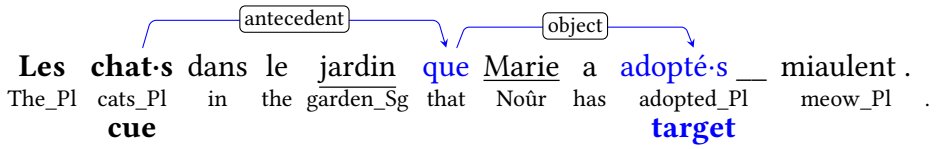
In this study, we extend the agreement predictions approach to non-English languages by considering French, a morpho-syntactically richer language. Unlike English, where agreement is primarily limited to subject-verb pairs in the third person, singular present tense, French exhibits a wider range of agreement features, including gender and number agreement across various grammatical categories like adjectives, pronouns, articles, and past participles. This complexity provides a richer testing ground for exploring the syntactic capabilities of neural language models.

The number agreement tasks in our study address two agreement phenomena: subject-verb agreement across relative clauses (henceforth *S-V agreement*) and object-past participle agreement (henceforth *O-PP agreement*) in French. In the following sections, for both types of agreement we refer to the noun item providing the agreement information the **cue**, and the verbal item as the **target**. We focus exclusively on sentences involving *object relatives* such as those analyzed in Figure 4.1, where the words that intervened between the *cue* and the *target* contain at least one relative clause. Despite the superficial similarities between the two phenomena — both featuring a relationship between a noun and a verbal form separated by a few words containing relative clause elements — they receive significantly different linguistic analyses.

The subject-verb agreement across relative clauses is a case that clearly necessitates hierarchical representation. In (4.1a), the main verb *miaulent* (meow) must agree in number with its syntactic subject *chats* (cats), regardless of the intervening elements. The relative pronoun *que* (that) and the entire embedded clause are not relevant for determining the form of the main verb. The model needs to distinguish the main clause subject (*chats*) from the embedded subject (*Marie*) and ignore irrelevant attractors like *jardin*. This requires a



(a) Example of subject-verb agreement



(b) Example of object-past participle agreement

Figure 4.1: In (a), the number of the main verb (*miaulent*, in red) is determined by the head of the subject *chats*. In (b), the past participle in the relative clause (*adopté*, in blue) has to agree in gender and number with its object (also in blue) when the latter precedes the verb.

.

nuanced representation of clause-specific syntax and verb argument structure.

Compared to subject-verb dependency, the agreement of the past participle in object relative clauses relies on an abstract set of relations between words occurring in different clauses. In French, the past participle conjugated with the auxiliary *avoir* (have) in compound tenses, such as passé composé, must agree in number and gender with the direct object that precedes it.[1] As shown in Figure 4.1b, the past participle within relative clause agrees in number and gender with its complement (the **cue**) in the main clause, because the latter moves before it. Specifically, when agreement is required, a `-s` suffix (resp. `-e`) is added to the singular masculine form for plural objects (resp. feminine). This agreement resolution involves an anaphora (indicated by the *antecedent* arc) and a filler-gap dependency. The filler is *que* (that) and the gap, indicated with an underscore in Figure 4.1, is an empty syntactic position licensed by the filler (Kayne and Benincà, 1989). In the example (4.1b), the relative pronoun *que* is the pre-verbal direct object of the past participle *adoptés* and triggers the agreement of the past participle. To obtain its agreement features, the relative pronoun has to be linked by anaphora to its nominal antecedent *chats*. In other words, to correctly agree the past participle in theory, it is necessary to identify the object relative pronoun *que* and its antecedent. Additionally, the model has to ignore the effect of attractors occurring between the antecedent of the relative pronoun and the past participle.

We only consider number agreement as *i)* number agreement is the only feature shared

[1]Although in standard French, normative grammars indicate object-past participle agreement under wh-movement as obligatory, it in fact appears to be optional in colloquial French, where the past participle is often produced in its default form, which, in French, corresponds to the singular, masculine form of the participle (Belletti, 2017). Please refer to §4.2.3 for a relevant analysis of the training data used in this study.

by the two agreements we consider;[2] *ii)* the main purpose is to design reasonably simple patterns allowing to extract a sufficiently large number of representative examples. These restrictions allow us to carry out a fine-grained contrastive analysis of NLMs ability to extract syntactic generalizations from non-annotated corpora (§4.2.4).

4.2.2 Datasets construction

As discussed in Chapter 3, common approaches for creating challenge sets typically rely on template-based generation or extraction from gold parses. Template-based synthetic data, similar to the artificial stimuli employed in human linguistic experiments, provides controlled testing grounds. Yet, they may lack ecological validity and the variability of natural language. For example, our Transformer language model, trained on French Wikipedia text, achieved a perplexity score of 27. However, the score rises to 308 for materials from human experiments on French object-past participle agreement used in Villata (2017), and increases to 654 on a synthetic corpus for French subject-verb agreement across relative clauses from Mueller et al. (2020). This score discrepancy highlights the potential detachment of synthetic data from the natural linguistic landscape, which could, in turn, affect the assessment of model capabilities. On the other hand, while gold parses from existing treebanks ensure accuracy, they may not provide a sufficient number of syntactically challenging examples. For instance, only 41 sentences were available for English in the study by Gulordava et al. (2018).

To overcome these limitations, we adopt an approach focusing on naturally occurring sentences. This approach not only respects the ecological validity, but also reflects the complexity and diversity inherent in natural language. By extracting our target sentences from a large automatically parsed corpus, we collect a substantial and varied set of test items. Furthermore, we conduct a qualitative evaluation of our automatic parsing and extraction pipeline to guarantee the quality of the evaluation sets.

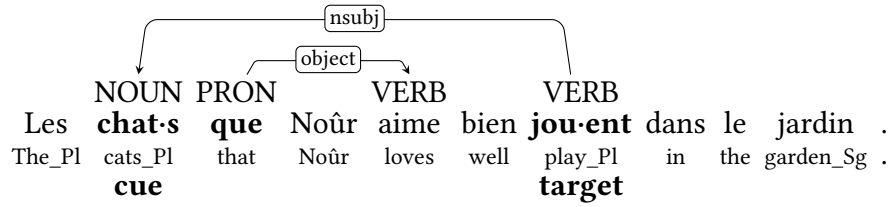
Overview To construct the evaluation datasets for the number agreement tasks we consider, sentences were automatically extracted from the French part of Project Gutenberg,[3] which contains over 8 million sentences. We used the French dependency parser (Grobol and Crabbé, 2021) along with pretrained French model from *spaCy* (Honnibal et al., 2020) to parse the corpora,[4] from which examples of target agreement phenomena were extracted

[2] In French, the verb has to agree in number with its subject, and the past participle conjugated with the auxiliary *avoir* agrees in number *and* in gender with its direct object if the latter appears before it.

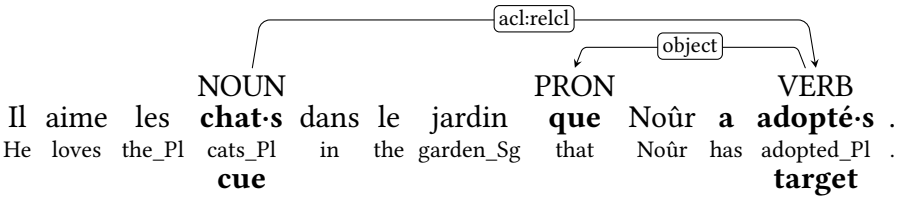
[3] `https://www.gutenberg.org/`

[4] The parser by Grobol and Crabbé (2021) annotated only the part-of-speech tags and dependency relations, *spaCy* supplemented these annotations with morphological features.

using simple rules (detailled in the following paragraph). This resulted in two evaluation sets: one for object-past participle agreement (O-PP agreement), consisting of 68,794 sentences (65% with a singular **target** and 35% with a plural one), and spanning 837 past participle lemmas and 2,489 word forms. Another for subject-verb agreement (S-V agreement) across relative clauses, consisting of 27,582 sentences (70% with a singular **target** 30% with a plural one), and spanning 536 verb lemmas and 1533 verb forms. Both sets consist of sentences including at least one object relative clause between the **cue** and **target**. There are fewer items in the S-V agreement set because noun phrases modified by relative clause(s) occur more frequently in the object position than in the subject position of the main clause. In these two evaluation sets, an arbitrary number of words can occur between the **cue** and **target**: an average of five tokens occur between the antecedent and the past participle, and 11 tokens between the head of the subject and the main verb. These "intervening" tokens include varied constructions such as prepositional phrases, participials, or nested relative clauses, which can pose additional challenges for agreement tasks. See Section A.3 in the Appendix for sample sentences from the two evaluation sets. Note that in all our experiments, we ensure that the evaluation sets were completely separate from models training data.



(a) Example of the extraction pattern for subject-verb agreement across relative clauses



(b) Example of the extraction pattern for object-past participle agreement

Extraction procedure The extraction rules used to construct evaluations datasets are based on the predicted dependency structure and morphological information of sentences. As illustrated in Figure 4.2a, for subject-verb agreement, a valid example must include a Noun and Verb connected by a *nsubj* dependency and at least one relative pronoun *que* acting as a direct object between them. For object-past participle agreement, as shown in Figure 4.2b, a valid example has to include a Noun and Verb connected by an *acl:relcl*

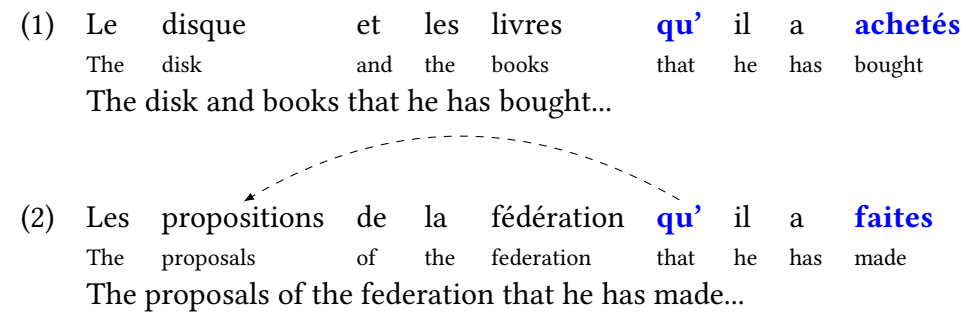


Figure 4.3: The test set excludes sentences with coordinate **cue** as shown in (1). But it includes syntactic phrases as **cue** cases like in (2), as the antecedent of the relative pronoun is unambiguous.

dependency, with a direct object *que* preceding the VERB, and the auxiliary used by the target VERB must be *avoir* (to have).

In the next step, we filtered out sentences containing ambiguous or non-agreement **target**-**cue** pairs. Based on syntactic-morphological information, we excluded sentences that involved the **cue** being part of a coordination structure, such as example (1) in Figure 4.3, and those with number-ambiguous **cue** or **target**, including collective nouns or nouns and verbs with identical singular and plural forms.[5] However, we retained syntactic phrases as **cue** cases, because they are typically endocentric: the head determines without ambiguity the agreement requirements for the entire phrase. For instance, in example (2) of Figure 4.3, the word *propositions* heads the entire NP modified by a prepositional phrase, and the upcoming verb should agree with *propositions* in number. We also excluded sentences in which the noun and the verb do not agree in number (and in gender for past participle agreement cases), as well as those in which not all words from the **cue** to the **target** were present in the language model's vocabulary.

Qualitative evaluation of extraction procedure Given that our evaluation sets are extracted from automatically parsed corpora, there is an inherent risk of introducing errors into the dataset (Bender et al., 2011). This makes a qualitative analysis especially crucial for our study. To assess the effectiveness of our automatic parsing and extraction procedure, we conducted a qualitative analysis based on the French Universal Dependency treebanks.[6] Using the object-past participle agreement pattern in Figure 4.2b, we identified a set of 107 valid sentences (68% singular and 32% plural) from the gold annotations of French treebanks.

[5] We filtered out nouns with endings in `-s`, `-x`, `-z` and past-participles that ends with `-s`, as these forms often remain the same in both singular and plural in French.

[6] `https://universaldependencies.org/`, we used the version 2.7 of the UD project.

We then used the parsers from our previously established automatic extraction procedure to parse the French treebanks. From this analysis, we extracted 106 instances of object-past participle agreement, achieving a precision of 99% and a recall of 98%. The single error, illustrated in (12), involves the incorrect identification of the antecedent as *manière* (way), instead of *révolution* (revolution). However, since we do not need to correctly identify the **cue** to create a valid test item for number agreement tasks, this error is inconsequential. Additionally, two instances were missed due to incorrect annotation of the intervening relative pronoun *qu'* in (13), and a verb attachment error, respectively. These high scores suggest that our automated process is reliable for the aims of this study.

(12) Une manière de **révolution** sur lui-même , qu'il a **opérée** en 1981
A way of revolution_**Fem_Sg** on himself , that he has operated_**Fem_Sg** in 1981

(13) la **formule** qu'avec un sens de la nuance plus marseillais que britannique,
the formula_**Fem_Sg** that with a sense of the nuance more Marseillais than British,
le président de l'académie a **appliquée**
the president of the academy has applied_**Fem_Sg**

4.2.3 Experimental setup

Models In this chapter, we focus on an autoregressive Transformer language model while also including an LSTM langauge model as a strong baseline for comparison. Both of these generative language models (See detailed description in §2.2) are designed to estimate the probability of a sentence $\mathbf{x}$ as:

$$\mathbb{P}(\mathbf{x}) = \prod_{i=1}^{n} \mathbb{P}(x_i | x_1 \dots x_{i-1}) \tag{4.1}$$

All models are trained to compute $\mathbb{P}(x_i | x_1 \dots x_{i-1})$ and they all use the same generic template:

$$\mathbb{P}(x_i | x_1 \dots x_{i-1}) = \textsc{softmax}(\mathbf{W}_{dec}\mathbf{c}_{i-1} + \mathbf{b}) \tag{4.2}$$

$$\mathbf{c}_{i-1} = \textsc{context}(\mathbf{e}_1 \dots \mathbf{e}_{i-1}) \tag{4.3}$$

$$\mathbf{e}_i = \mathbf{W}_{enc}\mathbf{x}_i \tag{4.4}$$

where $\mathbf{x}_i$ is one-hot word vector; $\mathbf{W}_{enc}$ and $\mathbf{W}_{dec}$ are tied parameter matrices, the latter being the transpose of the former, encoding respectively the word embeddings and the output layer of the language model. A context model (CONTEXT) is either an incremental LSTM or a Transformer decoder where the sequence of embeddings $\mathbf{e}_i \dots \mathbf{e}_n$ is masked (i.e. the probability of the i-th word is estimated knowing only the first (i-1) words of the

sentences, contrary to the "standard" Transformer models which assume that the whole sentence is known). The context vector $\mathbf{c}$ returned by the context model is either the hidden vector of the LSTM at step $i-1$ or the vector returned by the upper layer of the Transformer at step $i-1$.

Prior research mainly used sub-word-based pretrained models, which could only directly score words represented as a single wordpiece. Studies such as those by Goldberg (2019) and Lasri et al. (2022b) dealt with this limitation by restricting their evaluations to verbs that appear as single wordpieces in the model's vocabulary. We avoid such compromises by implementing word-based RNN and Transformer models using the PyTorch library,[7] offering a more suitable and flexible framework for linguistic experiments.

The studies in this chapter focus mainly on the Transformer model with 16 layers and 16 heads, featuring a total of 127 million parameters, denoted as $\mathcal{M}$. This model is comparable in size to the GPT-2 base model (Radford et al., 2019; Solaiman et al., 2019). To provide a strong baseline, we additionally incorporate a 2-layer LSTM model in Section 4.2, as it has shown a strong ability in resolving various number agreement tasks in English in prior research.[8] To provide a more nuanced comparison between the two architectures, we also include in Section 4.2, two Transformers language models that have a number of parameters comparable to our LSTM model: one featuring 2 layers, denoted as $\mathcal{M}_{shallow}$, while the other, $\mathcal{M}_{shared}$, has 16 layers with weights shared across all layers (Dehghani et al., 2018) (see Table 4.1 for details).[9] All models use embeddings of size 768 and are pre-trained on the same data, allowing for a reasonably fair comparison across models. For Transformers we add positional embeddings to the word embeddings $\mathbf{e}_i$ using the sinusoidal scheme and weighting described by Vaswani et al. (2017). We bound the vocabulary to the 50,000 most frequent tokens found in the training data and use an <unk> token to encode the least frequent tokens.

Language model training To train the language models, we extracted raw text from a French Wikipedia dump[10] using *WikiExtractor* (Attardi, 2015). We then segmented and tokenized it with the *Moses* tokenizer (Koehn et al., 2007). To ensure the quality of the dataset, we filtered out sentences with more than 5% unknown words based on the lemma annotations generated by *TreeTagger* (Schmid, 1995). Once filtered, we sampled a subset containing 100 million tokens, which mirrors the linguistic exposure of an 8-year-old (Brysbaert et al., 2016).

[7] `https://gitlab.huma-num.fr/bli/syntactic-ability-nlm`

[8] Many previous related studies have focused on LSTM, including LSTM models in this work also facilitates a comparison of subject-verb agreement results obtained in French with those from other languages reported in the literature.

[9] Considering an LSTM model with a larger number of parameters is computationally not tractable.

[10] We used the version of 2020-11-09 from: `https://dumps.wikimedia.org/frwiki/`

	#layers	#attention heads	#param ($\times 10^6$)	PPL
LSTM	2	–	48	$36.9_{\pm 0.1}$
$\mathcal{M}$	16	16	127	$27.0_{\pm 0.2}$
$\mathcal{M}_{shallow}$	2	16	49	$37.8_{\pm 0.7}$
$\mathcal{M}_{shared}$	16	16	48	$30.7_{\pm 0.6}$

Table 4.1: Parameters and perplexities (average across five models) of neural language models examined in this section.

This subset was then split into training, validation, and test sets with an 8:1:1 proportion.

We pre-trained all of our models using a language modeling objective, as described in Section 2.2.1. Training was carried out with stochastic gradient descent, with an initial learning rate to 0.02 and a cosine scheduling for 100 epochs without annealing. The first epoch was dedicated to warm-up, with a linear incremental schedule for the learning rate. The batch size was set to 64, running in parallel on 8 GPUs, except during the warm-up, where the size was fixed to 8. Hyperparameters were selected by minimizing the perplexity on the validation set, and the optimal combination of hyperparameters was used to train five models for each architecture. All results presented in this work are averaged across these five models. For further details regarding the models and hyperparameters tuning, please refer to Section A.1 in the Appendix.

Evaluation procedure We use the number agreement task (§3) to evaluate neural language models' ability to capture syntactic information. Language models provide us with a straightforward, unsupervised way to predict agreement: Let $\mathbb{P}(w_i|w_1, ..., w_{i-1})$ represent the predicted probability of a word w at position i in a sequence, conditioned on all preceding words $w_1, ..., w_{i-1}$ in the sequence. For each sentence in the number agreement test sets, we examine whether the condition in (4.5) holds. Specifically, we evaluate whether the model, given all the tokens preceding the **target**, allocates a higher probability to the correctly inflected target verb than to the verb inflected with the opposite number. We refer to this evaluation metric as *target verb evaluation*.

$$\mathbb{P}(w_{\text{target}}|w_1, ..., w_{i-1}) > \mathbb{P}(w_{\text{target}}^{\neg \text{ number}}|w_1, ..., w_{i-1}) \tag{4.5}$$

For instance, a pre-trained model is fed with a sentence prefix "Les chats dans le jardin que Marie a" in example (14), the expected upcoming target verb is in plural — *adoptés*. We then compare the probabilities the model assigns to the singular form *adopté* and the plural form *adoptés*. We consider the model has predicted the agreement correctly if the

form with the correct number has a higher probability than the form with the incorrect number, as illustrated in example 14a. Therefore, a model's syntactic ability is measured by the percentage of sentences for which the verb form with the higher probability is the one that respects the agreement rules of the language (i.e. matches the number of the **cue**).

(14) PREFIX: Les **chats** dans le jardin que Marie a __
The_Pl cats_**Pl** in the garden that Marie has __
EXPECTED VERB: **adoptés**, plural

a. $\mathbb{P}(\textbf{adoptés}|\text{prefix}) > \mathbb{P}(\textbf{adopté}|\text{prefix}) \Rightarrow$ predict "adoptés", plural ✓

b. $\mathbb{P}(\textbf{adopté}|\text{prefix}) > \mathbb{P}(\textbf{adoptés}|\text{prefix}) \Rightarrow$ predict "adopté", singular ✗

4.2.4 Heuristic-based evaluation protocol

Overall accuracy As shown in Table 4.2, all models achieve over 80% accuracy in both long-distance agreement tasks. Specifically, the LSTM made correct number prediction in 94.3% of the subject-verb agreement cases and in 82.1% of the object past participle agreement cases, a performance similar to those reported in the literature.[11] In most cases, the Transformer variants outperformed the LSTM model. These overall results support the conclusion, drawn by many studies, that neural networks are capable of tracking long-distance dependencies with high accuracy, which constitutes evidence that they encode a substantial amount of abstract syntactic information (§3).

Models & Baselines	S-V	O-PP
LSTM	$94.3_{\pm 0.3}$	$82.1_{\pm 1.1}$
$\mathcal{M}$	$98.9_{\pm 0.04}$	$94.6_{\pm 0.2}$
$\mathcal{M}_{shallow}$	$90.8_{\pm 0.4}$	$84.7_{\pm 0.7}$
$\mathcal{M}_{shared}$	$97.8_{\pm 0.3}$	$89.0_{\pm 0.3}$
Majority class	69.7	65.1
Surface rule: first noun	83.7	69.5

Table 4.2: Average accuracy (%) for both agreement tasks across five models for each architecture, compared to baselines.

However, we believe that this conclusion must be taken with great care. Confounding factors may enable a language model to produce correct predictions without genuinely

[11] For instance, for the subject-verb agreement task, Gulordava et al. (2018) reported an overall accuracy of 93.3% for Italian and Mueller et al. (2020) of 83% for a wide range of constructions in French.

capturing syntactic rules, as discussed in Chapter 3. For instance, the model could exploit surface-level patterns in natural language, where often the subject happens to be the first noun (Kuncoro et al., 2018a). In our subject-verb agreement evaluation set, a simplistic model that always matches the verb form with the first noun in the sentence can achieve an accuracy of 83.7%. This high score raises questions about the true nature of our models' capabilities. Given that both the abstract linguistic rule and the superficial pattern could lead to the same correct answer in most cases, it becomes hard to tell if a model is actually relying on the underlying hierarchical structure of sentence — the verb should match its grammatically determined subject in number, or simply exploiting the sequential pattern present in the data — the verb should match the first noun in number.

Heuristic-based evaluation Expanding upon the observation of Kuncoro et al. (2018a), we introduce five shallow heuristics in our framework that a statistical model could exploit to predict the verb's number only from surface information. These heuristics are organized in increasing order of complexity, and each one assumes that the **target** verb agrees in number systematically with:

h1. the *nearest token* marked for grammatical number;

h2. the *nearest noun*;

h3. the *first noun* of the sentence;

h4. the majority number expressed before the target;

h5. the *noun* preceding the closest *que* before the target.

It is worth noting that the fifth heuristic, which involves identifying relative pronouns, is arguably more complex and may not be as purely "surface-level" as the preceding ones.

(15) $_{(h4)}$Les chats$_{(h3)}$ dans les champs$_{(h5)}$ que Marie nourrit $\_\_{}_{Pl}$ ...
The$_{Pl}$ cats$_{Pl}$ in the$_{Pl}$ fields$_{Pl}$ that Marie$_{Sg}$ feeds$_{Sg}$ $\_\_{}_{Pl}$...

For instance, in the example (15), the sentence prefix anticipates a plural main verb, the correct verb form could be selected by applying heuristics of agreeing with the first noun (*h3*, plural), or agreeing with the noun preceding *that* (*h5*, plural), or the majority number expressed in the prefix (*h4*, plural). These heuristics are not tailored to the prediction of the two types of agreement in French, but can easily be adapted to other relevant tasks in different languages.

Interestingly, the accuracy of these heuristics on our evaluation sets ranges from 60.3% (for h1) to 95.7% (for h5), most of which are above the majority-class baselines, and the *h5* heuristic even outperforms the best model in the object past participle agreement task.[12] These observations call into question our previous conclusion, suggesting that the good performance of neural language models on agreement tasks could also result from their ability to extract and combine surface patterns rather than their capacity to learn underlying hierarchical structures. Given that the hierarchical structures appearing in natural language frequently co-occur with superficial statistical regularities, we propose in this study a heuristic-based evaluation protocol, which aims to mitigate this issue and provide a clearer understanding of NLM's capabilities in learning and processing language structures.

This novel evaluation protocol forms the first contribution of this dissertation. We propose using these heuristics to measure the prediction 'difficulty' of sentences in our evaluation sets. Specifically, for each test sentence, we count how many heuristics correctly predict the form of the target verb. The more heuristics that match, the easier the prediction task becomes. Therefore, a higher count of heuristics implies a lower prediction difficulty for a given sentence. Subsequently, we divide our test set into six subsets, each corresponding to the count of heuristics that match the form of the target verb. We then assess model performance across these varying levels of difficulty.

Count of heuristics	Difficulty of agreement	Examples
5	−−−	$_{(4)}$Si les **idées**$^{(3)}_{(5)}$ que ces mots$_{(2)}$ représentent$_{(1)}$ ne **sont** pas ... *If the **ideas** that these words represent **are** not...*
4	−−	$_{(4)}$Les **choses**$^{(3)}_{(5)}$ que nous avions vues cent fois avec indifférence nous$_{(1)}$ **touchent**... *The **things** that we had seen a hundred times with indifference **touch** us ...*
3	−	Un philosophe est curieux de savoir si les **idées**$^{(5)}_{(2)}$ qu'il a semées$_{(1)}$ **auront**... *A philosopher is curious to know if the **ideas** that he has sown **have**...*
2	+	Les **emblèmes**$^{(3)}_{(5)}$ qu'on y rencontre à chaque pas **disent** ... *The **emblems** that we meet at each step **say** ...*
1	++	Les **qualités**$_{(3)}$ qui t'ont fait arriver si jeune au grade que tu as **doivent** te porter ... *The **qualities** that made you arrive so young at the rank you have **should**$_{Pl}$ bring you ...*
0	+++	Ce soir les **hommes** que j'ai postés sur la route que doit suivre le roi **prendront** ... *Tonight the **men** that I have posted on the road that the king must follow **will_take**$_{Pl}$ ...*

Table 4.3: Examples from our evaluation set of subject-verb agreement, stratified by the count of surface heuristics predicting the *target*'s number, a proxy to the task difficulty. The target verbs and their subjects are in bold. The orange numbers in parentheses indicate the presence of different types of heuristics.

As illustrated in Table 4.3, the *5-heuristic* group gathers the 'easiest' examples: For instance, in the prefix "Si les$_{Pl}$ **idées**$_{Pl}$ que ces$_{Pl}$ mots$_{Pl}$ représentent$_{Pl}$ ne **sont**$_{Pl}$", all five heuristics match the target's number — plural. A model can easily predict the correct

[12]For a detailed breakdown of accuracies by each surface heuristic, see Table A.4 in the Appendix.

form of the target verb by simply applying any of the five surface heuristics (e.g. the target form should match the first noun/the nearest noun/...). In contrast, examples in the *0-heuristic* group are the most difficult. In the prefix "Ce$_{Sg}$ soir$_{Sg}$, les$_{Pl}$ **hommes**$_{Pl}$ que j'ai$_{Sg}$ postés$_{Pl}$ sur la$_{Sg}$ route$_{Sg}$ que doit$_{Sg}$ suivre le$_{Sg}$ roi$_{Sg}$ **prendront**$_{Pl}$...", all five defined superficial heuristic predict *singular*, while the target verb should be in plural. Therefore, a model that successfully predicts the plural form for this instance, must have learned a more abstract representation of the sentence, enabling it to track the long-distance subject-verb dependency. On the other hand, a model that relies on the surface heuristic strategies would be expected to fail on this instance. The evaluation analyses in the following studies will primarily focus on the more challenging cases (i.e. *0* and *1 heuristic* subsets), as correctly predicting the verb form in these instances would offer compelling evidence of a model's syntactic ability.

Constructions	Size (in sentences)	LSTM (# 47M)	$\mathcal{M}$ (# 126M)	$\mathcal{M}_{shallow}$ (# 49M)	$\mathcal{M}_{shared}$ (# 47M)
Subject-verb across object relative clause					
Overall	27,582	$94.3_{\pm 0.3}$	$98.9_{\pm 0.04}$	$90.8_{\pm 0.4}$	$97.8_{\pm 0.3}$
5 heuristics	14,708	$98.6_{\pm 0.1}$	$99.6_{\pm 0.05}$	$97.6_{\pm 0.2}$	$99.5_{\pm 0.1}$
4 heuristics	3,799	$95.2_{\pm 0.5}$	$99.0_{\pm 0.1}$	$92.2_{\pm 0.5}$	$97.9_{\pm 0.2}$
3 heuristics	4,189	$91.3_{\pm 0.8}$	$98.4_{\pm 0.1}$	$85.7_{\pm 0.4}$	$96.6_{\pm 0.2}$
2 heuristics	3,166	$84.8_{\pm 1.0}$	$97.7_{\pm 0.1}$	$77.0_{\pm 1.8}$	$94.5_{\pm 0.4}$
1 heuristic	1,451	$81.2_{\pm 1.8}$	$96.8_{\pm 0.1}$	$67.4_{\pm 2.1}$	$92.8_{\pm 0.3}$
0 heuristic	269	$74.7_{\pm 2.2}$	$94.1_{\pm 0.5}$	$63.9_{\pm 2.3}$	$87.0_{\pm 0.6}$
Object past participle					
Overall	68,497	$82.1_{\pm 1.1}$	$94.6_{\pm 0.2}$	$84.7_{\pm 0.7}$	$89.0_{\pm 0.3}$
5 heuristics	32,149	$95.3_{\pm 0.6}$	$99.2_{\pm 0.1}$	$96.7_{\pm 0.4}$	$98.5_{\pm 0.2}$
4 heuristics	12,711	$85.9_{\pm 1.0}$	$96.5_{\pm 0.1}$	$89.7_{\pm 0.8}$	$92.9_{\pm 0.2}$
3 heuristics	9,159	$71.9_{\pm 1.6}$	$91.6_{\pm 0.4}$	$75.0_{\pm 1.3}$	$82.8_{\pm 0.4}$
2 heuristics	10,621	$62.2_{\pm 2.4}$	$87.6_{\pm 0.4}$	$66.1_{\pm 2.1}$	$74.4_{\pm 0.5}$
1 heuristic	2,870	$37.4_{\pm 4.1}$	$77.9_{\pm 0.8}$	$42.5_{\pm 4.3}$	$58.6_{\pm 2.3}$
0 heuristic	987	$40.2_{\pm 2.7}$	$76.1_{\pm 1.0}$	$44.2_{\pm 3.1}$	$56.0_{\pm 2.1}$

Table 4.4: Accuracies(%) achieved by LSTM and Transformer models as a function of the agreement prediction difficulty. Transformer model $\mathcal{M}$ uses 16-layer decoders each with 16 heads, $\mathcal{M}_{shallow}$ has 2 layers each with 16 heads and $\mathcal{M}_{shared}$ is a variant of $\mathcal{M}$ using shared parameters across all 16 layers.

Results The breakdown of model performance based on the heuristic-based protocol, as shown in Table 4.4, reveals more nuanced results. With respect to the type of agreement, we observe that both LSTM and Transformer models achieve much better performance in S-V agreement compared to the O-PP agreement, especially in the most challenging cases (i.e. *0*

& 1 heuristic subsets). This is despite the fact that the linear distance between the **cue** and the **target** in the subject-verb dependency is, on average, twice as long as that in the O-PP agreement (11 tokens vs. 5 tokens). One possible explanation for this performance difference is the frequency of agreement patterns in the training data. Subject-verb agreement occurs in nearly every sentence of the training data, while only 0.35% of the training sentences involve an object-past participle agreement.[13] However, we do find a consistent pattern across both agreement tasks: model performance always decreases as the task difficulty increases. This is particularly evident for O-PP agreement, where models show over 95% accuracy in the simplest cases (*5-heuristic* subset), but see a sharp decline in performance with increasing task difficulty. In the most difficult cases (*0-heuristic* subset), *LSTM* and $\mathcal{M}_{shallow}$ achieve below 45% accuracy. Furthermore, these observations show that the impact of surface heuristics is not limited to a relatively infrequent and complex agreement, but also extends to more frequent subject-verb agreement. This underscores the need for cautious interpretation of results on long-distance agreement tasks, and surface heuristics should be taken into account when evaluating model performance.

Regarding the model architecture, the Transformer model $\mathcal{M}$ with the largest number of parameters consistently achieves the best performance across all subsets. For both types of agreement, $\mathcal{M}$ predicts the correct verb form most of the time, even in the hardest cases where the LSTM and the shallow Transformer struggle.[14] When comparing models with a similar number of parameters as LSTM, the shallow Transformer with two layers performs worse than the LSTM on the S-V agreement task, but slightly better on O-PP agreement. Interestingly, $\mathcal{M}_{shared}$, lightweight Transformer with shared parameters across all 16 layers, performs significantly better than the LSTM on both agreement tasks, especially in the harder subsets. This performance trend aligns with the model perplexity scores, echoing the findings of Dehghani et al. (2018) on Universal Transformers that the depth of the Transformer architecture is crucial for structure-sensitive tasks.

Above all, this comparison highlights the remarkable ability of Transformers to capture syntactic information that even the LSTM, a robust baseline upon which many conclusions about the syntactic capacity of neural networks have been drawn, struggles to capture. The Transformer model, $\mathcal{M}$, is able to generalize beyond superficial heuristics on long-distance agreement tasks, suggesting that it can extract certain abstract generalizations.

[13] See Section 4.2.5 for detailed analysis of frequency effects

[14] This is not to say that the Transformer model acquire perfect rule-based syntactic competence. It still struggles with complex structures such as subject inversion and nested clauses. For a qualitative analysis of error patterns, please refer to Section A.2.1 in the Appendix.

4.2.5 Control experiments

In the previous section, our experiments using the heuristic-based evaluation protocol revealed that the Transformer LM is able to abstract away from superficial heuristics. To corroborate these findings and address some known pitfalls of the agreement task (§3), we perform a set of control experiments and analyses, focusing on the impact of semantic cues, frequency effects, and the choice of evaluation metrics.

Experiment 1: Impact of semantic or collocational information

A well-known confounding factor in syntactic evaluation tests is the influence of semantic or collocational information. To investigate this, we first perform an exploratory analysis to assess the impact of collocational information on model performance in agreement tasks. Specifically, we examine the performance of models on **cue-target** pairs that either co-occurred or never appeared together in the same sentence during training. An evaluation sentence is considered as 'unseen' if the **cue-target** pair never appeared in the same sentence during training. Table 4.5 shows that both models significantly outperform a baseline that simply predicts the more frequently observed cue-target pairs. However, both models show a decrease in performance on the unseen subset: the Transformer exhibits a drop of less than 5%, the LSTM experiences a decrease of over 10%. This observation indicates that while both models are somewhat influenced by collocational information, the Transformer exhibits greater robustness when encountering unseen cue-target combinations.

	S-V		O-PP	
	Seen	Unseen	Seen	Unseen
Transformer	$99.0\%_{\pm 0.1}$	$98.4\%_{\pm 0.1}$	$95.7\%_{\pm 0.2}$	$90.8\%_{\pm 0.4}$
LSTM	$96.2\%_{\pm 0.3}$	$84.5\%_{\pm 0.9}$	$84.8\%_{\pm 0.9}$	$72.0\%_{\pm 2.4}$
Argmax_v(cue-target)	$75.2\%_{\pm 0.0}$	$38.3\%_{\pm 0.0}$	$83.1\%_{\pm 0.0}$	$37.7\%_{\pm 0.0}$

Table 4.5: Accuracy breakdown based on whether the **cue-target** pair was seen (occurrence > 0) or unseen (occurrence = 0) during pre-training. The baseline $\texttt{argmax}_v$**(cue-target)** consistently predicts the more frequently observed pairs. If both the target and competing pairs were unseen, this baseline model randomly selects one pair.

To further investigate the impact of collocational information beyond the **cue-target** pair and consider the overall context of target constructions, we adopt the method introduced by Gulordava et al. (2018) (§3). This approach aims to assess models' syntactic abilities in the absence of meaningful semantics or collocational cues, by transforming the original evaluation set into a nonsensical but syntactically well-formed evaluation set, which we refer to as the **Nonce** set. For each original sentence, we generate three nonsensical sen-

tences by replacing every content word with a random word that shares the same POS and morphological features.[15] Below is an example of a nonsensical sentence (16b) generated from its original counterpart (16a):

(16) Original sentence → nonce sentence:

a. ORIGINAL: Les **offres** que le directeur a **acceptées** __
*The offers_**Pl** that the director has accepted_**Pl** __*

b. NONCE:
Les **omellettes** que le professeur a **attachées** __
*The omelettes_**Pl** that the professor has attached_**Pl** __*

During the substitution procedure, we excluded word forms that appeared in the treebank with more than one PoS annotations to ensure that all randomly selected words have unambiguous PoS. For example, *données* is not a suitable random word candidate because it can be a plural noun (data) or the plural past participle of the verb *donner* (to give). To maintain argument structure constraints, the target verb can only be replaced by another random transitive word, and all function words (e.g., prepositions, conjunctions, ...) and punctuation remain unchanged. Consequently, the **Nonce** set preserves the grammatical syntax of the original sentences, but is highly semantically implausible.

Using the evaluation metric described in Section (§4.2.4), we also evaluate the syntactic abilities of the models in the **Nonce** set for the same number agreement tasks. Figure 4.4 reports models performance on the **Nonce** set compared to the original set. Overall, both architectures exhibit only a mild degradation in accuracy relative to the *original* setting across the two agreement tasks. Interestingly, the extent of performance degradation seems to correlate with the complexity of the agreement prediction task. As sentences become more abstract, semantic cues appear to have a greater impact on model decisions. Specifically, for the most challenging S-V agreement subset, the Transformer's accuracy drops by 11.6 percentage points, while the LSTM's drops by 16.7 points. For the most difficult O-PP agreement subset, the declines are 6.9 and 4.4 points for the Transformer and LSTM, respectively.[16] Across both agreement tasks, the observed drops in accuracy are similar in scale to what has been reported in prior studies by Gulordava et al. (2018) and Goldberg (2019), suggesting that semantic or collocational confounds have only a moderate impact on our models' performance in agreement prediction tasks.[17] It further implies that our models primarily rely on syntactic information to determine the correct form of the verb.

[15]The random words were selected from the version 2.7 of the Universal Dependency French treebanks
[16]For detailed scores, please refer to Table A.5 in the Appendix.
[17]See Section A.2.2 in the Appendix for an analysis on lexical variation in the results.

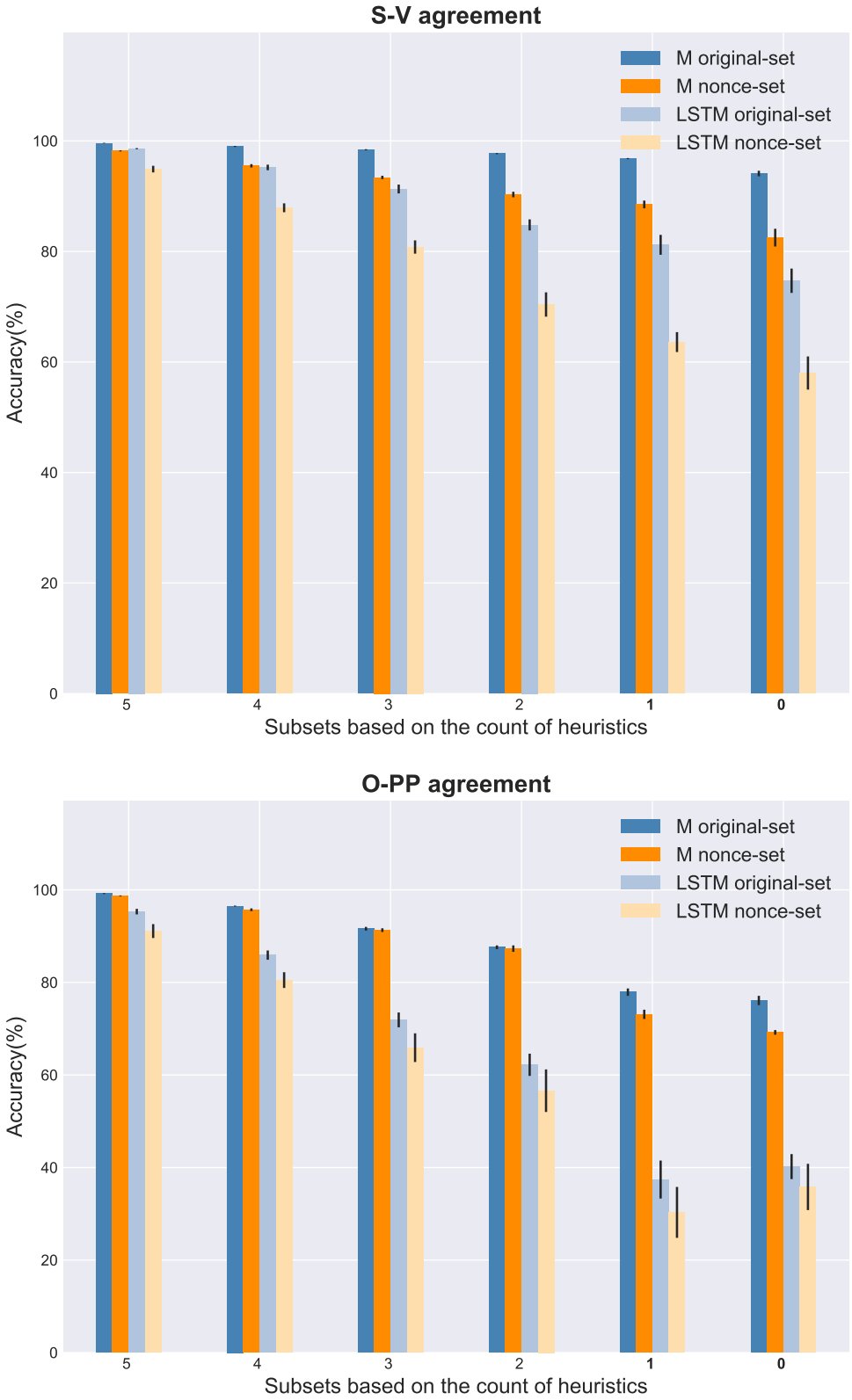


Figure 4.4: Average accuracy of LSTM (indicated by lighter color bars) and Transformer models on the *Nonce set*, represented by orange bars, and the *Original set*, indicated by blue bars.

Experiment 2: Impact of frequency effects

In French, as in many other languages, there is a considerable frequency disparity between singular and plural verbs. In written French, singular third-person verbs are observed to be five to ten times more common than their plural equivalents (Ågren and Van de Weijer, 2013). Such frequency effects, which can influence various levels of human language processing (Marantz, 2013), tend to reduce errors when higher-frequency forms are the target and induce errors when a competing lower-frequency form is the target (Ambridge et al., 2015). Empirical studies on agreement tasks involving human subjects reflect this trend. For instance, Villata (2017) observed that French speakers tend to produce more correct agreement for the O-PP agreement when the target is singular. This trend is often attributed to human's general bias towards the production of default singular forms (Greenberg et al., 1963; Corbett and Fraser, 2000). These findings prompt us to investigate further: Do neural language models exhibit similar biases as humans in number agreement tasks? To what extent are the decisions made by these models a reflection of the frequency distributions encountered during training?

	S-V		O-PP	
	Singular	Plural	Singular	Plural
Transformer	$99.4_{\pm 0.05}$	$97.8_{\pm 0.1}$	$99.2_{\pm 0.1}$	$86.2_{\pm 0.4}$
LSTM	$98.0_{\pm 0.3}$	$85.9_{\pm 1.5}$	$95.4_{\pm 0.7}$	$57.2_{\pm 2.9}$
Argmax_v(target)	$99.3_{\pm 0.0}$	$0.5_{\pm 0.0}$	$93.2_{\pm 0.0}$	$8.6_{\pm 0.0}$

Table 4.6: Accuracy breakdown based on the grammatical number of the **target**. The baseline $\texttt{argmax}_v\texttt{(target)}$ consistently predicts the more frequently observed number of the **target**.

As shown in Table 4.6, our further breakdown of the experimental results in Section 4.2 reveals consistent trends across both agreement tasks. Transformer and LSTM achieve over 95% accuracy in singular conditions, but show consistent lower performance in plural conditions, suggesting a model bias towards singular forms under our current evaluation metric. Additionally, this performance disparity correlates with the frequency ratio of the target form to the competing form in the pre-training data, as shown in Figure 4.5. This echoes the findings of Ambridge et al. (2015) that higher-frequency forms as targets tend to reduce errors. Interestingly, even though the frequency ratios for the singular class in S-V and O-PP agreements are similar, and a similar trend exists for plurals across both tasks (Figure 4.5), the performance gap between the plural and its corresponding singular is less pronounced in S-V agreement than in O-PP agreement (Table 4.6).

In the case of S-V agreement, the Transformer model shows only a minor drop of 1.6

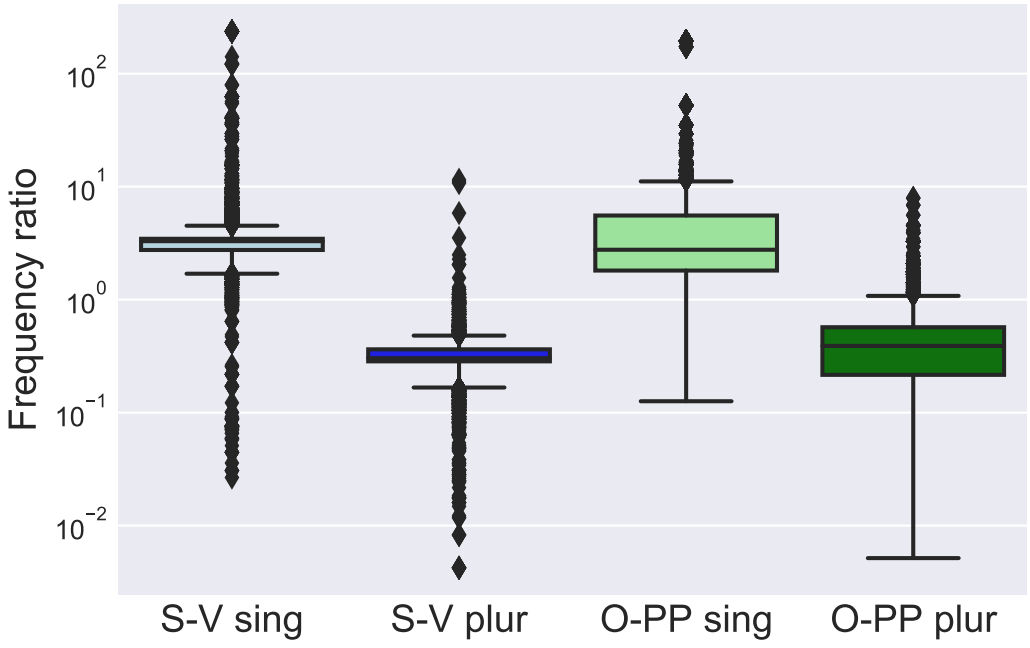


Figure 4.5: Frequency ratio of target form to competing form. For instance, for the *S-V sing* condition, a ratio of 10^1 indicates that the target verb form (singular) occurs 10 times more frequently in the pretraining data than its competing form (plural).

percentage points in accuracy for plural conditions, compared to the near-perfect accuracy with singular forms. This contrasts sharply with the near-zero accuracy of the heuristic baseline, demonstrating the Transformer's consistent preference for less frequent but grammatically correct verb forms over more frequently occurring forms. These results suggest that Transformer generally applies the subject-verb agreement rules with high accuracy, even when faced with a strong frequency bias.

In contrast, the models' performance in O-PP agreement exhibits a significant difference between singular and plural conditions. Specifically, Transformer models experience a 13% drop in accuracy for plural forms, while LSTMs see a more substantial decrease of over 38%. This suggests that both types of model struggle more with plural forms in O-PP agreement compared to S-V agreement. Given that the O-PP agreement is a relatively rare syntactic phenomenon compared to the S-V agreement, as discussed in §4.2.4, this could partially explain the lower overall accuracy of the model in the former task. Additionally, we observe a marked discrepancy in the absolute frequency of **target** verbs between S-V agreement and O-PP agreement. As depicted in Figure 4.6, target verbs in S-V agreement are much more frequent in models' pretraining data compared to those in O-PP agreement. For plural

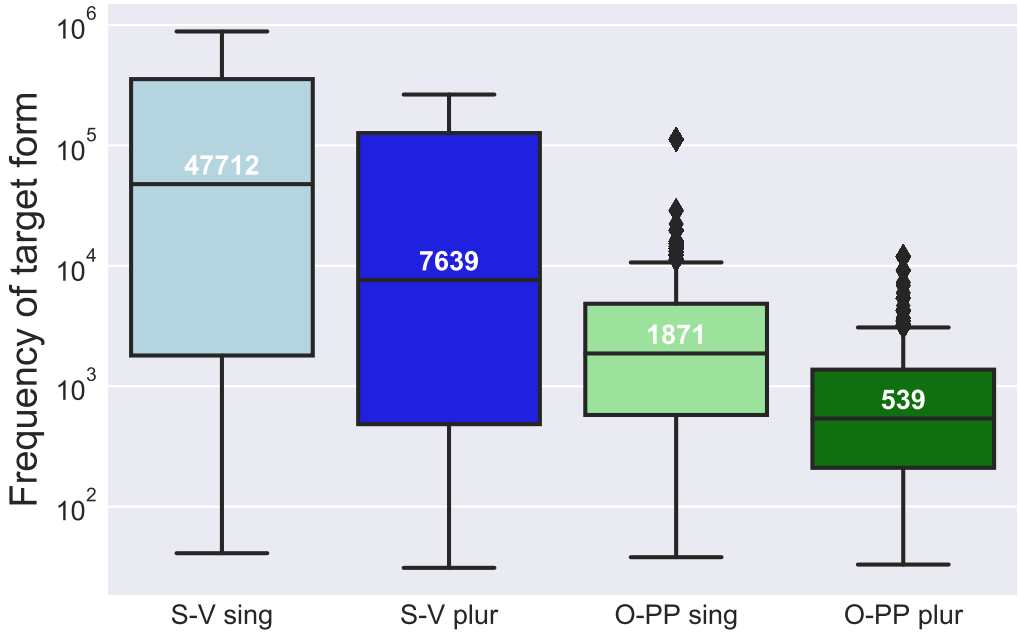


Figure 4.6: Absolute frequency of **target** verbs in pre-training data, with medians displayed in white numbers

forms, our results align with Wei et al. (2021), indicating that more frequent target verbs are more likely to be correctly inflected in number agreement tasks. This is consistent with the frequency effects observed in human language processing. However, for singular forms, despite the absolute training frequency discrepancy of **target** forms across the two agreement tasks, models show equivalently high performance on singular conditions for both tasks. This suggests that, similar to humans, neural language models, may also default to using singular forms when handling number agreement tasks in French, corroborating previous research on default reasoning in language models (Jumelet et al., 2019). Notably, the Transformer model appears to be capable of effectively leveraging syntactic structures to override this default reasoning in plural conditions, as evidenced by its strong performance in both singular and plural agreement tasks.

While frequency effects could partially account for the observed asymmetry between singular and plural, task complexity also appears to play a significant role in models' lower performance in plural conditions. Further analysis of our evaluation sets reveals a consistent correlation between class distribution and task difficulty (measured by the count of heuristics; §4.2.4). In the easiest cases, where any of the five heuristics can solve the task (*5-heuristic* subset), singular target forms predominate, accounting for 94% of the cases in

O-PP agreement and 91% in S-V agreement. In contrast, in the most challenging cases where no heuristic allows to predict the agreement (*0-heuristic* subset), the plural class becomes the dominant category (O-PP: 99%, S-V: 96%). This correlation holds true for both O-PP and S-V agreement. It suggests that in natural corpora, plural verbs tend to appear in more complex and potentially confounding long-distance agreement contexts compared to their singular counterparts. This further explains the models' lower performance in plural conditions. The reasons behind this empirical observation remain elusive. We intend to delve deeper into its implications in future research, using controlled experiments that account for syntactic complexity and class balance.

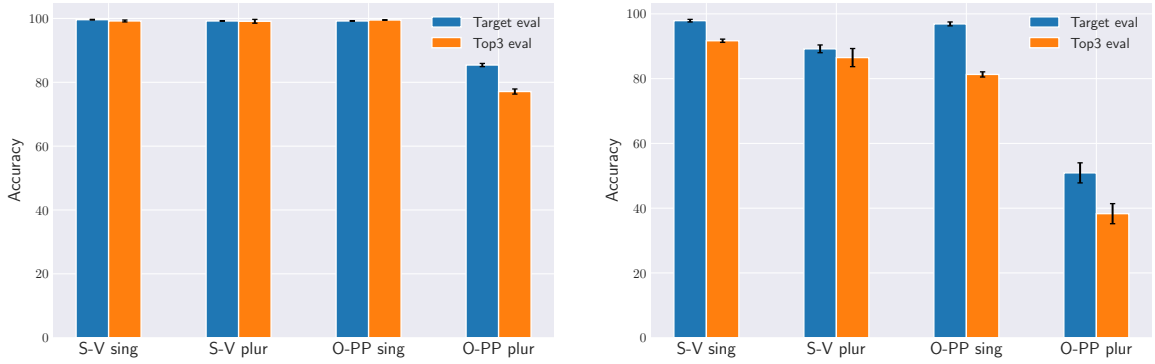
In summary, our analysis reveals that both Transformer and LSTM models consistently exhibit better performance in singular conditions than in plural ones across two agreement tasks. This trend suggests that these models might possess a frequency-driven bias in number agreement tasks, similar to that observed in humans. Notably, the Transformer model is better at mitigating this singular bias when processing plural conditions, highlighting its ability to leverage syntactic structures. The observed performance asymmetry could arise from several factors, including the higher frequency of singular verbs in the French language, the imbalanced distribution of grammatical number among syntactic constructions of varying complexity, and potential distinctions in how the models encode singularity and plurality, as suggested by Jumelet et al. (2019).

Experiment 3: Top-k evaluation metric

The evaluation metric we have adopted, which aligns with common practices in the literature (§3), may introduce its own set of biases into our assessment. Specifically, our metric focuses on the model's ability to discriminate between the singular and plural forms of a target verb that naturally occur in a corpus. This approach does not necessarily capture the model's ability to generate a verb form that is both contextually appropriate and correctly inflected for number. This limitation has been noted in previous work, such as the study by Newman et al. (2021), which observed that language models perform better on verbs they predict to be contextually likely. These considerations raise an important question regarding whether our evaluation metric faithfully reflects the models' likely behavior. In other words, do the most likely words, predicted by the models given a sentence prefix, exhibit consistent agreement features as those obtained from our *target verb* evaluation metric?

To address this question, we propose an alternative evaluation metric, referred to as the *top*-3 evaluation metric, to better measure models likely behavior. Instead of comparing the probabilities of two forms given an evaluation sentence prefix, we focus on the words

the models consider most likely to occur. Specifically, we sample the top ten most probable word predictions made by the models for a given sentence prefix. From these words, we use the morphologizer of a pre-trained French model in *spaCy* (Honnibal et al., 2020),[18] to get the three most probable verbs. We then consider the majority number expressed among these three verbs as the models' agreement prediction for that sentence. To ensure a fair comparison between our initial evaluation metric and this *top*-3 evaluation metric, we exclude sentences where the top ten most probable word predictions from the models do not contain any verbs.[19]



(a) **Transformer** language model (b) **LSTM** language model

Figure 4.7: Comparison of models' accuracy in two agreement tasks using *top*-3 evaluation metric (orange bars) and *target verb* evaluation metric (blue bars).

As shown in Figures 4.7a and 4.7b, the trends observed with the *top*-3 evaluation metric closely align with those of the *target verb* evaluation metric. For both Transformer and LSTM, the two metrics demonstrate a persistent performance asymmetry between singular and plural conditions in both agreement tasks. The performance breakdown based on task difficulty (Table A.6 and A.7 in the Appendix) also indicates that all models performance decrease with the task difficulty. Notably, for the Transformer, a high level of consistency is observed between the two metrics, with an inter-agreement rate surpassing 92% in both tasks (see Table A.8 in the Appendix for full results).

For the Transformer model, the two evaluation metrics yield similar scores for the S-V agreement and the singular condition of O-PP agreement. However, a decrease of 8.3% is observed in the plural condition of the O-PP agreement when using the *top*-3 metric. This decrease may be partially attributed to the presence of part-of-speech ambiguous words (e.g.,

[18] `https://spacy.io/models/fr`, we used the model: *fr_dep_news_trf*

[19] For the Transformer model, we excluded 7.9% (resp. 29.8% for LSTM) of the total evaluation sentences (27,582) in the S-V agreement. Similarly, in the O-PP agreement task, 0.3% of the total evaluation sentences (68,497) were excluded for the Transformer, compared to 45.7% for the LSTM. Among the excluded sentences, the top ten LSTM predictions are mainly punctuation, prepositions, articles, or nouns.

“données" can be both a noun and a verb) or number ambiguous words (e.g., “appris" can be both a singular and plural form). In contrast, the strong baseline model, LSTM, consistently exhibits lower accuracy across the board when evaluated with the top-3 metric. Additionally, for both agreement tasks, over 29% of the top ten predictions of LSTM (compared to less than 8% for Transformers) do not contain any verbs. This finding indicates that the Transformer model demonstrates similar behavior under the two evaluation metrics and exhibits more robust syntactic behavior compared to the baseline LSTM model.

Given the similar performance trends of the Transformer on both agreement tasks across two evaluation metrics, we consider the differences between these metrics minor enough to proceed with the main objective of our study: investigating how the Transformer LM represents syntactic structures when handling two superficially similar long-distance agreement phenomena. Additionally, we want to avoid the potential noise introduced by the top-3 evaluation metric, which relies on a pre-trained model to predict morphological features. Therefore, all subsequent experiments are conducted using the *target verb* evaluation metric, aligning with the common practice in the literature (§3).

Interestingly, our findings based on naturalistic corpora diverge from the observations of Newman et al. (2021). While their study, using synthetic data, suggests that models perform better on verbs predicted to be the most likely in context, we did not observe this improvement in our study. Future research could further investigate the contributing factors to the observed performance asymmetry between singular and plural forms. This could involve artificially manipulating the relative frequency of singular and plural nouns within different constructions in the model's training data to better understand their influence on performance.

4.2.6 Conclusion

In this section, we evaluated the autoregressive Transformer's ability to process two syntax-sensitive phenomena in French, using number agreement tasks. Our initial experiments demonstrate strong overall performance for both types of agreement, indicating that the model's behavior aligns closely with human language use. Furthermore, we investigated the impact of surface heuristics and other known confounding factors on the model's performance. These findings lend further support to existing research (Section 2.3), confirming that the Transformer model exhibits a robust capability to capture and generalize syntactic information beyond surface heuristics and semantic or collocational cues.

We observed that while the Transformer does show some sensitivity to frequency effects, it generally displays a consistent preference for grammatically correct forms, effectively

overcoming strong biases present in the training data. Compared to the LSTM, a strong baseline in the literature for evaluating syntactic abilities of NLMs (§3), the Transformer is less influenced by surface heuristics and frequency effects, and exhibits more consistent performance under different evaluation metrics. These findings suggest that the Transformer meets the first criterion — behavioral-level similarity — for genuine syntactic generalization.

Additionally, it is important to recognize that humans also make agreement errors (Bock and Miller, 1991), and also display a bias favoring singular forms, leading to fewer agreement errors for these forms in French object-past participle agreement tasks, as observed by Villata (2017). In this context, the heuristics that affect the model's performance might hold relevance not just for the domain of artificial neural networks, but could also offer valuable insights into the study of human linguistic abilities. Specifically, these heuristics could stimulate the development of testable hypotheses for experiments aimed at understanding human syntactic performance. Our heuristic-based approach for crafting evaluation sets could help to build stimuli that effectively measure human capacity for rule-based linguistic generalization.

4.3 Locating syntactic information in Transformer language model

The experiments presented in the previous section demonstrate that the Transformer language model consistently outperforms the strong baseline model, LSTM, in long-range subject-verb and object-past participle agreements. Crucially, Transformer is able to abstract away from potential confounds such as lexical co-occurrences or superficial heuristics. This successful behavioral assessment allows us to delve deeper to evaluate the Transformer's representational adequacy as a model that helps to explain human syntactic processing. In light of this success, which aligns with prior research indicating that Transformers capture a "substantial amount" of syntactic information (§2.3), two questions emerge naturally: First, where is this syntactic information located within the Transformer's internal representations? Second, given the superficial similarities between the two types of agreement tasks we studied, does the Transformer model use a uniform internal representation for both, or are there distinct representations that reflect the theoretical nuances of each task?

In this section, focusing on Transformer LM, we investigate the question of **where** syntactic information is encoded from two perspectives.[20] First, in Section 4.3.1, we use probing classifiers, detailed earlier in Section 2.3.2, to identify token positions where the

[20]Datasets and code: `https://gitlab.huma-num.fr/bli/syntactic-info-distribution`

agreement information is encoded within the model's internal representation. This analysis enables us to localize the agreement feature across token representations within a sentence. Second, in Section 4.3.2, we use a feature selection method associated with probing to identify the specific subspace within the Transformer's representations that encodes the relevant agreement information.

4.3.1 Distribution of syntactic agreement information across token positions

In this section, we investigate where the Transformer encodes the syntactic information necessary for predicting the correct *target* form in the two types of agreement tasks. Specifically, we explore whether this agreement information is distributed across all tokens following the *cue* in the sentence, as theoretically allowed by the self-attention mechanism and observed by Klafka and Ettinger (2020). Alternatively, is this information encoded more locally, centered around the **cue** and **target** tokens, as predicted by the specific agreement rules?

To investigate these hypotheses, we use probing classifiers following the approach of Giulianelli et al. (2018) (§3). In our study, we denote the representation generated by a Transformer LM for the token t at layer l by:

$$r_t = \text{Transformer}_l(t) \tag{4.6}$$

Given an evaluation sentence, our goal is to examine whether the representation of a token t within the sentence contains the relevant syntactic agreement feature, denoted as $\mathcal{A}$, which corresponds to the number of the **cue** (either 'Singular' or 'Plural'). To achieve this, we train a classifier defined as a function $\mathcal{C}$ that maps the representation of each token to the agreement feature of the sentence, $\mathcal{A}$:

$$\mathcal{C} : r_t \rightsquigarrow \mathcal{A}, \text{with } \mathcal{A} \in \{\text{Singular}, \text{Plural}\} \tag{4.7}$$

The core assumption underlying this approach is that if the Transformer has encoded syntactic agreement information within its representation space, then a probing classifier should be able to "extract" this information from the corresponding token representations produced by the Transformer (§2.3.2). In this study, we use a logistic regression classifier,[21] defined as:

$$\sigma(\theta^T r_t + b) \rightsquigarrow \mathcal{A}, \text{with } r_t \in \mathbb{R}^{768} \tag{4.8}$$

[21]This choice follows the recommendation of Hewitt and Liang (2019), who found that non-linear probes tend to memorize the probing task by leveraging surface pattern recognition, rather than relying on the information captured in the representations of the probed model.

Here, r_t is the token representation extracted from the Transformer's last layer (i.e., the 16th layer in our experiments) for the token t, σ denoting the sigmoid function. The parameters vector (i.e., coefficients) is represented by θ, and b represents the bias term.

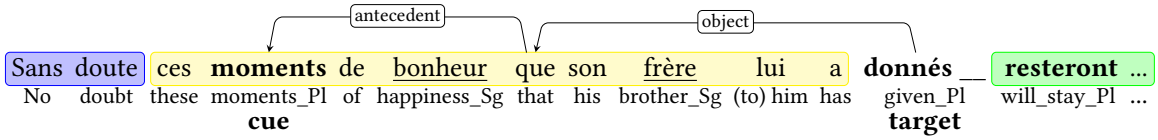


Figure 4.8: For the O-PP agreement, the prefix is highlighted in blue, the context in yellow and the suffix in green.

Training To train the probing classifiers, we construct a training set $\mathcal{D} = \{x^{(i)}, \mathcal{A}^{(i)}\}$ as follows. For each token in the sentences of our evaluation set, we extract its representations from the last layer of the Transformer and associate it with a label $\mathcal{A}^{(i)} \in \{\text{Singular}, \text{Plural}\}$ indicating the number of the **cue**. Next, as illustrated in Figure 4.8, we divide each sentence into three parts:

- *prefix*: words before the **cue** and its dependent words;
- *context*: words from the **cue** (and its dependent words) to just before the **target**;
- *suffix*: words following the **target**

We train individual probing classifiers for each category of word within each part of the sentences. This approach allows each classifier to specialize in PoS-specific representations of long-distance agreement information. To ensure fair comparison across sentence parts, we exclude tokens with PoS tags that occur less than 100 times, namely SYM, SCONJ, INTJ, PART, and X. This results in a total of 11 token categories in each sentence part, giving us $11 * 3$ probing classifiers.

For training and evaluation, we split the examples into 80% training data and 20% evaluation data. Each classifier is trained using three different train/test splits.[22] The averaged results are reported in Table 4.7, and more detailed results per word category are provided in Figure A.1 of the Appendix.

Results The average accuracy achieved by our probes on different parts of the sentence is presented in Table 4.7. We observe a similar pattern for both O-PP and S-V agreement: the syntactic agreement information about the number of the **cue** is essentially encoded

[22] All classifiers were implemented with the `scikit-Learn` library Pedregosa et al. (2011). A grid search with 5-fold cross-validation was performed to select the optimal value of the regularization parameter C. The `max_iter` parameter was set to 1,000 during the training process. Random_state = 0, 20 and 42

	Mean probing Accuracy	
	O-PP agreement	S-V agreement
prefix	$58.6\%_{\pm 0.1}$	$59.5\%_{\pm 0.2}$
context	$92.3\%_{\pm 0.2}$	$93.0\%_{\pm 0.1}$
suffix	$73.6\%_{\pm 0.2}$	$78.1\%_{\pm 0.2}$

Table 4.7: Probing results across different sentence parts (see Figure 4.1). The reported scores represent the average accuracy of all PoS-based classifiers for each sentence segment.

within the tokens of the *context*. It is not distributed across all tokens following the cue in the sentence.

As expected, in both tasks, the probe performance on the *prefix* is very low. Given the autoregressive nature of the model, token representations in the *prefix* cannot attend to the **cue**, and thus, cannot encode its number information. The accuracy observed on the *prefix* mainly reflects the difference in prior probabilities of the two grammatical number classes within the evaluation set.[23] In contrast, when using tokens from the *context* as input features, the probe accuracy is consistently high for both agreement types. However, the accuracy significantly drops for the *suffix* tokens, though it remains higher than that observed for the *prefix*. This suggests that the information required to predict the correct *target* form is distributed across all tokens between the *cue* (where the number of the *target* verb is specified) and the *target* (where this information is being 'used'). This finding, to some extent, challenges the observations of Wisniewski et al. (2021), who discovered that gender information in a neural translation system is distributed throughout the source and target representations. However, it should be noted that their study focused on a different type of information and was limited to sentences with a simple structure.

Results so-far indicate that the agreement information related to **cue** is mainly distributed across all tokens in the *context* part of the sentences. To gain a more precise understanding of how the Transformer model tracks this agreement information from **cue** to **target**, we conduct an experiment that focused on a specific sentence pattern with a fixed six-word *context*. Specifically, we focus on sentences where **cue** is separated from the relative pronoun only by a prepositional phrase. This pattern applies to sentences such as the one shown in (17) for long-distance subject-verb agreement, and (18) for object-past participle agreement.

(17)

Sentence:	...	**bureau·x**	en	métal	qu'	il	aime	**coût·ent** ...
	...	desks	Prep.	metal	that	he	loves	cost ...
Pattern:	...	`Subject`	`ADP`	`NOUN`	`que`	`PRON`	`V`	`target V`...

[23] As discussed in §4.2.2, within the two evaluation sets, 65% of the target past participles in O-PP agreement are singular, while 70% of the target verbs in S-V agreement are singular.

(18)

Sentence:	...	**bureau·x**	en	métal	qu'	il	a	**trouvé·s** ...
	...	desks	Prep.	metal	that	he	has	found_Pl ...
Pattern:	...	`Antecedent`	`ADP`	`NOUN`	`que`	`PRON`	`AUX`	`target PP` ...

Training To examine the distribution of the agreement information between the *cue* and the *target*, we build a dataset for each agreement phenomenon, following the previously defined patterns with a fixed six-word *context*. Each position within the *context* is labeled with the corresponding PoS tag of the tokens, as illustrated in (17) and (18). Additionally, we also consider the five tokens before and after the six-word *context* window, denoted as b_i (for tokens before) and a_i (for tokens after), where `i` represents the position relative to the pattern, as illustrated by the X-axis labels in Figure 4.9.

For the training set, we randomly sample 800 examples for each agreement phenomenon, ensuring a balance between singular and plural forms. For the test set, we sample a balanced set of 200 examples, with 100 sentences where the embedded noun (at the 'NOUN' position within the *context*) is an *attractor*, and 100 sentences where this noun has the same number as the *cue*.[24] Unlike the previous experiment where we trained probing classifiers on representations of all words in the sentence based on their word categories and their location in the sentence, in this experiment, we train distinct classifiers for each position within the defined scope of sentences.

Results We plot in Figure 4.9 the average probing accuracy at different positions of the specific construction for both agreement phenomena. The results show a consistent pattern: the accuracy of the probes is initially low in the *prefix* (i.e., b-positions) but starts to increase from the position just before the *cue*. This position often corresponds to determiners or adjectives that need to agree in number with the *cue*. As we move into the *context*, the accuracy stabilizes, with probes achieving very high accuracy, even at the attractor position. The accuracy then drops sharply after passing the *target*, especially when an attractor is present in the *context*. It appears that once the *target* has been encountered and the number information of the *cue* is no longer relevant, subsequent tokens no longer encode it. This trend is consistently observed for both types of agreement phenomena.

Unsurprisingly, the probes achieve perfect scores at and just after the *cue* positions, as well as at the position immediately following the *target*. This suggests that the Transformer has learned to recognize and store the number information of the *cue* and *target* in its internal representations. And this information is encoded in a linearly extractable manner.

[24]We conducted the sampling process using three different seeds, and for each sampling, we performed three train/test splits. The reported scores are averaged across all splits.

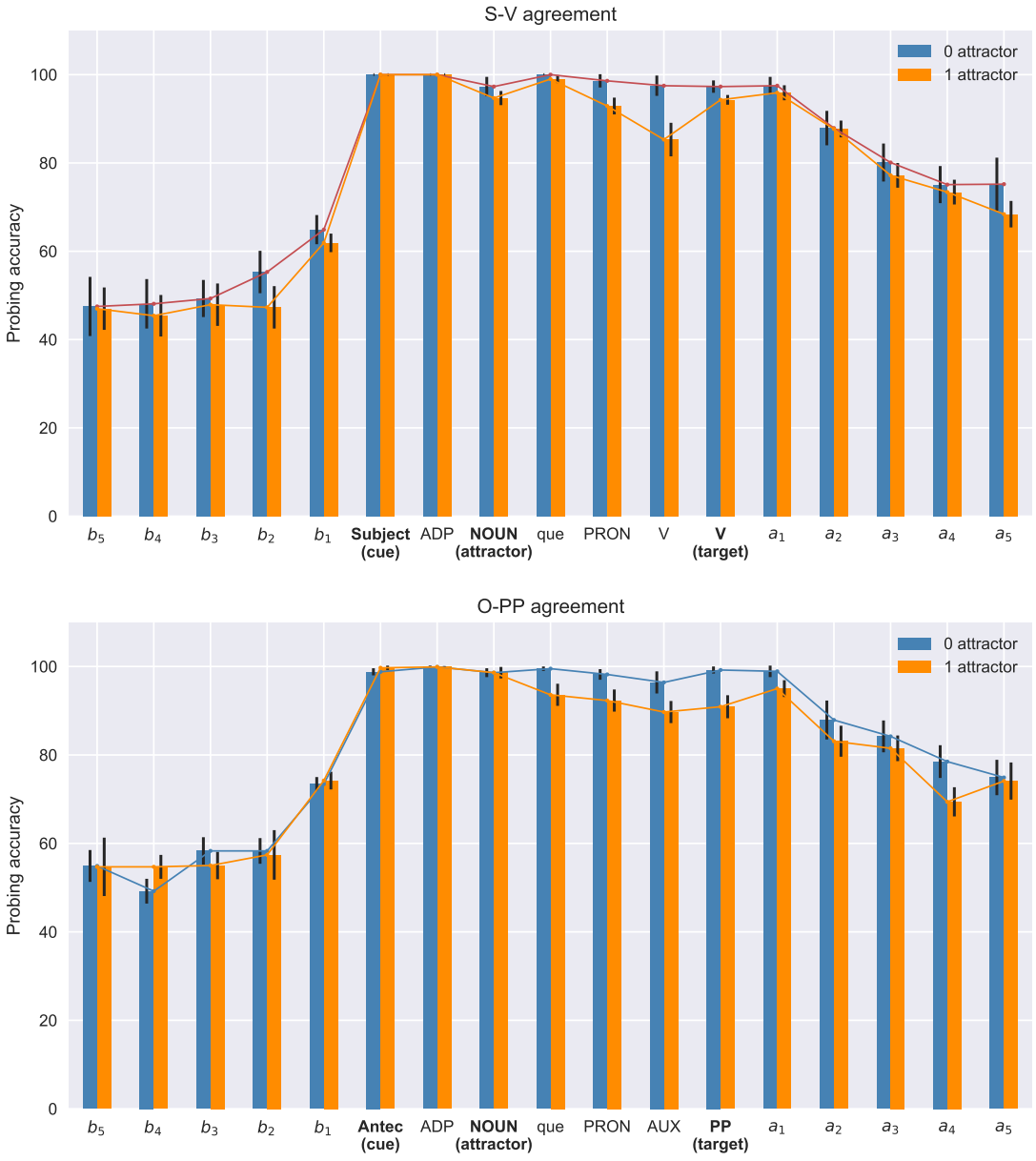


Figure 4.9: Average probing accuracy at each position based on the number of the *cue*. The b_i (resp. a_i) position denotes the i-th token before (resp. after) the pattern. The position labeled as 'Noun' corresponds to a noun with the opposite number as the *cue* in the 1-attractor subset, and a noun with the same number as the *cue* in the 0-attractor subset.

A particularly interesting observation arises when considering the *1-attractor* subset. The accuracy of the probes only slightly degrades at positions immediately following the attractor, which carries the opposite number to the probed grammatical number. Intriguingly, at the target position, where the agreement information from the *cue* is being used, the probe accuracy shows a reboost, in particular in S-V agreement. This suggests that the model appears to know where to pinpoint the syntactic number information, enabling it to avoid potential misleading cues. These observations suggest a coherent and robust flow of agreement information within the Transformer's representations.

4.3.2 Probing internal representations components

Our previous experiment using probing classifiers revealed that agreement information is encoded across all tokens within the *context*. In this study, we aim to determine **where** within the Transformer's representation space this information is encoded. Specifically, we want to identify which components of the token representations generated by the Transformer are most crucial for capturing this syntactic agreement information.

To achieve this, we use an ℓ_1-regularized logistic regression model, known for its tendency to produce sparse feature vectors by driving many feature coefficients, denoted by w_i in the equation (4.9), towards zero (Tibshirani, 1996; Ng, 2004). This characteristic makes it well-suited for feature selection tasks, allowing us to identify the most relevant components within the Transformer's representations that are responsible for capturing the agreement information. The ℓ_1-regularized logistic regression model follows the formulation:

$$\begin{aligned} \mathbb{P}_{\mathbf{w},b}(y = \text{Singular}|x_i) &= \sigma(\mathbf{w}^T x_i + b), \\ \text{where } \mathbf{w}^T x_i &= w_1 * x_1 + w_2 * x_2 + \ldots + w_{768} * x_{768} \end{aligned} \tag{4.9}$$

where $\mathbf{w}$ represents the parameter (i.e., coefficients) vector, x_i denotes the token representation at position i in a sentence, and y represents the grammatical number of the **cue**, which is the agreement information. This model minimizes the objective function with an ℓ_1 regularization term:

$$\sum_{i=1}^{n} - \log P(y_i|\mathbf{x}_i; \mathbf{w}) + \frac{1}{C}||\mathbf{w}||_1 \tag{4.10}$$

where, C represents the inverse of regularization strength. As the value of C increases, the number of features with non-zero coefficients w_i also increases. By varying the values of C, we can control the sparsity level of the solution $\mathbf{w}$, thereby identifying the most relevant components of the Transformer's representations for the probing tasks.

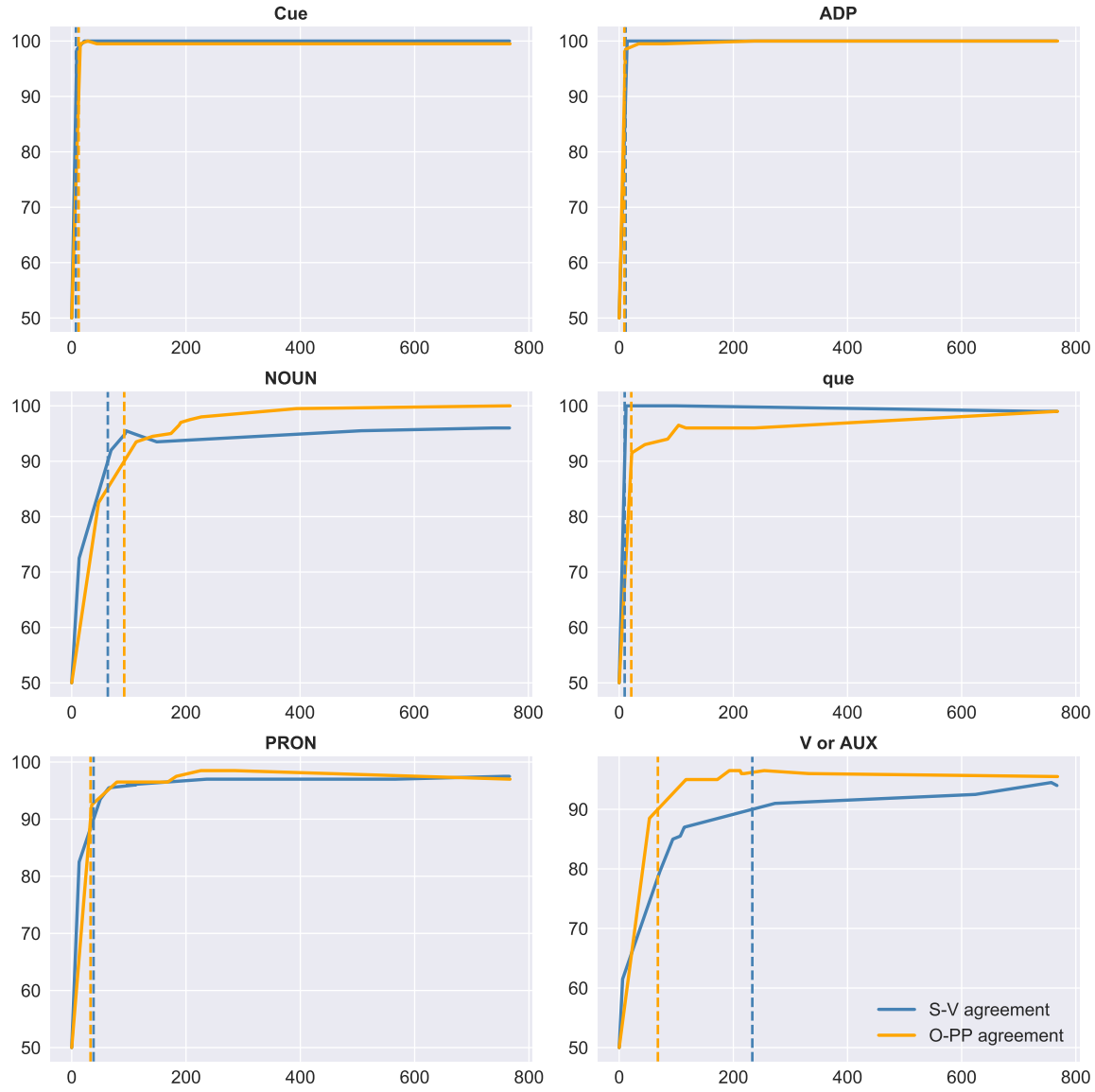


Figure 4.10: Probing accuracy as a function of the count of dimensions (for 768-dimension token representations) with non-zero coefficients, obtained through feature selection using ℓ_1 regularized logistic regression for each position within *context*. The X-axis denotes the count of non-zero coefficient dimensions, and the Y-axis represents probing accuracy. Vertical dashed lines indicate the points at which the accuracy reaches 90%.

Training In contrast to the previous experiment where we assessed probing accuracy with a fixed optimal regularization parameter C, in this study we vary the values of C in the ℓ_1-regularized logistic model as a means for feature selection. We replicate the same task from Section 4.3.1: training a classifier to map token representations to a binary label indicating the grammatical number of the **cue**. We use the same training and evaluation dataset: sentences of six-word *context* (§4.3.1). For each position within the *context*, we train a separate classifier (total of six classifiers). We first determine the lowest bound for C such that the feature coefficients are guaranteed to be non-zero.[25] C is then increased evenly on a log space to decrease the regularization strength. Finally, we compute and plot the *regularization path* of models from most to least regularized.

Results Figure 4.10 reports the regularization path of the probing classifiers for each position within the *context*. It is clear that high probing accuracy can be achieved using only a small number of dimensions in most positions. Remarkably, at the **cue** position, the probe can distinguish the grammatical number feature with just one dimension of token representations, reaching over 90% accuracy for both agreement phenomena. Moreover, 7 out of the top 9 dimensions with the most significant coefficients are shared between the two types of **cue**. This observation aligns with the observations of (Amini et al., 2023), suggesting that the Transformer's representation linearly encodes the grammatical number information of nouns within a few dimensions. We additionally found that for the ADP (immediately following **cue**) and que positions, which do not possess inherent grammatical number features, fewer than ten dimensions are required for the probing classifier to achieve an accuracy greater than 90%. These crucial dimensions differ between the two types of agreement constructions and also vary from one position to another.

Interestingly, even when the most relevant dimensions[26] identified by the feature selection process are removed from these representations, probes trained on the remaining dimensions still achieve over 90% accuracy. This holds true for both types of agreement phenomena, suggesting that the agreement information is redundantly encoded in the Transformer's representations.

4.3.3 Conclusion

In this section, we explored the encoding and location of syntactic agreement information in a Transformer language model that demonstrates strong overall performance in number

[25] We used the *l1_min_c* function in scikit-learn Pedregosa et al. (2011) to compute this lowest bound.
[26] The minimal dimensions that enable the respective probe to achieve over 90% accuracy.

agreement tasks (§4.2). Our probing experiments provided clear evidence of a localized distribution of agreement information within the *context* tokens, even though the self-attention mechanism theoretically allows this information to spread across all subsequent tokens after the **cue**. Additionally, we used a feature selection method to investigate the localization of agreement information within contextualized representations. Our findings reveal that while this information is encoded in a small number of highly correlated dimensions, it is also fuzzily encoded in a redundant way across the remaining dimensions.

The results of the probing experiments indicate that the Transformer language model encodes syntactic agreement information in a very similar way for both long-range agreements. In terms of acquired abstractions, the probing methodology does not provide evidence to suggest that the model acquires substantially different representations for each agreement phenomenon.

4.4 Right for the right reason: Exploring mechanisms of agreement computations

In the previous section, we used probing classifiers to locate the encoding of agreement information, revealing that it is primarily encoded across all token representations between the **cue** and the **target**. However, probing comes with a notable limitation as outlined by Belinkov and Glass (2019): it only reveals a correlation between the representations and the syntactic information measured by the probe, without providing insight into **whether** and **how** this information is actually involved in the model's prediction process. Consequently, the validity of conclusions drawn from probing experiments has been a subject of debate (§2.3.3).

In this section, we take inspiration from more recent work (Elazar et al., 2021; Finlayson et al., 2021; Ravfogel et al., 2021, ; i.a.) that focuses on understanding the causal relationship between the linguistic properties of interest and the model's behavior. We propose a novel causal framework for intervening in the self-attention mechanism to identify which tokens are genuinely responsible for providing the number information used by the model during the agreement resolution process. This not only contributes to our understanding of the model's inner workings but also serves to assess its representational adequacy. Specifically, we aim to examine whether the Transformer's approach to resolving S-V and O-PP agreements aligns with established linguistic theories. In light of its empirical success in behavioral assessment, this theoretical alignment can serve as the second critical requirement for using the Transformer model to offer explanatory insights into human syntactic processing.

This section is structured as follows. First, in Section 4.4.1, we present the causal framework and define the testable hypotheses. Subsequently, we describe the experimental setup and present the results in section 4.4.2. Finally, we provide an in-depth discussion, analyzing the implications of our findings, and draw conclusions in section 4.4.3.

4.4.1 The Causal Framework

This study aims to investigate the causal relationship between the Transformer model's behavior in number agreement tasks and its encoding of agreement information within its representations. Specifically, we seek to understand if the linear encoding of agreement information within the *context*, as revealed by the probing classifiers (§4.3), causally affects the Transformer's prediction for NA tasks. To address this question, we propose a causal framework inspired by the theory of causal inference (Pearl and Mackenzie, 2018). Central to our approach is the concept of causal interventions, where we modify the state of a specific variable — in this case, the encoding function to compute the token representation for the **target** — to observe the resulting effects on the system's behavior. This methodology allows us to explore counterfactual scenarios: How would the Transformer's behavior change if it were deprived of access to certain token representations, and consequently, the agreement information encoded within them? By answering this counterfactual question, we can measure the usefulness of specific information to the model's prediction and compare how the Transformer actually uses this encoded information in handling both types of agreement.

Causal intervention on self-attention computation Transformers rely on self-attention mechanism to build a contextualized representation for each token by iteratively computing (as a first approximation) the token representation as a linear combination of all previous token representations in the sentence (Figure 4.11). To investigate the causal impact of specific tokens on the model's agreement prediction at the **target** position, we propose an analysis method based on causal intervention. This method involves cutting the direct attention from the **target** position to the tokens of interest, effectively neutralizing their contribution to the construction of the **target**'s representation. For instance, in Figure 4.11, when the Transformer is predicting the **target** verb, the intervention prevents the self-attention from attending to the "que" token. This intervention enables us to build a counterfactual representation for the *target* that does not take into account the representation of the "que" token, thus removing any direct access to the agreement information encoded in the representation of the neutralized token — "que". This neutralization process thus approximates the do (•) operator in the causal inference literature.

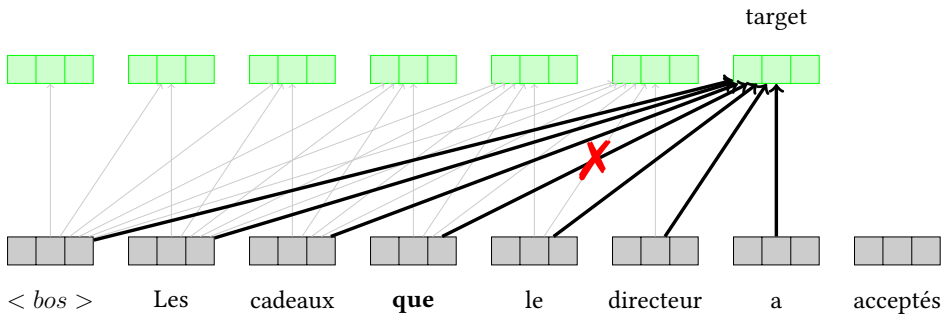


Figure 4.11: With the initial masked self-attention mechanism, the next token representation is computed as a weighted sum of all previous token representations. To assess the impact of "que" on the model's agreement behavior, the causal intervention involves cutting the direct attention from the **target** position to the token "que" (denoted by ✗), and then comparing the Transformer's prediction before and after this intervention.

By comparing the model's prediction on the agreement tasks before and after different interventions, we can assess whether the representations of one or several specific token(s) have a direct impact on the model's behavior. Table 4.8 provides an example from our evaluation set, highlighting the effect of an intervention targeting the token "que". As the intervention occurs only when the target verb is being predicted, there is no impact on the tokens preceding it (i.e., no changes in log probabilities have been observed up to the **target**). In this example, the Transformer originally assigned a higher probability to the correct plural form "accepté·s" than to the incorrect singular form "accepté". However, after the intervention, the situation is reversed, and the model prefers the (incorrect) singular form. This shift indicates that, for this specific sentence, the direct attention to the token "que" has a crucial causal impact on the model's agreement behavior.

	<bos>	Les The_Pl	cadeau**x** gifts_**Pl**	que that	le the	directeur director	a has	accepté·**s** / accepté* accepted_**Pl** / accepted_Sg*	$\mathcal{A}$
Original		-2.8	-9.5	-7.3	-1.8	-6.1	-3.9	**-5.9** / -8.3	1
Mask 'que'		-2.8	-9.5	-7.3	-1.8	-6.1	-3.9	-13.7 / **-11.9**	0

Table 4.8: Comparison of log-probabilities for each token of example sentences processed by our Transformer LM, before and after the intervention on "que". Sentences contain either the plural form of the target verb *acceptés*, or its singular form *accepté*. $\mathcal{A}$-column: 1 indicates a predicted agreement feature matching the gold label, 0 indicates no match.

Specifically, we perform interventions on self-attention across all layers and heads of our Transformer language model. As discussed in Section 2.2.2, the original attention mask

matrix, $\textsc{mask} \in \{0, -\infty\}^{n \times n}$, is defined as:

$$\textsc{mask}_{ij} = \begin{cases} -\infty & \text{if } j > i \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

The mask sets future positions (relative to the current token) to negative infinity and past positions to zero. By adding this mask to raw attention scores before applying the softmax function, future positions get an attention score of 0 — this is because the softmax of negative infinity is 0. Previous and current positions remain unchanged since adding zero does not alter their raw scores. As a result, we effectively zero out the attention scores for all positions in the future.

To implement our causal interventions, we extend the original attention mask by additionally setting the weights of specific tokens of interest to be zero. For instance, in a sentence where the position of "que" is denoted as q and the target position as t, we modify the initial attention mask, $\textsc{mask}$, by setting $\textsc{mask}^{l}_{t,q} = -\infty$ across all attention layers and heads. This effectively cuts the direct attention from the **target** position to the token "que" while keeping the rest of the self-attention mask unchanged, as shown in Figure 4.11.

We specifically aim to estimate the causal effect of direct attention from **target** to specific tokens on the model's behavior. It's worth mentioning that agreement information may not be exclusively conveyed through direct attention; intermediate tokens can also convey relevant details (Klafka and Ettinger, 2020; Lasri et al., 2022b). In Figure 4.11, for instance, the intermediate tokens between "que" and **target** continue to incorporate "que" directly into their representations. Since the representation of **target** indeed relies on the representations of all preceding unmasked tokens, the information encoded in "que" can still be indirectly considered.

Abstract causal model We now formalize the causal intervention by defining an abstract causal model as illustrated in Figure 4.12:

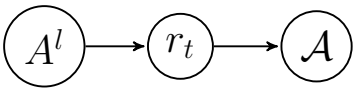


Figure 4.12: Causal model showing dependencies between the attention weights A^l at layer l, the **target**'s contextualized representation r_t and Transformer's predicated agreement feature $\mathcal{A}$.

Given a sentence prefix $S =< S_p, S_c, S_q, S_i >$ for NA tasks, we aggregate groups of tokens into the following abstract variables (as illustrated in Figure 4.13):

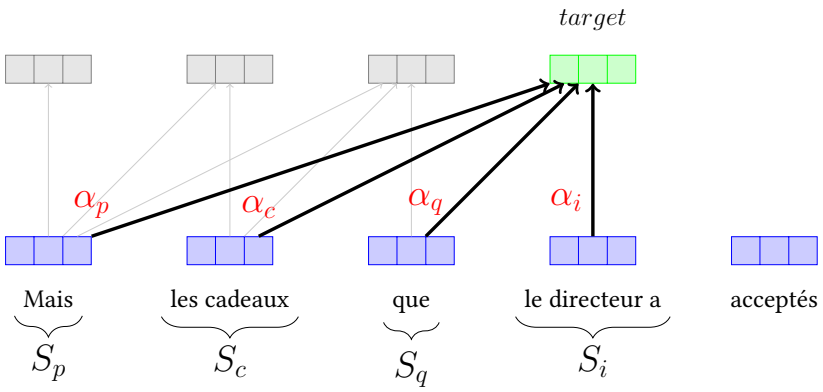


Figure 4.13: Target representation as a linear combination of all preceding token representations, weighted by their attention scores $A_t^l =< \alpha_p, \alpha_c, \alpha_q, \alpha_i >$

- S_c: **cue** and its dependent words
- S_q: relative pronoun "que"
- S_i: intermediate tokens between the **cue** and **target**, excluding those in S_c and S_q
- S_p: tokens preceding S_c

The corresponding aggregated contextualized representations and attention weights are denoted as $R =< r_p, r_c, r_q, r_i >$ and $A^l =< \alpha_p, \alpha_c, \alpha_q, \alpha_i >$, respectively. The target representation, r_t, is obtained as the output of a pre-trained Transformer LM when given the sentence prefix S as input: $r_t = \texttt{Transformer}(S)$. The causal model's outcome, denoted as $\mathcal{A} \in \{0, 1\}$, indicates if the Transformer's predicted agreement feature matches the gold label, and is defined as the output of our NA tasks $\mathcal{A} = \texttt{NA}(r_t)$.

Causal assumptions We make the following causal assumptions and formulate two types of hypotheses related to the most relevant tokens that influence the Transformer's agreement predictions. It is important to note that these hypotheses are not mutually exclusive.

- r_t is causally dependent on A^l. The contextualized representation for the **target** is computed, in a simplified view, as a linear combination of all the preceding token representations, R, weighted by the attention scores A_t^l.
  - **Linear combination hypothesis:** *r_c, r_q, r_i contribute similarly to r_t and thus affect the model's prediction for S-V and O-PP agreement in a similar way.*[27]
    Our probing experiments in Section 4.3 reveal very similar distribution patterns of agreement information across S-V and O-PP agreement: it is mainly encoded across tokens between the **cue** and the **target**.

[27] We exclude S_p from our analysis in this study, as the low probing accuracy in probing experiments (§4.3) suggests that the agreement information is not encoded (in a useful way) in r_p.

- **Linguistic motivated hypothesis:** *The tokens involved in the respective agreement rules serve as the main source of the agreement feature encoded in* r_t.
  Therefore, for the S-V agreement, S_c is predominantly responsible for providing the agreement feature. In the case of the O-PP agreement, both S_c and S_q play important roles.

- $\mathcal{A}$ is causally dependent on r_t, as agreement feature predictions are obtained by applying NA task through r_t.

In causal inference theory, the do(•) operator denotes an intervention on a causal diagram. In this study, we intervene on the attention weights between the **target** and S_c, S_q, S_i — tokens that encode linearly extractable agreement information, and some of which are relevant to agreement rules. Concretely, the example in Figure 4.11 illustrates a do($\alpha_q = 0$) operation, which means intervening on the causal graph by setting $A^l_{t,q}$ — the attention to "que" — to be zero without changing any other variables. As the relative pronoun "que" plays a very different role in S-V agreement and O-PP agreement according to theoretic linguistics, we would expect different intervention effects resulting from do($\alpha_q = 0$) for the two types of agreement if the Transformer bases its predictions mainly on tokens involved in relevant agreement rules. Following the linguistic motivated hypothesis, we also expect that do($\alpha_c = 0$), which remove the direct contribution of the **cue**, would result in a substantial degradation in the model's agreement prediction for both NA tasks.

Causal effect We define the causal effect of a variable α_i on $\mathcal{A} = \texttt{NA}(r_t)$ as the difference in $\mathcal{A}$ between the original scenario (with original masked attention weights) and a counterfactual scenario (with the weight of α_i set to zero). Formally, for a specific sentence-variable pair (S,α_i), the individual causal effect of α_i on $\mathcal{A}$ is:

$$\begin{aligned} \Delta(S, \alpha_i) &= \texttt{NA}(r_t) - \texttt{NA}(r'_t), \text{ where} \\ r_t &= \texttt{Transformer}(S, A^l) \\ r'_t &= \texttt{Transformer}(S, A^l, \text{ do}(\alpha_i = 0)) \end{aligned} \tag{4.12}$$

The $\texttt{NA}(r_t)$ function implements our number agreement tasks, inputting target form representations and outputting agreement feature $\mathcal{A}$. The Transformer model, $\texttt{Transformer}(S, A^l)$ function, processes the sentence prefix and yields r_t. Here, the causal effect of a token i (whose attention weight to **target** is α_i) on $\mathcal{A}$ can be measured by $\Delta(S, \alpha_i)$. For instance, in Table 4.8, for the original sentence prefix, the model predicts the correct agreement feature

($\texttt{NA}(r_t) = 1$). After the intervention of forcing the attention weight between 'que' and the **target** to be zero, the model's prediction does not match the gold label, thus $\texttt{NA}(r'_t) = 0$. The causal effect of the token 'que' on model's prediction $\mathcal{A}$ for this sentence is $\Delta(S, \alpha_q)$= 1.

4.4.2 Causal experiments and results

Experimental setup Our experiments are based on the same NA tasks discussed in Section 4.2. In this study, the evaluation sets for both types of agreement only include sentences for which the Transformer LM correctly predicted the agreement feature, based on the results described in Section 4.2.4. More specifically, for S-V agreement, we have a dataset denoted as $\mathcal{D}'_{s-v} = \{S^{(i)}, \mathcal{A}^{(i)}\}$, where i =27,278, covering 98.9% of the total examples in the entire S-V agreement evaluation set. Similarly, for O-PP agreement, we have $\mathcal{D}'_{o-pp} = \{S^{(i)}, \mathcal{A}^{(i)}\}$, with i =64,798, representing 94.6% of the total examples from the entire O-PP agreement evaluation set (§4.2.2). Here, $(S^{(i)}, \mathcal{A}^{(i)})$ stands for the pairing of a sentence prefix with the grammatical number of the **cue**. Given the causal model in Figure 4.12 and the equation of individual causal effect (4.12), before any intervention, the outcome $\mathcal{A} = \texttt{NA}(r_t)$ is consistently 1 across both evaluation sets.

We then execute the NA tasks with the Transformer LM again (Section 4.2), but with a twist: when predicting the target verb (only at this moment!), we apply causal interventions. These involve eliminating the direct attention from the **target** to:

i) $\mathcal{S}_c$, which includes the cue and its dependents;

ii) $\mathcal{S}_q$, representing the relative pronoun *que* in the *context*;

iii) both $\mathcal{S}_c$ and $\mathcal{S}_q$;

iv) $\mathcal{S}_i$, which consists of all tokens in the *context* excluding $\mathcal{S}_c$ and $\mathcal{S}_q$.

Average causal effect Considering the individual causal effect equation (4.12) and an evaluation set $\mathcal{D}$ (for which the probed model achieved 100% accuracy before any interventions), we define the average causal effect (ACE) of a specific intervention α_i as:

$$ACE = \frac{\sum_{(S,\mathcal{A})\in\mathcal{D}} \Delta(S, \alpha_i)}{|\mathcal{D}|} \tag{4.13}$$

Simply put, ACE denotes the proportion of initially correctly predicted examples that are incorrectly labeled after an intervention $(do)(\alpha_i = 0)$. ACE can also be interpreted here as the performance degradation caused by a particular intervention.

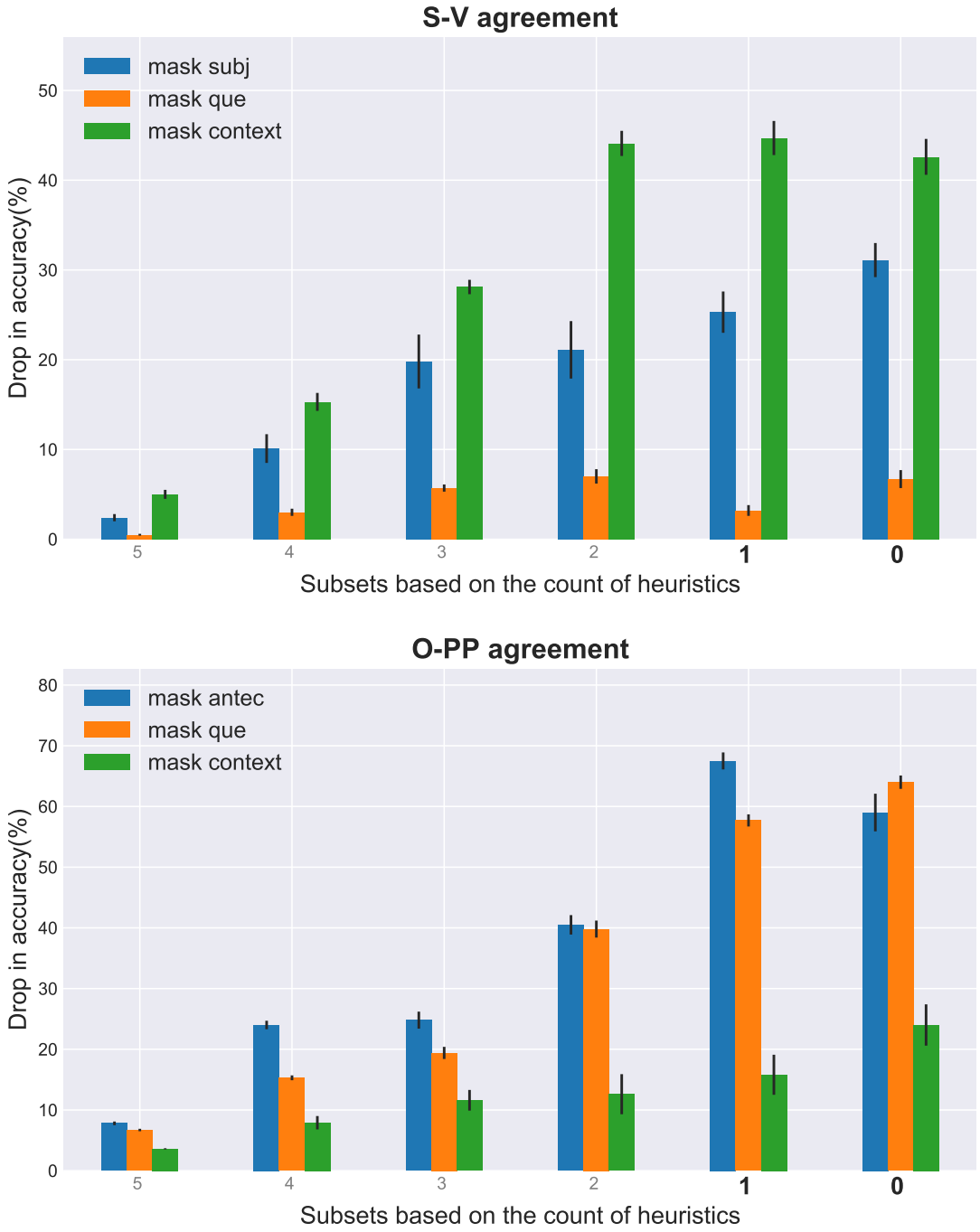


Figure 4.14: Average causal effect of interventions on Transformer's NA task performance, quantified by drop in accuracy before and after different interventions, and further broken down based on prediction difficulty measured by the number of heuristics. The term *cue* here refers to the antecedent and its modifiers (determiners and adjectives) in O-PP agreement, and to the subject and its modifiers in S-V agreement.

Results In Figure 4.14, we report the changes caused by different interventions, as quantified by the average causal effect. This effect represents the drop in performance on NA tasks for both S-V and O-PP agreement.[28] These causal effects are further dissected based on task difficulty. As noted in Section 4.2.4, our investigation primarily focuses on the more challenging cases (i.e. *0* and *1 heuristic* subsets), which cannot be resolved via surface heuristics and thus provide robust evidence of a model's capacity to capture sentence structure information.

As observed, the *cue* (i.e. the antecedent or subject groups) turns out to be critical for predicting the corresponding agreement for both types of agreement. Masking these tokens strongly degrades Transformer's performance on the 0-, 1-heuristic subsets. For the O-PP agreement, we notice a performance drop of over 59%, and for the S-V agreement, a decline of over 25%. Interestingly, the impact of other interventions on the two types of agreement displays marked differences. The role of the relative pronoun "que" in determining the form of the target verbs in these two agreement phenomena significantly diverges. In the case of O-PP agreement, masking the relative pronoun leads to a significant decrease in prediction accuracy, decreasing by over 57%. Conversely, it has minimal effect on the prediction of subject-verb agreement, with accuracy decreasing by no more than 7 percentage points. This suggests that that even though the two agreement phenomena exhibit highly similar surface forms and the model encodes agreement information in a similar manner (as detailed in Section 2.3.2), the Transformer uses separate agreement mechanisms to handle the S-V and O-PP agreements. This distinction thereby lends support to the linguistically-motivated hypothesis.

Figure 4.14 also demonstrates that, for S-V agreement across object relatives, the *context* tokens excluding the **cue** and "que", contribute more significantly to the model's decision than the subject group tokens (i.e., the subject and its dependents) with which the verb agrees. This indicates that **target** receives more agreement information from intermediate tokens than from the direct attention to the nominal subject and its dependent words. This pattern contrasts with the O-PP agreement, where direct attention to the two linguistically-motivated components (i.e., antecedent and "que") can induce an over 80% performance drop, compared to a maximum causal effect of 24% for context tokens. This surprising observation appears to confirm the findings of Ravfogel et al. (2021), who suggested that to predict S-V agreement, the model uses information about relative clause boundaries encoded in its representations. To account for this intriguing observation, we hypothesize that while the agreement information is distributed across all tokens in the *context* segment, the relative clause boundary information is vital for the model to determine how to use this information

[28] See the Table A.9 in the Appendix for the full results.

to inflect the main verb. This would clarify why the *context* tokens play such a crucial role in controlling the agreement. However, further experiments are necessary to confirm this hypothesis.

4.4.3 Conclusion

In this section, our objective is to identify which tokens mainly provide the agreement information used by the model to resolve the NA tasks, and further determine whether the usage pattern reflects the distinct theoretical modeling of S-V and O-PP agreement phenomena. To this end, we designed a causal experiment based on self-attention interventions. In this framework, the model performed the NA tasks from Section 4.2, but with a twist: when predicting the **target**, we cut the direct attention from the **target** to tokens proposed to provide agreement information, based on two hypotheses: the linguistically motivated hypothesis and the linear combination hypothesis (supported by probing results in Section 4.3). The model's post-intervention performance was then compared with the pre-intervention performance, with the performance drop indicative of the causal effect of the intervened tokens.

Our experimental findings reveal a distinct pattern in how Transformers use encoded agreement information across the S-V and O-PP agreements. In the case of O-PP agreement, both the **cue** and relative pronoun "que" serve as crucial sources of agreement information. In contrast, for S-V agreement across relative clauses, while the **cue** plays an important role in determining the **target**'s number, the relative pronoun "que" has minimal impact on the model's agreement behavior. This discrepancy aligns with the linguistically motivated hypothesis and resonates with the theoretical linguistic analysis of the two agreement phenomena, supporting the Transformer's representational adequacy for capturing syntactic information. Additionally, this reinforces the findings of Elazar et al. (2021); Hanna et al. (2023), suggesting that the encoding of linguistic properties, as revealed by probing classifiers, may not necessarily be functionally relevant to the model's predictions. This highlights the importance of transitioning from correlational analysis to causal approaches for a more accurate understanding of model behavior.

This study also opens up several avenues for future research. A primary focus could be on identifying token positions that provide misleading agreement information, leading to incorrect model behavior. To address this, a more controlled experimental setup is needed: Common error patterns from the model's predictions can be extracted (§A.2.1), serving as the basis for creating a template-based evaluation set. Subsequently, the causal framework presented in this section could be applied to individual token positions to identify the

sources of the model's erroneous predictions. Additionally, questions persist about the underlying mechanisms that allow the *target* token to obtain precise agreement information from intermediate tokens, as well as how the model encodes and uses information about relative boundaries. These questions present compelling areas for future investigation.

4.5 Word order: the impact of positional encoding on NLM's syntactic abstraction capacity

In the preceding section, we explored the inner workings of the Transformer language model by applying causal intervention on its self-attention mechanism. Our results indicate that the model is capable of leveraging the hierarchical structure of sentences for nuanced, grammar-based generalization. Yet, one might wonder how a Transformer-based language model can approximate a hierarchical understanding of sentence structure when it processes all tokens simultaneously from linear sequence input. To address this, the current section shifts focus to a critical aspect of language that the self-attention mechanism is not inherently equipped to handle: word order information.

Unlike RNNs, which naturally encode word-order information by sequentially processing input elements, the Transformer model processes all tokens in a sequence simultaneously. As a result, the Transformer does not inherently account for the order of the tokens. This order is crucial for many languages where position encodes grammatical functions. Even in free-order languages, token order remains significant, especially given tokenization into sub-word units, making it essential for tasks like language modeling.

To address this, the Transformer integrates positional embeddings with token embeddings before feeding them into the self-attention mechanism. As detailed in Section 2.2.2, autoregressive language models use an incrementally applied masked self-attention mechanism, which forces the model to attend only to preceding words. This could make positional embeddings redundant, as observed in recent work Haviv et al. (2022). In contrast, masked language models do not have this inherent order modeling, making positional embeddings the sole source of order information.

This study aims to investigate the role of positional embeddings in language modeling and their impact on the syntactic abstraction capacity of Transformer-based language models. Building on the methodology of ablation studies (Meyes et al., 2019), we perform a targeted ablation experiment that focuses on positional embeddings. We compare the performance of autoregressive Transformer LM with and without positional embeddings, and then we run similar experiments with bidirectional Transformer LM (Devlin et al., 2019). Our experiment

is designed to understand the relationship between the model's ability to abstract syntactic structures and its awareness of token order within a sequence.

4.5.1 Positional embeddings in Autoregressive Transformer LM

Experimental setup In all our previous experiments, we considered an autoregressive Transformer LM, denoted as $\mathcal{M}$, with the sinusoidal positional embeddings described in Vaswani et al. (2017), which is the standard setting. To delve deeper into the role of explicit position encoding within Transformer LMs, we consider a variant of this model without positional embeddings, denoted as $\mathcal{M}_{nopos}$. The training process for this position-deprived Transformer LM mirrors that of the original model, using the same training data and the same hyperparameters (§A.1.1).

To assess the importance of positional embeddings for the language modeling objective itself, we conducted an intrinsic evaluation by comparing the validation set perplexity of the model $\mathcal{M}_{nopos}$ and the original model $\mathcal{M}$. As suggested by Hu et al. (2020), perplexity scores do not always give us a clear picture of a model's syntactic ability. Therefore, we also conduct an extrinsic evaluation by comparing the performance of both models, $\mathcal{M}_{nopos}$ and $\mathcal{M}$, on NA tasks. This evaluation helps us assess the importance of explicit position encoding in the model's syntactic abstraction capacity.

Results In terms of the perplexity obtained on the validation set, the model without positional embeddings, $\mathcal{M}_{nopos}$, has an average score of 27.2 across five pre-trained instances, which is strikingly close to the score of 27.0 for the original Transformer. This counterintuitive result suggests that explicit positional encoding may not be as crucial as we thought for pre-training the autoregressive Transformer LM.

When it comes to accuracy on NA tasks, as shown in Figure 4.15, the ablation of positional embeddings surprisingly has a negligible impact. This holds true for both the overall accuracy and the stratified accuracies across subsets of varying difficulty, as determined by our heuristic-based evaluation protocol. Again, this is particularly striking considering the often crucial role of word order in encoding syntactic relationships in languages like French.

A plausible explanation for these surprising results is that the autoregressive Transformer's incremental attention mask, which forces each token to attend only to its preceding tokens, may inherently encode word order information. Although all tokens in a sequence are processed simultaneously, the ability of the model to take into account the predecessors of a given token may effectively allow it to deduce its position within the sequence. We

explore this hypothesis in the following experiment.

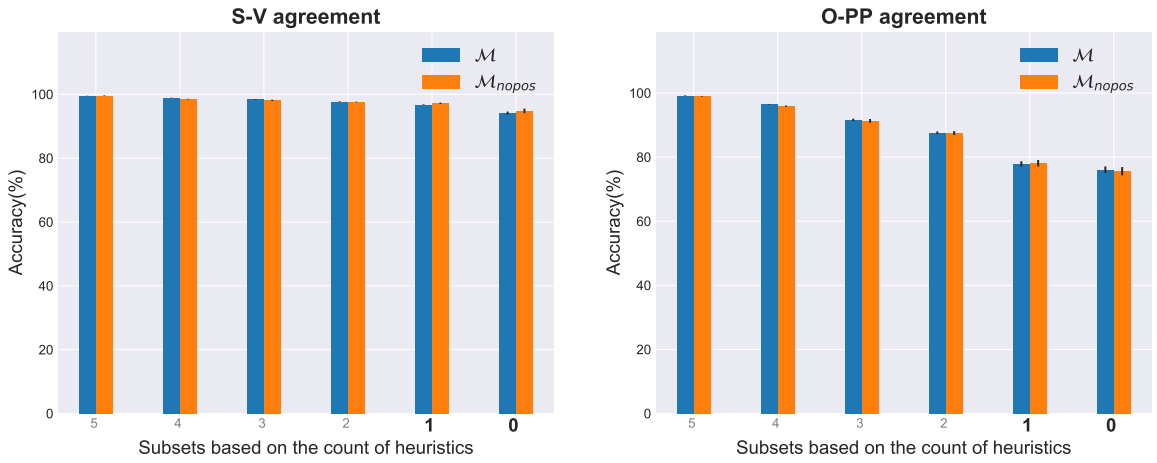


Figure 4.15: Accuracy comparison of autoregressive Transformer LM on two NA tasks with and without positional embeddings. Detailed scores are reported in Appendix Table A.10.

4.5.2 Positional embeddings in masked Transformer LM

The previous experiment reveals that an autoregressive Transformer LM deprived of positional embeddings can still perform comparably in language modeling and NA tasks. This leads us to hypothesize that the incremental self-attention mask might be enabling the model to implicitly reconstruct word order position information during the pretraining. To test this hypothesis, we extend our ablation experiment to a Transformer language model trained with a masked language modeling objective (Devlin et al., 2019). Unlike autoregressive language modeling, where the model predicts each subsequent word based on previous tokens, masked language modeling entails predicting randomly masked tokens using both preceding and succeeding context (§2.2.1). In this context, positional embeddings serve as the sole source of order information. When removed, the MLM generates token representations independent of the actual position of tokens in the input sequence, behaving like a bag-of-words model. The goal here is to investigate whether MLMs can also implicitly learn word order during pre-training without explicit positional embeddings. If they cannot, it would suggest that the incremental attention mask indeed plays a crucial role in the autoregressive model's ability to learn word order information.

Experimental setup We adapted our generic language model to implement a bidirectional Transformer model, which was then trained using a masked language modeling objective (Devlin et al., 2019). We pre-trained the MLMs both with and without positional embeddings on the same training data (as described in 4.2.3), following the same training process used for

the autoregressive models. For each model, we train five different seeds using the optimal hyperparameter configuration.[29] We repeat the ablation experiment from §4.5.1 to compare the perplexity scores of the pretrained MLMs and their performance on NA tasks in the absence of positional embeddings.

As discussed in Section 2.2.1, perplexity is a standard metric for evaluating autoregressive LMs. This metric is not suitable for models trained using a masked language modeling objective, where a masked token w_i is predicted based on its surrounding context $S_{\setminus i}$. To evaluate MLMs, we adopt the pseudo-perplexity approach from Salazar et al. (2020), calculated as the average of the conditional log probabilities $log\mathbb{P}_{MLM}(w_i|S_{\setminus i})$ for each token. While not comparable to conventional perplexity, it allows for a direct comparison between MLMs. More details can be found in the appendix A.1.2.

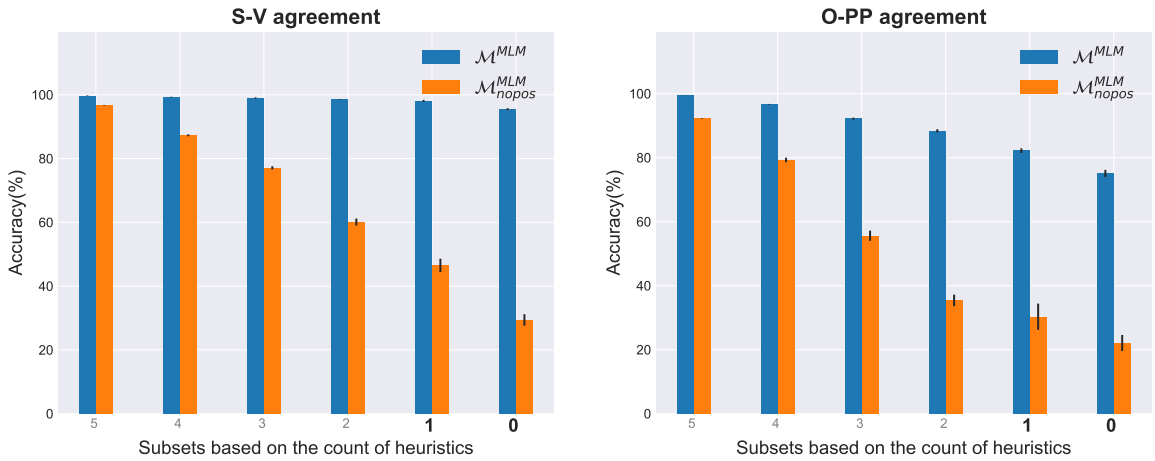


Figure 4.16: Masked Transformer LM's accuracy on two NA tasks with and without positional embeddings. Detailed scores are reported in Appendix Table A.11

Results Our experiments show a substantial difference in pseudo-perplexity scores between the masked language models with and without positional embeddings. The position-aware MLM converges to a very low score of 5.6, whereas the nopos MLM performs significantly worse, with a score of 57.2. This result aligns with the observations of previous studies such as Sinha et al. (2021) and Haviv et al. (2022), which noted that MLMs deprived of explicit position encoding suffer a substantial decline in pretraining task performance.

Regarding the performance on number agreement tasks, as seen in Figure 4.16, the ablation of positional embeddings leads to a substantial decrease in accuracy across both agreement tasks. Particularly, in the most challenging cases, the performance drop reaches 66% for S-V agreement and 53% for O-PP agreement. These findings indicate that explicit

[29]Please refer to §A.1.1 for details on the hyperparameters.

position encoding plays a critical role in MLM's syntactic abstraction ability. This, in turn, provides evidence supporting our initial hypothesis concerning autoregressive LM, suggesting that the incremental attention mask may enable it to implicitly reconstruct absolute word position information.

4.5.3 Conclusion

In this study, we have performed a set of positional embedding ablation experiments with both autoregressive and bidirectional Transformer LMs. These models are evaluated based on their performance in language modeling and number agreement tasks, comparing the outcomes of models with positional embeddings against those without. Our results show that the autoregressive language model deprived of positional embeddings (nopos) achieves competitive performance compared to its original counterpart in both the language modeling task and the NA tasks. In contrast, bidirectional language models without positional embeddings experience substantial performance degradation in both language modeling and NA tasks. This stark contrast highlights the critical role positional embeddings play in bidirectional models in identifying token positions. Meanwhile, autoregressive LMs appear to leverage the incremental attention mask to implicitly reconstruct word order information, thereby the absence of explicit position encoding has very little impact on the model's performance.

4.6 Conclusion and discussion

In this chapter, we conducted a contrastive study to explore the core question of my thesis: Does the Transformer language model exploit abstract sentence structures, or does it primarily rely on surface patterns when handling structure-sensitive phenomena? Our primary goals are twofold: to assess the behavioral and representational adequacy of the autoregressive Transformer model in relation to human syntactic processing, and to develop a linguistically-informed framework to enhance the interpretability of this complex model. To achieve this, we use number agreement tasks to explore how the Transformer LM processes two forms of agreement in French: long-distance subject-verb and object-past participle agreements, both involving object relative clauses. While these two types of agreement share superficial similarities in word sequences, their linguistic analyses fundamentally diverge.

Our approach begins with the proposal of a heuristic-based evaluation protocol, which effectively constrains the impact of surface heuristics in conventional number agreement

tasks, providing a robust groundwork for our subsequent experiments. In our initial set of experiments, we assessed the ability of an autoregressive Transformer language model to predict these two types of agreement. The results indicate that the model exhibits high predictive accuracy, even under challenging conditions where all surface heuristics fall short. Further control experiments underscore the Transformer's ability to generalize beyond collocational cues and strong frequency biases. Taken together, these results strongly suggest that the Transformer is not merely exploiting surface patterns, but may be capturing some form of abstract sentence structure. This evidence indicates that Transformer meets the first prerequisite — behavioral-level similarity — for genuine syntactic generalization.

Building on the strong behavioral performance of the Transformer, we took a more in-depth investigation to assess where syntactic agreement information is located within the model's inner representations, as a measure of its representational adequacy. Our second set of experiments, using a probing approach, reveal that the relevant agreement information is mainly linearly encoded across all tokens between the **cue** and the **target**. Interestingly, within the contextualized representations, this information is found in a small number of highly correlated dimensions, while also being fuzzily encoded in a redundant manner across the remaining dimensions. Notably, we observe a very similar distribution pattern of agreement information for both types of agreement phenomenon.

To go beyond the limitations of probing, which mainly reveals correlations between encoded information and the model behavior, we introduced a causal framework. This framework relies on counterfactual analysis and involves intervening directly on the model's self-attention mechanism. Our causal experiments provide further evidence that the Transformer model's success is based on linguistically justified cues, consistent with French grammar. Importantly, the abstract structure uncovered by the Transformer model aligns with the distinct theoretical modeling of the two structure-sensitive phenomena we examined. This alignment supports the Transformer's representational adequacy for capturing syntactic information, suggesting that its internal mechanisms are not merely statistically efficient but also linguistically meaningful. Consequently, this lends additional credibility to the potential of the Transformer as an explanatory tool for human syntactic processing.

Additionally, to investigate how Transformer language models approximate syntactic structures from string input, we conducted a set of positional embedding ablation experiments with autoregressive and bidirectional Transformer LMs. We find that explicit position encoding has little impact on the general function and syntactic abstraction ability of the autoregressive LM. This is likely because the model can leverage the absolute word order information from the incremental attention mask.

Our study represents an initial step towards a deeper understanding of how neural language models function. Our analysis framework, which begins with behavioral syntactic tasks fortified by heuristic-based evaluation, then pairs with linguistic probes, and finally explores through counterfactual analysis via causal intervention, provides a robust methodology to assess the syntactic abstraction capacity of neural language models. Notably, our findings regarding the linguistically motivated distribution of syntactic information in Transformers' representations could extend easily to other linguistic phenomena and languages.

Nevertheless, many questions remain unresolved, such as the precise mechanism by which Transformers track agreement information and how they encode long-distance dependencies from linear word sequences. It is also of interest to explore whether the model can emulate a human-like rule-based generalization to dynamically recombine familiar structures in novel situations. These avenues represent exciting directions for further investigation. Our work thus far only marks the beginning of a rich and exciting journey toward deciphering the complex inner workings of neural language models.

Part III

Assessing model capacity to generalize compositionally observed structures

Chapter 5

SLOG: A Structural Generalization Test for Semantic Parsing

5.1 Introduction . . . 99
5.2 Overview of SLOG benchmark . . . 103
5.2.1 Novel recursion depth . . . 105
5.2.2 Novel combination of modified phrases and grammatical roles . . . 107
5.2.3 Novel gap positions . . . 109
5.2.4 Novel *wh*-questions . . . 110
5.3 Dataset generation . . . 111
5.4 Experimental setup . . . 113
5.4.1 Models . . . 113
5.4.2 Evaluation metric . . . 115
5.5 Results . . . 115
5.5.1 Unobserved depth and length both affect depth generalization 117
5.5.2 Unobserved long-distance dependencies make generalization difficult . . . 118
5.5.3 Gap generalizations are challenging for all tested models . 119
5.6 Conclusion . . . 124

In the previous part of this dissertation, we assessed the syntactic capabilities of an autoregressive Transformer, specifically focusing on its ability to handle two syntax-sensitive phenomena. Our findings indicate that the model acquires remarkably nuanced representations of sentence structure, as evidenced by its strong performance on both behavioral-level tasks and measures of representational adequacy when evaluated on unseen, held-out evaluation sets.

However, this leads us to another critical dimension of inquiry: the nature and depth of the model's observed syntactic generalization. Early studies by Fodor and Lepore (2002) and Marcus (2003) posited that neural language models like RNNs may lack the capacity for genuine compositional syntactic generalization due to the absence of explicit symbolic representation. These models, they argued, often rely on similarity-based inference derived from patterns encountered during training. This notion is supported by recent studies, such as Bender et al. (2021), which describe language models as "stochastic parrots" that primarily memorize and shallowly recombine observed examples. Additionally, the capacity for extensive memorization in neural models is well recognized in the literature (Halevy et al., 2009; Zhang et al., 2021).

Let's consider the case of subject-verb agreement across relative clauses, as detailed in Section 4.2.1. Our findings from Chapter 4 suggest that the Transformer leverages the structural relationships between words to accurately predict the long-distance dependency illustrated in (19).

(19) Target sentence: $\mathbf{NP}_1$ + Relative Clause_1 + $\mathbf{V}_1$

(20) Compositional generalization:

  a. $\mathbf{NP}_2$ + $\mathbf{V}_2$

  b. ... V + $\mathbf{NP}_3$ + Relative Clause_3.

(21) Similarity-based generalization: $\mathbf{NP}_4$ + Relative Clause_4 + $\mathbf{V}_4$

As proposed by Fodor and Marcus, we can hypothesize two potential ways the model learns such a structural relationship. First, the model might rely on training sentences such as (20a) and (20b). By compositionally combining these structures, the model could infer the long-distance dependency relationship in (19). This approach exemplifies compositional generalization. Second, the model might extrapolate grammatical knowledge from training sentences like in (21) to the target sentence (19) based on their structural similarity. In this process, the model could grasp concepts like syntactic subjecthood, morpho-syntactic number, and the boundaries of relative clauses, subsequently formulating distributional rules at these abstract category levels. By recapitulating its training data through structural similarity and lexico-categorical abstraction, the model can generalize to unfamiliar sentences

with known structures. Given that in Chapter 4 we do not control the types of syntactic structures in the model's pretraining data, it is plausible that the model's performance on unseen sentences is driven more by memorization of structures encountered in the training data than by genuine compositional generalization.

This brings us to the core questions: To what extent do these models rely on memorizing structures they have encountered during training? More importantly, can these models achieve generalizable abstractions by compositionally applying observed syntactic rules to interpret new, unseen linguistic patterns? While systematic compositional generalization is a key component in human linguistic cognition, it remains an open question whether these neural models can also dynamically recombine known elements in a compositionally consistent manner with their underlying syntactic structure.

To explore these questions, the current chapter introduces a compositional generalization challenge test. This test aims to directly probe the model's capability to compositionally interpret unseen syntactic constructions through the combination of known structures, and will include experiments with similar Transformer-based models.

Outline In this chapter,[1] Section 5.1 outlines the foundational aspects of compositional generalization and the semantic parsing task, followed by an introduction to the COGS benchmark (Kim and Linzen, 2020), which serves as the starting point of this study. Section 5.2 provides an overview of our SLOG benchmark, a dataset specifically constructed to focus on compositional structural generalization. Section 5.3 details the dataset generation, and Section 5.4 describes the experimental setting, discussing the models evaluated and the evaluation metric. Moving on to Section 5.5, we present the findings of our investigation into three Transformers-based models and a structure-informed parsing model. And Section 5.6 provides a summarizing conclusion.

5.1 Introduction

The immense productivity of human language enables us to understand and produce a potentially infinite number of sentences from finite input elements (Chomsky, 1965; Hauser et al., 2002). This linguistic productivity is generally attributed to the principle of compositionality — the assumption that the meaning of an expression is a function of the meanings

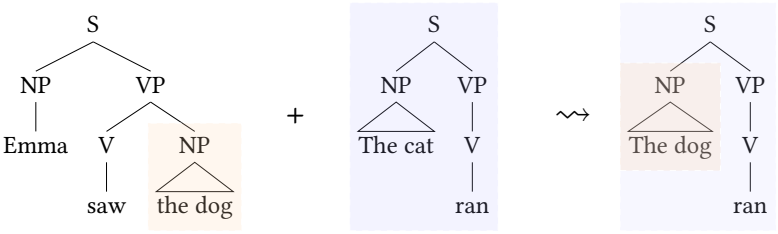
[1]This chapter stems from my visiting project at New York University, mentored by Prof. Tal Linzen and Dr. Najoung Kim, and conducted in collaboration with Alexander Koller, Yuekun Yao, and Lucia Donatelli. This chapter draws largely from our paper titled "SLOG: A Structural Generalization Benchmark for Semantic Parsing", which I primarily authored and has been accepted by EMNLP 2023.

of its components and the way they are syntactically combined (Frege, 1948; Partee, 1984). Reflecting this principle, human linguistic competence exhibits *compositional generalization*: the algebraic capacity to understand and produce novel sentences by reassembling known elements (Montague, 1974).

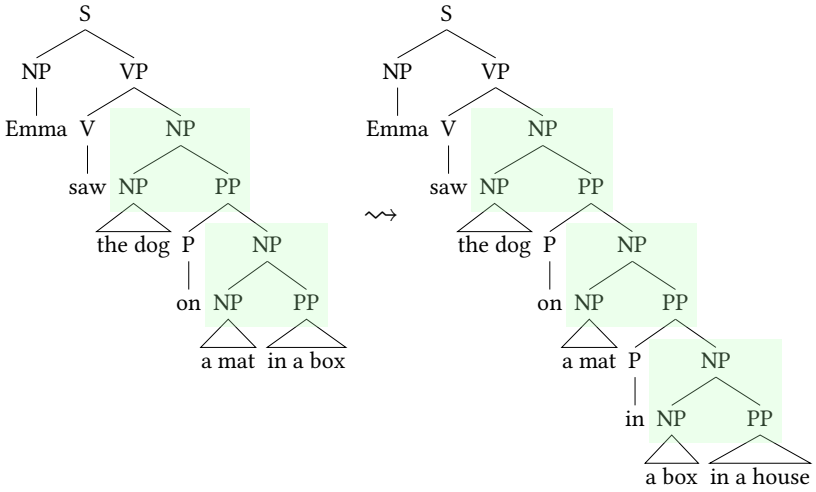
Central to this compositional generalization are two key concepts: *systematicity* and *productivity*, as presented by Fodor and Pylyshyn (1988). Systematicity refers to the consistent application of compositional rules to linguistic elements to derive meaning. This is analogous to how algebraic functions are consistently applied to appropriate variables. In practical terms, systematicity allows humans to extend their understanding to sentences or concepts that are systematically related. Productivity, on the other hand, is the ability to generate an infinite variety of sentences or thoughts from a finite set of words or concepts. In language, this is seen in our ability to produce and understand new sentences that we have never encountered before.

A classic illustration of this, presented by Fodor, is that people who know the meaning of *John loves Mary*, along with its underlying syntactic rules, can naturally understand the meaning of *Mary loves John*, despite never having encountered it before (Fodor and Pylyshyn, 1988). This exemplifies the systematicity in human language, where understanding such sentences involves the application of the same rules to recombine the same lexical units. In these two sentences, the verb *loves* operates as a function, taking two variables (the subject and the object) and recombining the lexical units *John* and *Mary* in a way that generates different semantic meanings. This rule-based systematic generalization mechanism is widely assumed as the means humans use to handle linguistic productivity.

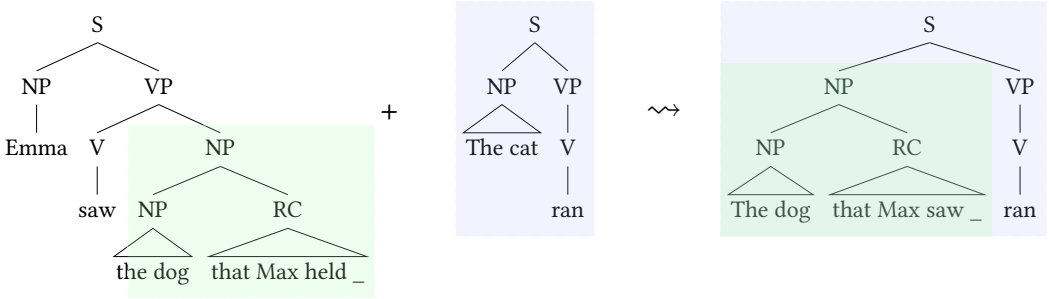
Recent advances in NLP, particularly those based on neural networks, do not explicitly rely on the principle of compositionality. Despite this, their empirical success in various tasks suggests that they must have some form of effective generalization. This raises the question: Do these models learn to generalize in a manner similar to human-like compositional understanding, capturing both systematicity and productivity? In recent years, a growing body of research has explored whether models possess such capability. Benchmarks for compositional generalization in semantic parsing have emerged as a useful tool to assess model's compositional capability (Lake and Baroni, 2018; Hupkes et al., 2020; Keysers et al., 2007; Kim and Linzen, 2020). Semantic parsing tasks in these studies involve translating natural language expressions into semantic representations. The models are evaluated on a *generalization set*, which is sampled from a distribution that systematically differs from the training distribution. This shift from training to evaluation is designed under the principle of compositionality and often includes new combinations of lexical units and observed rules, deeper recursions of observed patterns, or longer sequences.



(a) Lexical generalization: object → subject (COGS)



(b) Structural generalization: deeper recursion depths (COGS & SLOG)



(c) Structural generalization: RC object → RC subject (SLOG)

Figure 5.1: Examples of lexical generalization in COGS — (a), and structural generalization in COGS — (b) and in SLOG — (b, c). The SLOG task requires mapping the generalization examples to their logical forms; the corresponding logical forms are shown in Table 5.1.

The COGS (**CO**mpositional **G**eneralization Challenge based on **S**emantic Interpretation) dataset (Kim and Linzen, 2020) in particular has become a widely used benchmark, as it is designed to expose a generalization gap between training and testing data that many recent semantic parsers still struggle with. COGS distinguishes two types of generalization challenges that require different types of algebraic compositional strategies: *lexical generalization* tests the ability to interpret novel combinations of known lexical items and known linguistic structures (Figure 5.1a), and *structural generalization* tests the ability to combine known structures into a novel structure (Figure 5.1b and 5.1c). Importantly, the majority of

generalization types in COGS target lexical generalization (18 of 21 generalization types, 86% of the dataset). As lexical generalization is arguably easier than structural generalization (e.g., solvable by simple slot-filling), this imbalance may lead to overall performance numbers that are overly optimistic with regard to a model's capacity to generalize compositionally, as pointed out by Weißenhorn et al. (2022) and Yao and Koller (2022).

To facilitate a more comprehensive evaluation of structural generalization, we introduce SLOG, a **S**tructural **LO**ng-distance dependencies **G**eneralization benchmark. SLOG extends COGS to include 17 cases of structural generalization in total (14 new cases and 3 existing cases from COGS) (§5.2). The novel generalizations we introduce target two key structural features of human language: recursion and filler-gap dependencies.

Prior research has extensively investigated the processing of recursive constructions, a key feature of human language enabling the creation of complex, nested structures and hierarchical relationships (Hauser et al., 2002). This area of study spans both artificial neural networks and human cognition (Christiansen and Chater, 1999; Lakretz et al., 2021a; McCoy et al., 2021, ; i.a.). Using artificial languages to isolate syntactic properties, research has shown that humans can learn and extrapolate nested patterns to deeper levels (McCoy et al., 2021). In contrast, the capabilities of Transformer-based models in capturing recursive regularities have yielded mixed results (Bhattamishra et al., 2020; Hahn, 2020; Ebrahimi et al., 2020; Lakretz et al., 2021a). For filler-gap dependencies, a particularly challenging type of long-distance dependency involves generalization about the absence of material. Prior work has centered on syntactic tasks involving *wh*-questions or relative clauses (Wilcox et al., 2018; Marvin and Linzen, 2018; Li et al., 2023b, ; i.a.). These studies primarily use language modeling as the task and do not require mapping to semantic representations. SLOG diverges from these works by incorporating recursion and filler-gap dependency in a more naturalistic setting and directly assess the semantic mapping. Importantly, rather than isolating syntactic generalization from linguistic meaning, our approach aims to assess whether models rely on syntactic generalization that aligns with human inductive biases to derive the meaning of complex sentences.

We use SLOG to evaluate a Seq2Seq Transformer model trained from scratch (Vaswani et al., 2017), two pre-trained Transformers (T5-base; Raffel et al. 2020 and LLaMA; Touvron et al. 2023), and a structure-informed[2] model (AM-Parser; Weißenhorn et al. 2022). In comparison to their overall performance on COGS, all models exhibit considerably lower performance on SLOG (§5.5). The generalization accuracy of Transformer-based models, including pre-trained ones, only reaches 40.6%, and even a structure-informed parser, while

[2]In this study, 'structure-informed' refers specifically to models that incorporate explicit representations of linguistic structure.

exhibiting near-perfect generalization on COGS structural cases, only achieves 70.8% on SLOG. The cases in which models struggle exhibit distinct patterns and suggest varied approaches of models to solving the task. An error analysis reveals that the AM-Parser generalizes well on existing structural generalization cases in COGS but struggles with the gap constructions introduced in SLOG due to inherent structural limitations, which we discuss in Section §5.5.3. Transformers tend to erroneously repeat frequent meaning representation subsequences observed during training. Even with pretraining, they struggle with unseen long-distance dependencies, which we attribute to their bias towards shorter predicate-argument dependencies (§5.5.2).

Overall, the discrepancy in performance between SLOG and COGS illuminates the notable gap between models' lexical and structural generalization abilities. It highlights the utility of SLOG in exposing the limitations of current models that have been shown to achieve high performance on existing generalization benchmarks, and helps foreground the different weaknesses of these models. While Transformer-based models can approximate compositional behavior to a certain extent, our findings suggest that they do not seem to rely on the kind of syntactic generalization rooted in symbolic compositional rules, which are believed to drive human linguistic systematicity and productivity.

5.2 Overview of SLOG benchmark

SLOG follows the semantic parsing format used in COGS, where the task is to translate English expressions into logic-based meaning representations (Table 5.1). The dataset structure follows the basic design principles of COGS: there is a systematic gap between the training set and the generalization set, where target constructions in the generalization set are not included in the training set, but pieces of the training set can be recombined to arrive at their correct meanings. For example, as illustrated in example (c) of Table 5.1, noun phrases that appear only in object position during training must be reinterpreted in subject position during generalization.

SLOG is generated using manually specified rules (§5.3), adopting the same meaning representation as COGS. The COGS logical form (LF), derived from Reddy et al. (2017), is based on Neo-Davidsonian view of verbal arguments. In this approach, the semantic units are mapped to indexed variables. For example, in:

(22) The cat **ran**.
$\rightsquigarrow$ `*cat(`x_1`); run.agent(`x_2, x_1`)`

The variable x_1 denotes an entity that is both a cat and the agent of a running event, while x_2 represents the running event. The variable indices are determined by the linear position

	Training	Generalization
COGS	(a)The cat **ran**. $\rightsquigarrow$ *cat(x_1); run.agent(x_2, x_1) Emma saw **the dog**. $\rightsquigarrow$ *dog(x_3);see.agent(x_1,Emma) $\wedge$ see.theme(x_1, x_3)	**The dog** ran. $\rightsquigarrow$ *dog(x_1); run.agent(x_2, x_1)
COGS & SLOG	(b) Emma saw the dog **on a mat in a box**. $\rightsquigarrow$ *dog(x_3);see.agent(x_1,Emma) $\wedge$ see.theme(x_1, x_3) dog.nmod.on(x_3, x_6) $\wedge$ mat(x_6) $\wedge$ mat.nmod.in(x_6, x_9) $\wedge$ box(x_9)	Emma saw the dog **on a mat in a box in a house**. $\rightsquigarrow$*dog(x_3);see.agent(x_1,Emma) $\wedge$ see.theme(x_1, x_3) dog.nmod.on(x_3, x_6) $\wedge$ mat(x_6) $\wedge$ mat.nmod.in(x_6, x_9) $\wedge$ box(x_9) $\wedge$ box.nmod.in(x_9, x_{12}) $\wedge$ house(x_{12})
SLOG	(c) The cat **ran**. $\rightsquigarrow$ *cat(x_1); run.agent(x_2, x_1) Emma saw **the dog that Max held**. $\rightsquigarrow$ *dog(x_3); see.agent(x_1,Emma) $\wedge$ see.theme(x_1, x_3) $\wedge$ dog.nmod(x_3, x_6) $\wedge$ hold.agent(x_6,Max) $\wedge$ hold.theme(x_6, x_3)	**The dog that Max saw** ran. $\rightsquigarrow$ *dog(x_1); see.agent(x_4,Max) $\wedge$ see.theme(x_4, x_1) $\wedge$ dog.nmod(x_1, x_4) $\wedge$ run.agent(x_5, x_1)

Table 5.1: Examples of two distinct types of generalization: lexical generalization in COGS — (a), structural generalization in COGS—(b) and in SLOG — (b, c). The symbol $\rightsquigarrow$ indicates the task of translating an English sentence into its corresponding meaning representation.

of the phrasal head in the input sentence. For example, *cat* corresponds to x_1, since, under 0-indexing, *cat* appears in linear position 1 of the English sentence *The cat ran.* Definite descriptions are marked by a preceding asterisk and are placed at the beginning of the LF: *cat(x_1) is separated from the remaining conjuncts by a ';'.[3] This format can represent coreferential relations effectively, for example:

(23) Emma saw **the dog that Max held __**. $\rightsquigarrow$
*dog(x_3); see.agent(x_1, Emma) $\wedge$ see.theme(x_1, x_3) $\wedge$ dog.nmod(x_3, x_6) $\wedge$ hold.agent(x_6, Max) $\wedge$ hold.theme(x_6, x_3)

The variable x_3 denotes a *dog* entity that is both the theme of a seeing event in the main clause and the theme of a holding event in the relative clause.

SLOG contains 17 structural generalization cases grouped into four categories. These generalization cases are primarily motivated by frequency asymmetries in natural lan-

[3]Proper nouns, treated as constants, are not represented by numbered variables but by their actual word forms as illustrated in the example (23).

guage, where simpler structures are more common than complex ones; in other words, SLOG assesses whether NLP models can extrapolate from frequent patterns to their less frequent counterparts. We describe the four categories below; see Table 5.2 for the full list of generalization cases.

5.2.1 Novel recursion depth

Recursion allows small, hierarchical phrases to be combined to create larger phrases composed of identical substructures. This combination process can be repeated an unbounded number of times. The COGS dataset tests a model's ability to apply recursion via two cases: sentential complements (e.g. (24); tail complementizer phrase (CP) recursion henceforth) and nominal prepositional phrase modifiers (e.g. (25); PPs recursion henceforth). For both cases, the training set contains recursive depths of 0–2 (0 indicating no PP/CP), and the generalization set contains strictly greater depths of 3–12.

(24) Tail CP recursion depths 2:
Mary knows [that John knows [that Emma cooks]$_{CP}$]$_{CP}$

(25) PP recursion depths 2:
Ava saw the ball [in the bottle [on the table]$_{PP}$]$_{PP}$

By contrast, the SLOG training set includes recursion of depth 0–2 and 4, and the generalization set contains both an intermediate depth of 3 and deeper depths of 5–12. Including both shallower and deeper embeddings allows us to determine if any difficulty in generalizing to an unseen embedding depth is a consequence of the model's more general difficulty in processing longer sequences than observed in training (Lake and Baroni, 2018; Herzig et al., 2021; Anil et al., 2022) rather than a more specific issue with applying recursion to generate novel constructions.

In addition to this new depth split, SLOG introduces a new recursion construction. COGS involves only tail recursion, which features recursive PPs and CPs with right-branch embeddings. SLOG extends this with center embedding, where a phrase is embedded in the middle of another phrase of the same type, leaving elements on both side of the embedded component and producing well-parenthesized long-distance dependencies, as illustrated by the subscripts in (26).

(26) Eva saw the mouse [that the cat$_1$ [that the dog$_2$ chased$_2$] held$_1$].

At the same recursion depths, the average LF length increases from PP recursion to tail CP

Generalization cases	Training	Generalization
§5.2.1 Novel Recursion Depth		
Deeper depth generalization		
✓Prepositional phrase (PP) max depth 4 → depth 5-12	Ava saw the ball **in** the bottle **on** the table.	Ava saw the cat **in** the box **on** the mat **on** the bed **on** the floor **in** the room.
✓Tail CP recursion max depth 4 → depth 5-12	Ava believed **that** Emma said **that** a fish froze.	Ava said **that** Emma liked **that** Max believed **that** Noah found **that** Liam saw **that** the cat slept.
Center embedding max depth 4 → depth 5-12	Eva saw the cat **that** the horse **that** the dog liked chased.	Ava held the dress **that** a store **that** a girl **that** a boy **that** a cat **that** a man drew saw loved knew sold.
Shallower depth generalization		
PP recursion max depth 4 → depth 3	Emma saw the ball **in** the bottle **on** the table **on** the floor **in** the office.	Ava saw the cat **on** the mat **on** the floor **in** the office.
Tail CP recursion max depth 4 → depth 3	Ava believed **that** Emma said **that** Max found **that** a cat saw **that** a fish froze.	Ava said **that** Emma liked **that** Max believed **that** the cat slept.
Center embedding max depth 4 → depth 3	Eva saw the cat **that** the horse **that** the dog **that** the man **that** the girl loved found liked chased.	Emma bought the dress **that** the store **that** the woman **that** Mike knew liked sold.
§5.2.2 Novel Combination of Modified Phrases and Grammatical Roles		
PP in direct object NPs		
✓→ PP in subject NPs	Noah ate **the cake on the plate**.	**The cake on the table** burned.
→ PP in indirect object NPs	Noah ate **the cake on the plate**.	Max gave a fish to **a cat on a table**.
PC in direct object NPs		
→ RC in subject NPs	Noah saw **the cat that froze**.	**The cat that froze** smiled.
→ RC in indirect object NPs	Noah saw **the cat that froze**.	Max gave a fish to **a cat that ran**.
§5.2.3 Novel Gap positions		
Subject, direct object-extracted RC → Indirect object-extracted RC	Noah saw the cat that gave a fish to Liam. ⊕ Noah saw the cat that Liam liked _.	Noah saw the cat that Emma gave a cake to _ .
Subject, direct object *wh*-questions → Indirect object *wh*-questions	Who saw the cat? ⊕ What did Emma see _?	Who did Noah give the cake to _?
§5.2.4 Novel *Wh*-questions		
Subject, object *wh*-Q of simple transitives		
→ Active subject *wh*-questions	**Who saw** the cat? ⊕ Emma **wanted** to sleep.	**Who wanted** to sleep ?
→ Passive subject *wh*-questions	**Who** did Emma see _? ⊕ The boy **was found** by Emma.	**Who was helped** by Emma?
→ Direct object *wh*-questions with ditransitive verbs	**What** did Emma see _? ⊕ Emma **gave** a fish **to** the cat.	**What** did Emma **give** _ to the cat?
→ *Wh*-questions with modified NPs	What did **the cat** see _? ⊕ the cat **on the mat**	What did **the cat on a table** see _?
→ *Wh*-questions long movement	**What** did the cat **see** _? ⊕ Emma **said that** the cat saw a fish.	**What** did Emma **say that** the cat **found** _?

Table 5.2: A full list of SLOG generalization cases. Each sentence in the table corresponds to a (sentence, logical form) pair, as illustrated in Figure 5.1. ⊕ denotes the combination of two observed structures, which allows to interpret the target novel structure. Some cases cover multiple sub-case constructions: e.g. all ditransitive verbs include both double-object and prepositional constructions. Due to space limitations, only one example is provided for each case. The three cases marked with '✓' are already present in the COGS dataset.

recursion to center embedding.

In natural language, the depth of recursion is rarely greater than five and center embedding is generally limited to two levels (Karlsson, 2007, 2010). SLOG tests deeper recursive patterns up to depth 12. While this may surpass human processing abilities for reasons presumed to be linked to memory constraints (Gibson and Thomas, 1999; Karlsson, 2007), deeper embedding depth remains grammatical, echoing Chomsky's competence versus performance distinction. Importantly, we also note that our goal with SLOG is to assess the linguistic competence of NLP models and to investigate whether they achieve their performance through human-like inductive biases that favor compositional generalization. Testing at these greater depths allows us to more comprehensively probe the models' capabilities and limitations.

5.2.2 Novel combination of modified phrases and grammatical roles

SLOG also tests the capacity to generalize complex NPs to new syntactic positions. SLOG introduces relative clause modifiers, in addition to PP modifiers already included in COGS.

Prepositional Phrase modifiers

The most challenging case in COGS involves interpreting prepositional phrases (PPs) within subject NPs (27), while the training set only contains PPs within direct object NPs (28). Recent Seq2Seq models consistently failed to handle this case (Akyurek and Andreas, 2021; Zheng and Lapata, 2022; Yao and Koller, 2022). To further investigate what makes this challenging generalization hard for models, we take a two-fold approach in SLOG. First, we additionally include generalization targeting indirect object modification, as illustrated in (29).

(27) [The **cat** on the mat]$_{subj}$ **ran**.

(28) Emma saw [the cat on a table]$_{dobj}$.

(29) Sub-cases of indirect object modification:

  a. Emma **gave** [a cat on the mat]$_{iobj}$ a **fish**.

  b. Emma gave a fish to [a cat on the mat]$_{iobj}$.

  c. A fish was given to [a cat on the mat]$_{iobj}$.

We expect sub-cases of indirect object modification to pose challenges of varying difficulty, depending on the distance of the predicate-argument dependency. For example,

generalization to indirect object modification in active oblique datives (29a) introduces an unobserved long-distance dependency between the verb *gave* and the direct object *a fish* across a non-argument NP *the mat*.[4] In contrast, sub-cases like (29b) and (29c), where the non-argument NP *the mat* occurs at the end of the sentences, do not introduce this kind of predicate-argument dependency across an intervening NP and are therefore expected to be relatively easier.

Second, SLOG's training set additionally includes standalone PP-modified NPs, as exemplified in (31a), to prevent modifiers from being associated with only a particular range of token indices, as pointed out by Wu et al. (2023): In COGS, PPs were restricted to the object position, such as in (30a), where the modifier conjunct in the logic form — `cat.nmod.on` (x_i, x_j) — applies only for i>=3, so models never observed the association of modifiers with linearly-earlier indices (e.g., in (32), `cat.nmod.on` (x_i, x_j), with i=1). This makes it difficult to isolate the impact of indices correlation from structural generalization. The inclusion of such fragments in SLOG, absent in COGS but common in child-directed speech (Wells and Bridges, 1981; Cameron-Faulkner et al., 2003), serve as a signal that the range of variables indices associated with PP modifiers is not restricted to the object position.[5]

(30) COGS Training

a. Emma saw the cat on a table $\rightsquigarrow$ `*cat(`x_3`); see.agent(`x_1`, Emma)` $\wedge$ `see.theme(`x_1, x_3`)` $\wedge$ `cat.nmod.on(`$\mathbf{x_3}, x_6$`)` $\wedge$ `table(`x_6`)`

b. The dog ran. $\rightsquigarrow$ `*dog(`x_1`); run.agent(`x_2, x_1`)`

(31) SLOG Training

a. the cat on a table $\rightsquigarrow$ `*cat(`x_1`); cat.nmod.on(`$\mathbf{x_1}, x_4$`)` $\wedge$ `table(`x_4`)`

b. COGS Training

(32) Generalization

The cat on a mat ran. $\rightsquigarrow$ `*cat(`x_1`); cat.nmod.on(`$\mathbf{x_1}, x_4$`)` $\wedge$ `map(`x_4`)` $\wedge$ run.agent (x_5, x_1)

Relative clause modifiers

Similar to PP modifiers, NPs with relative clause (RC) modifiers, as in (33), can occupy any position that an unmodified NP can fill. We expect RC modifiers to pose a greater challenge compared to PP modifiers, as they involve *gap constructions*, in which a phrase needs to

[4] This observation holds true for the generalization to subject modification shown in (27).

[5] This phenomenon is also evident in the CHILDES corpora, where we observed many standalone PP-modified NPs (e.g., *the CD on the desk!*) in child-directed speech.

be interpreted in a position other than its canonical position in a declarative clause — we will refer to this as *extraction* (Sag, 2010). We mark gap positions with an underscore. In (33), *the dog* should be interpreted as if it occupies the gap position as the direct object of *held*; in the logical form, this is represented by the fact that x_3 is filling both `see.theme` and `hold.theme`.

(33) Emma saw the dog that Max held __.

$\rightsquigarrow$ `*dog(`x_3`); see.agent(`x_1`, Emma)` $\wedge$ **`see.theme`**`(`x_1`,` x_3`)` $\wedge$ `dog.nmod(`x_3, x_6`)` $\wedge$ `hold.agent(`x_6`, Max)` $\wedge$ **`hold.theme`**`(`x_6`,` x_3`)`

To test for generalization to RC-modified NPs in unseen grammatical roles, SLOG's training set contains RC modifiers in direct object NPs (34b) as well as standalone RC-modified NPs like (34a), and the generalization set contains RC modifiers in subject NPs such as (35a) and indirect object NPs (35b). This is analogous to the PP modifier cases.

(34) Training

a. the cat that Liam fed __

b. Emma saw [the cat that Max held __]$_{dobj}$

(35) Generalization

a. [The cat that Emma found __]$_{subj}$ smiled.

b. Liam gave [a cat that Emma held __]$_{iobj}$ a fish.

5.2.3 Novel gap positions

SLOG's training set contains both subject and direct object-extraction; these are the most frequent extraction positions in both written and spoken English corpora (Roland et al., 2007; Atkinson et al., 2018). We test generalization to a less frequent extraction position: indirect object. In this case, the training set only includes subject-extracted and direct object-extracted examples as in (36). Models must then interpret indirect object-extracted relative clauses like (37).

(36) Training

a. Liam saw the boy that ate a cake.

b. Liam saw the boy that Emma loved __ .

(37) Generalization

a. Liam saw the boy that Emma gave a cake to __ .

SLOG also tests the interpretation of novel gap positions in *wh*-questions. As with relative clauses, subject and direct object-extracted questions are provided in training (38), and the generalization set contains indirect object-extracted questions (39).

(38) Training

a. Who ate a cake?

b. Who did Emma love __?

(39) Generalization

a. Who did Emma give a cake to _?.

In a *wh*-question (38b), a *wh*-filler (who) in the initial position of the clause is interpreted as if it occupied the gap (again indicated with an underscore) in the direct object position of the verb *love*.

5.2.4 Novel *wh*-questions

While the previous category targets an unseen gap position (indirect object), SLOG further assesses extraction generalizations that involve familiar gap positions — subject and direct object — paired with verb types that have never been observed in *wh*-questions during training. For this case, the training set contains *wh*-questions with simple transitive verbs (40) and declarative sentences with various verb types: transitive, intransitive and ditransitive. The generalization set includes five novel types of *wh*-questions that have not been observed during training, though their declarative counterparts have.

The novel *wh*-questions have varying distance between the *wh*-filler and the gap. Subject *wh*-questions, which maintain the same word order as their declarative counterparts, exhibit no gap. Questions about the direct objects of ditransitive verbs (41c), as well as questions with NPs modified by either a PP or an RC (41d),[6] have moderately long filler-gap distances. The filler-gap distance is longest for object extraction out of embedded clauses (41e).

(40) Training

(Includes also declarative counterparts with the verbs used in the questions in (41))

a. Who **saw** a cat ?

b. What did Emma **see** __?

(41) Generalization

[6] *Wh*-questions with PP- or RC-modified NPs include various constructions where modifiers appear in subjects, direct objects, or indirect objects, exhibiting an average filler-gap distance similar to ditransitive verb *wh*-questions.

a. Who froze ?

b. What was frozen ?

c. What did the boy give __ to Liam?

d. What did Max give a cat that slept __?

e. What did a boy say that Max believed that the cat saw __?

5.3 Dataset generation

Grammar and logical forms Our dataset[7] is generated from a probabilistic Synchronous Context-Free Grammar (SCFG) using Alto (Gontrum et al., 2017), which simultaneously generates the English expressions and their corresponding meaning representations. Since SCFG cannot handle logical variables (Wong and Mooney, 2007), we use a variable-free representation proposed by Qiu et al. (2022a) (42a) as an intermediate representation during generation. The variable-free LF can be deterministically postprocessed into the original COGS LF (42b) with additional information and specific constraints: (i) We rely on the word order information in the input sentence to label the Skolem constants (i.e. variables); (ii) While the variable-free LF is unable to represent binding relations correctly as pointed out by Wu et al. (2023), an additional constraint that disallows duplicate nouns enables the intended binding relations to be identified unambiguously.

(42) A cat slept. $\rightsquigarrow$

a. Variable-free LF:
`sleep(agent=cat)`

b. COGS LF:
$\texttt{cat}(x_1) \wedge \texttt{sleep.agent}(x_2, x_1)$

(43) A cat wanted to sleep. $\rightsquigarrow$

a. Variable-free LF:
`want(agent=cat, xcomp=sleep(agent=cat))`

b. COGS LF:
$\texttt{cat}(x_1) \wedge \texttt{want.agent}(x_2, x_1) \wedge \texttt{want.xcomp}(x_2, x_4) \wedge \texttt{sleep.agent}(x_4, x_1)$

In the original COGS LF, entities or events specified by the predicates are represented by indexed variables (42b). In its variable-free counterpart (42a), *sleep* denotes the sleeping event, *cat* expresses the existence of a cat entity and fills the *agent* role of the sleeping event. In this way, each predicate in the LF has a set of arguments directly connected to their thematic roles without using variables.

[7] https://github.com/bingzhilee/SLOG

Since the variable-free LF often results in a more compact LF, it has been adopted as the primary meaning representation in several prior work (Qiu et al., 2022b; Drozdov et al., 2022). We move away from this practice and keep the original COGS LF as the main meaning representation — as briefly mentioned above, the variable-free LF cannot represent binding relations accurately unless some external heuristic or constraint is introduced for disambiguation. For example, the variable-free LF in (43a) is ambiguous between the meaning of *A cat wanted to sleep* and *A cat wanted a (different) cat to sleep*, whereas the COGS LF in (43b) unambiguously represents the meaning of *A cat wanted to sleep*. While we release the SLOG dataset in both LFs and report the results using the variable-free LF in Appendix A.7 to enable comparison with existing work, we strongly recommend using the original COGS LF for evaluation on SLOG in future work.

Following COGS, our grammar implements simplified selectional restrictions, focusing mainly on animacy constraints. For instance, the subjects of unergative verbs are limited to animate entities, as in *the cat smiled*. As a result, our generated sentences may include semantically odd but syntactically well-formed sentences, such as non-edible object being the theme of *eat* or spatial incongruities like *a house in a bottle*. While these semantic limitations are unlikely to affect models trained from scratch, they may influence the performance of models that have been pretrained on naturalistic language data. It's important to note that our primary aim is to assess the extent to which models rely on compositional structural generalization to derive meaning. In line with the classic example "colorless green ideas sleep furiously" Chomsky (1957), which demonstrates that syntactic structure can be independent of semantic coherence, we argue that a model capable of compositional generalization should be able to map such sentences to an appropriate logical form as long as they are structurally well-formed.

Training and generalization sets We follow a similar sampling procedure to COGS. A total of 10,607 sentences are sampled from the probabilistic SCFG and then split into training, in-domain validation and in-domain test sets with an 8:1:1 ratio. The splits are then merged with the corresponding COGS splits. We then add 100 standalone PP-modified NPs and 100 standalone RC-modified NPs to the training set, as discussed in Section 5.2.2.

We also include what we refer to as primitive exposure examples for each ditransitive verb and verb accepting CP arguments,[8] totaling 40 primitives. These are standalone verb lexical meanings, such as, *hope* $\rightsquigarrow$ λ`a.`λ`b.`λ`e.hope.agent(e,b)` $\wedge$ `hope.ccomp(e,a)`. This results in a final training set of 32,755 examples, and 4046 in both validation and

[8]Primitive examples of these two verb types let us incorporate their infinitive forms, used in *wh*-questions, into SLOG's vocabulary.

in-distribution test sets.

For the generalization set, we use separate grammars for each generalization case. We sample 1000 examples from each of the 17 cases, yielding a total of 17,000 examples. For the training set and the generalization set, the maximum lengths of the input English sentences are 28 and 61 tokens, respectively. The maximum lengths of the corresponding output logic forms are 229 and 599 tokens. See Appendix A.5 for more details.

5.4 Experimental setup

5.4.1 Models

We evaluate the performance of Seq2Seq, autoregressive and structure-informed models on SLOG. For seq2seq, we trained a Transformer model on SLOG from scratch (*vanilla Transformer* henceforth; Vaswani et al. 2017); and finetuned a pretrained Transformer model (T5; Raffel et al. 2020) that has demonstrated strong performance on multiple compositional generalization tasks (Herzig et al., 2021). We also finetuned LLaMa, a recently released pretrained autoregressive Transformer (Touvron et al., 2023), on SLOG.

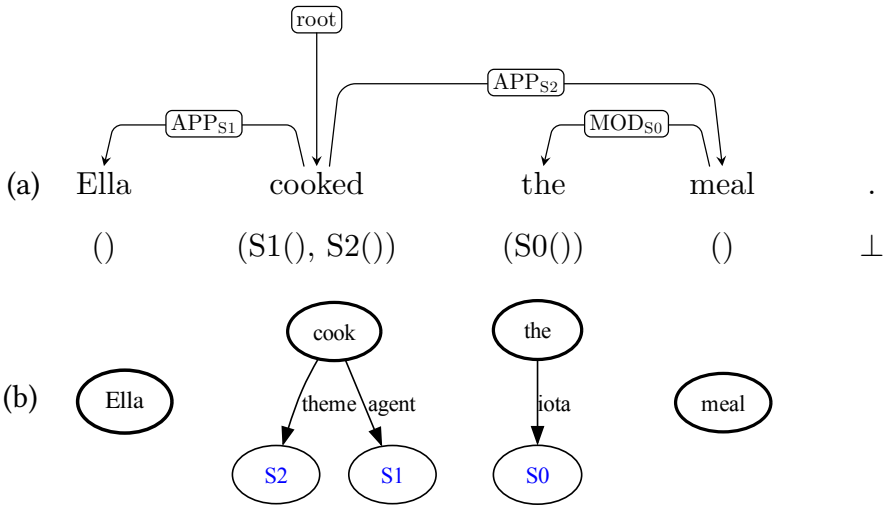


Figure 5.2: Example of an AM dependency tree: (b) displays the supertags assigned to each token, while (a) presents the dependency tree connecting them.

To provide a valuable point of comparison, we finally evaluate a structure-informed model: AM-Parser (Groschwitz et al., 2018), which achieves near-perfect accuracy on COGS (Weißenhorn et al., 2022). This allows us to measure how closely Transformer-based models can approximate the performance of a parser that explicitly incorporates compositional biases. Previous work has shown that structure-informed models perform well on composi-

tional generalization tasks, specifically those involving structural generalization (Yao and Koller, 2022). Following Weißenhorn et al. (2022), we first have the AM-Parser predict an intermediate dependency tree (as shown in Figure 5.2), and then convert it to a graph-based representation of the SLOG logical form. The AM dependency tree labels each token with a *supertag*, a small graph as illustrated in Figure 5.2b, which captures the lexical meaning of each word. The tree's edges (Fig. 5.2a) represent the compositional structure of the sentence, which specifies how the meaning of the sentence is recursively computed from the supertags. For example, the second supertag in Figure 5.2b represents the meaning of *cooked* in the sentence *Ella cooked the meal.* The blue markers "S1" and "S2" indicate that two arguments are needed to fill the agent and theme roles of *cook*.

We use the A* AM-parser from Lindemann et al. (2020) for our experiments, as it yields the best overall results compared to alternative versions of AM-parser, such as the one in Groschwitz et al. (2018).[9]

Hyperparameters The architecture of the vanilla Transformer model is the same as in original COGS, which consists of 2 encoder and 2 decoder layers, 4 attention heads per layer, and a feedforward dimension of 512. We use the best combination of hyperparameters from Csordás et al. (2021) on COGS: a learning rate of 0.0001 with no label smoothing, warmup, or early stopping. Absolute positional embeddings with down scaling scheme (He et al., 2015; Csordás et al., 2021) is used due to stability issues observed with relative positional embeddings in recursive depth generalization cases, a similar phenomenon also noted in Csordas and colleague's experiments. Models are trained for 50k steps with a batch size of 128.

For the T5 experiments, we finetune T5-base[10] using a learning rate of 1.5e-5 and no label smoothing, warmup or early stopping. We finetune the model for 50k steps using a batch size of 2048.

For the LLaMA experiments, we finetune `llama-7b-hf` with LoRA Hu et al. (2021).[11] We set the learning rate to 3e-4, LoRA rank to 8, alpha to 32 and dropout to 0.1. We finetune the model for 5K steps with a batch size of 64, with 100 warmup steps and no label smoothing or early stopping. We apply LoRA to W_q and W_v weight matrices in the model.

All our experiments were run five times, using different random seeds. The final checkpoints from each run were used for evaluation on both the in-domain test and out-of-domain generalization sets.

[9]For a more detailed discussion on alternative AM-parser models, please refer to Section 5.5.3.
[10]`https://huggingface.co/t5-base`
[11]Low-Rank Adaptation of Large Language Models: `https://github.com/tloen/alpaca-lora`

5.4.2 Evaluation metric

Most studies report exact match accuracy on COGS. This metric has two limitations that may lead to an underestimation of a model's generalization capacity. First, because the COGS LF is conjunctive, any reordering of the conjuncts are semantically equivalent; yet, under exact match accuracy, only a single order is considered correct. Second, the COGS LF uses Skolem constants with a naming scheme tied to the linear indices of phrasal heads in the input. For example, in (44a), the constant saturating `baby` is x_3 because, assuming 0-indexing, *baby* appears in linear position 3 of the English expression *What did the baby eat?*. While a commitment to a systematic naming scheme is necessary for consistent evaluation, different naming schemes up to the renaming of the constants in the gold LF yield equivalent LFs (e.g., (44a) vs. (44b)). Such LFs would be considered incorrect under exact match.

To incorporate semantic equivalence up to conjunct reordering and constant renaming, at evaluation time, we alphabetically sort the conjuncts of the gold LFs, and subsequently index variables based on their appearance order in the sorted LFs. The same modifications are applied to the model output. This process results in the reformatted output as shown in (45); applying these modifications to (44a) and (44b) yields the same outcome. Then, computing the exact match on these postprocessed LFs captures the targeted semantic equivalence.

(44) Gold LF and model-predicted LF for *What did the baby eat?*:

a. Gold: `eat.theme (`x_4`, ?) ∧ eat.agent (`x_4`, `x_3`) ∧ baby (`x_3`)`

b. Out: `eat.agent (`x_3`, `x_6`) ∧ eat.theme (`x_3`,?) ∧ baby(`x_6`)`

(45) Re-indexed and re-ordered version:

a. `baby (`y_2`) ∧ eat.agent (`y_1`, `y_2`) ∧ eat.theme (`y_1`, ?)`

This reformatted exact match metric is used for all results reported in the main text; see Appendix A.6.1 and Table A.12 for more details.

5.5 Results

Overall, seq2seq Transformers, both trained from scratch and pretrained, display low accuracy on SLOG (Figure 5.3), in line with earlier studies on structural generalization in seq2seq models (Yao and Koller, 2022). This is also the case for the more recent autoregressive Transformer LLaMa, whose performance is similar to that of T5.

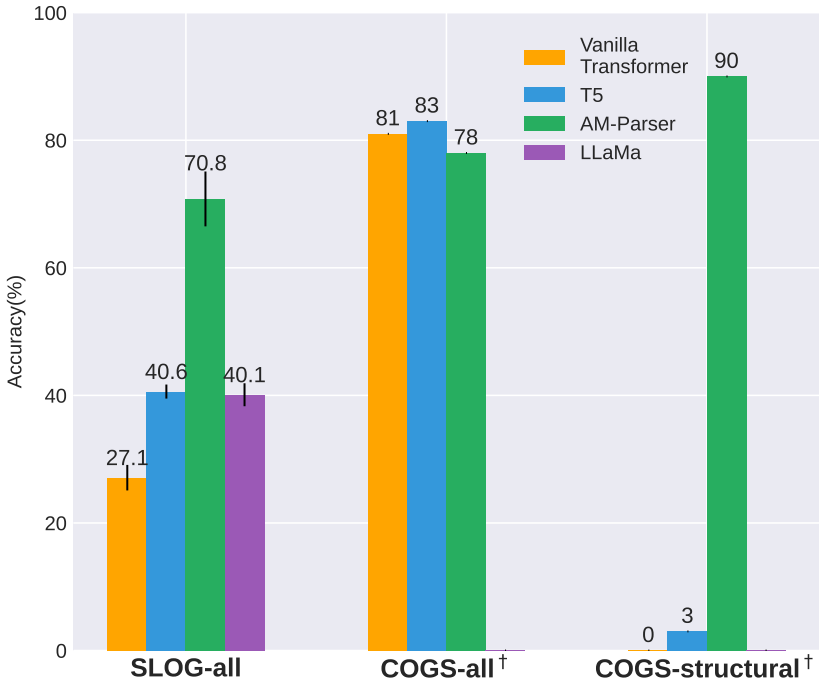


Figure 5.3: Accuracy on SLOG, with error bars indicating variations across five runs. We also show the best published results on COGS (indicated with †), as reported in Yao and Koller (2022).

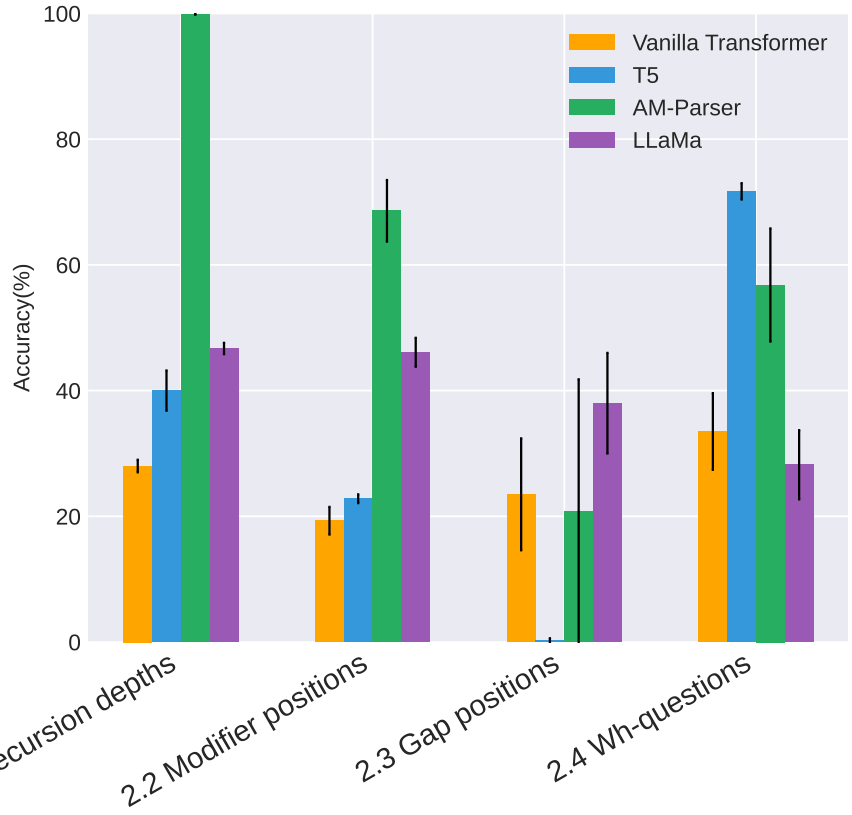


Figure 5.4: Aggregate accuracy on SLOG by generalization category, with error bars denoting the variations across generalization cases within each category over five model runs.

As Figure 5.3 shows, high accuracy on the full COGS dataset, where 86% of the generalization cases are lexical, can obscure low performance on structural generalization, highlighting the need for the expanded structural generalization tests included in SLOG.

SLOG additionally reveals weaknesses in the AM-Parser that COGS did not. While AM-Parser achieves 90% accuracy on the structural generalization subset of COGS (Figure 5.3), it faces systematic difficulties with several generalization types introduced in SLOG (Figure 5.4). We provide a detailed discussion of these difficulties in Section 5.5.3.

Performance varied substantially across generalization categories (Figure 5.4); in particular, all models achieve near-perfect accuracy on *Active subject wh-questions* and *Shallower PP recursion*. These cases were the least structurally complex in their respective categories (§5.2.3 and §5.2.1). We highlight several error patterns in individual generalization cases in more detail in the remainder of this section; see Appendix A.6 for full results and additional error analysis.

5.5.1 Unobserved depth and length both affect depth generalization

The maximum depth observed in training was four levels of embedding for all three recursive structures tested. All models achieve greater than 90% accuracy on unseen shallower PP recursion (three levels of embedding). A considerable lower performance is observed for Seq2Seq models with shallower tail CP recursion (<61%); in particular, the vanilla Transformer consistently fails to generalize to shallower center embedding, with zero accuracy overall. Transformer models show systematically lower performance on deeper recursions (5-12 levels of embedding), whereas the structure-informed parsing model is robust to depth variation.

	Vanilla Transformer	T5	LLaMa	AM parser
Within max training length				
PP recursion	29.3	37.0	46.0	100.0
Tail CP recursion	3.0	17.7	40.2	100.0
Center embedding	0.0	0.0	0.0	100.0
Beyond max training length				
PP recursion	0.0	0.0	0.0	100.0
Tail CP recursion	0.0	0.0	0.0	100.0
Center embedding	0.0	0.0	0.0	100.0

Table 5.3: Mean accuracy (%) on unseen deeper recursion cases within and beyond the range of training output lengths (maximum training output = 229 tokens).

We investigate the relation between length and depth generalization further by dividing the deeper depth generalization cases into examples that are shorter vs. longer than the maximum output length observed in training (229 output tokens). Results are shown in Table 5.3. Both the vanilla Transformer and two pretrained models are unable to generalize to examples longer than the maximum output length observed in training; this result is consistent with the difficulty of length extrapolation observed in the literature (Hupkes et al., 2020; Anil et al., 2022). Length extrapolation does not capture the full story, however: their performance is limited even when the length of the generalization examples fall within the range of observed output lengths. This indicates that unobserved depth indeed plays a role in these models' poor generalization to deeper structures, in addition to known difficulties in length generalization.

5.5.2 Unobserved long-distance dependencies make generalization difficult

Generalizing to subject modification (both PP and RC) is one of the most challenging cases in SLOG, Seq2seq models achieve near-zero accuracy, even with the additional cue from the standalone modified NPs that modification can appear outside of object positions. This challenge echoes previous findings on COGS (Akyurek and Andreas, 2021; Zheng and Lapata, 2022; Yao and Koller, 2022). The remainder of this subsection focuses on the analysis of PP modification cases, but similar patterns are observed for RC modifiers, which we discuss in Appendix A.6.2.

Common error patterns across vanilla Transformer and two pre-trained models reveal a model bias towards shorter predicate-argument dependencies, which partly explains the difficulty of this generalization case. For instance, in sentences like *A* **cat** *on the mat* **froze**, models often misinterpret the closer NP *the mat* as the subject of *froze*.

A further breakdown of the modifier generalization performance (Table 5.4) illustrates the difficulty of long-distance dependencies clearly. As discussed in Section 5.2.2, the sub-cases in indirect object modification feature predicate-argument dependencies of varying distance. We can see that generalization examples involving long predicate-argument dependency (i.e., there is an intervening non-argument NP between the predicate and the argument) tend to be more difficult for all models. However, the vanilla Transformer and pre-trained models show a stronger bias towards linearly adjacent predicate-argument structures.

For both constructions involving long predicate-argument dependencies, indirect object position seems less challenging than subject position. A possible explanation is that the

Generalization cases	Long pred-arg dependency?	Vanilla Transformer	T5	LLaMa	AM parser
Sub-case: Passive indirect objects **A fish was given** to [a cat on the mat]$_{\text{iobj}}$.	✗	95.5	97.5	98.2	93.6
Sub-case: Indirect object in PP datives Emma **gave a fish** to [a cat on the mat]$_{\text{iobj}}$.	✗	22.9	50.5	75.5	100.0
Sub-case: Indirect object in double object datives Emma **gave** [a cat on the mat]$_{\text{iobj}}$ **a fish**.	✓	4.5	9.7	36.3	77.9
Subject [**A cat** on a mat]$_{\text{subj}}$ **ate** a fish.	✓	0.0	0.8	28.9	57.6

Table 5.4: Performance of PP modification generalization broken down by construction. Bold orange words denote long predicate-argument dependencies, while bold black words indicate short ones.

former has a closer surface resemblance to direct object modification — modifiers attach to an immediate post-verb NP. Indeed, we observe that a higher proportion of indirect object modifications are partially correct; models correctly predicted the PP-modified NP, but erred in the argument structure.

We furthermore note that the results in Table 5.4 also show lower performance of Transformer models for *Indirect object in PP datives* compared to *Passive indirect objects*, although neither subcase introduces long predicate-argument dependencies. The predominant error pattern in the former subcase is the incorrect attachment of PP modifiers to the direct object NP. For example in (46b), NP inside the modifier *on the mat* denoted by x_9 was attached to *a fish* instead of *the cat*. This suggests that Transformers additionally apply the incorrect modification rule "attach PPs to NPs in immediate post-verb position", which is compatible with the training data but does not lead to correct generalization.

(46) Gold LF and model-predicted LF for *Emma gave a fish to the cat on the mat*:

a. Gold: `*cat` (x_6); `*mat`(x_9);
`give.agent` $(x_1$,`Emma)` $\wedge$ `give.theme` (x_4, x_3) $\wedge$ `give.recipient` $(x_1, x_6)\wedge$
`fish`(x_3) $\wedge$ **`cat`**`.nmod.on` $(\mathtt{x}_6, x_9)$

b. Out: `*cat` (x_6); `*mat`(x_9);
`give.agent` $(x_1$,`Emma)` $\wedge$ `give.theme` (x_4, x_3) $\wedge$ `give.recipient` $(x_1, x_6)\wedge$
`fish`(x_3) $\wedge$ **`fish`**`.nmod.on` $(\mathtt{x}_3, x_9)$

5.5.3 Gap generalizations are challenging for all tested models

All tested models encounter significant difficulties with gap constructions, as evidenced by their low accuracy and considerable variability across runs. In the case of indirect object-extracted relative clauses (47), a common error pattern emerges across all models: they

tend to mirror the training pattern of direct object-extracted RCs, as demonstrated by the incorrect output (47b). In contrast, when handling *wh*-questions, the models show distinct difficulties, revealing varied error patterns.

(47) Input: Ella cooked the servant that Emma gave a tool to __.

a. Gold: *servant(x_3);cook.agent(x_1, Ella) ∧ cook.theme(x_1, x_3) ∧ servant.nmod(x_3, x_6) ∧ give.agent(x_6, Emma) ∧ give.theme (x_6, x_8) ∧ give.recipient(x_6, x_3) ∧ tool (x_8)

b. Models output: *servant(x_3);cook.agent(x_1, Ella) ∧ cook.theme(x_1, x_3) ∧ servant.nmod(x_3, x_6) ∧ give.agent(x_6, Emma) ∧ give.theme (x_6, x_3) ∧ give.recipient(x_6, x_8) ∧ tool (x_8)

Direct and indirect *wh*-questions The vanilla Transformer and LLaMa frequently misinterpret the theme role in direct object *wh*-questions. For example, they often fail to map *wh*-words to '?' as illustrated in (48b):

(48) Input: What did Emma sell to Liam ?

a. Gold:sell.theme (x_3, ?)∧ sell.agent (x_3, Emma) ∧ sell.recipient(x_3,Liam)

b. Output of vanilla Transformer and LLaMa:
sell.theme (x_3, x_3) ∧ sell.agent (x_3, Emma)∧ sell.recipient(x_3,Liam)

c. AM parser's output:
sell.agent (x_3, ?) ∧ sell.theme (x_3, Emma)∧ sell.recipient(x_3,Liam)

This error pattern can be traced back to frequency of the subsequences in the training data. Three types of tokens can appear post-comma in the output LF space: x, ? (denoting *wh*-words), or a proper noun (PropN), such as Emma. The subsequence theme(x_i, x_j) is 20 times more frequent than theme(x_i,?) and theme(x_i,PropN). This discrepancy does not affect all models equally; in fact, T5 can generalize correctly for some constructions despite this skewed label distribution, achieving near-perfect accuracy for direct object *wh*-questions. However, when it comes to less frequent constructions — indirect object *wh*-questions, T5 overgeneralizes. In 94.6% of these cases, it erroneously produces the observed direct object *wh*-questions pattern theme(x_i,?), instead of the correct but unseen recipient(x_i,?). This observation aligns with the findings of Wu et al. (2023); Yao and Koller (2022), who noted that the decoder of Transformer models tends to exhibit a heavy bias towards generating observed n-grams.

AM-Parser shows considerable fluctuation in performance across different runs on the indirect and direct object *wh*-questions cases, with accuracies ranging from 0 to 80

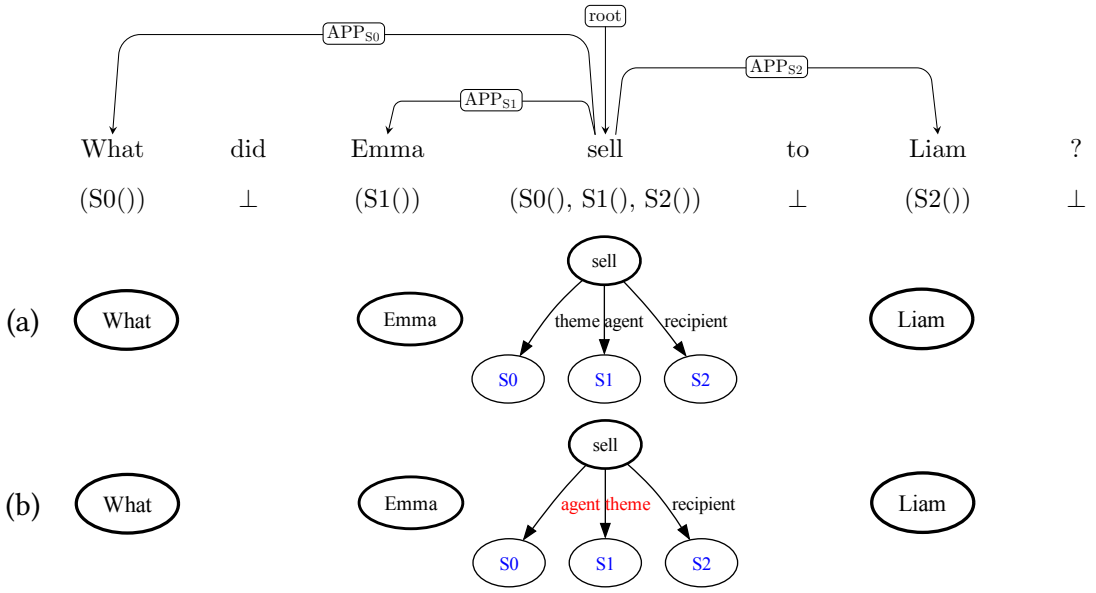


Figure 5.5: AM dependency tree for a direct object *wh*-question. (a) displays the gold supertags and (b) shows the incorrect predicted supertags.

depending on the random seed. This is because at the bottom of its compositional process, the AM-Parser predicts the lexical meaning for each token in the sentence (*supertag*). In these generalization types, the gold meaning representations in the test set require supertags that are infrequent in training.

We show an example of AM dependency trees for an *direct object wh-question* sentence in Figure 5.5, with gold supertags in Figure 5.5a and predicated supertags in Figure 5.5b. The issue here is that the model predicts the wrong supertag for *sell*, treating *What* as its `agent` instead of `theme`, and *Emma* as its `theme` rather than `agent`, which results in the erroneous output LF as shown in (48c). The AM-Parser is limited to using supertags that it observed at training time (possibly with different node labels to accommodate novel lexical material). In this case, the correct supertag was actually present in the training data, but it was much less frequent than the one in Figure 5.5b. We conjecture that the AM-Parser was overly sensitive to the supertag distribution in the training data in this case, pointing to a further architectural limitation.

Thus, while the AM-Parser can compensate the distribution shift of the meaning representations as a whole, SLOG exposes its weakness to distribution shifts in the lexical supertags.

***Wh*-questions with long movement** All models achieve very low accuracy when generalizing to longer filler-gap dependency across CPs.

In example (49b), we show an example of a *wh-question with long movement*, with its gold meaning representation (49a) and the most common errors produced by Transformer-based models. As shown in (49b), the vanilla Transformer commonly misinterprets the complementizer *that* (corresponding to ccomp in the LF) as a relative pronoun (nmod). Additionally, it tends to interpret the *wh*-word as the direct object of the CP verb, *e.g., say*. In the most common errors for T5 and LLaMa (49c), the whole gap conjunct (paint.theme(x_7, ?)) is missing, revealing their difficulties in establishing long-range filler-gap dependencies between the initial *wh*-word and the embedded gap position.

(49) Input: What did Liam say that the bear painted __ ?

a. Gold: *bear(x_6); say.agent(x_3,Liam) ∧ say.ccomp(x_3,x_7) ∧ paint.agent(x_7,x_6) ∧ paint.theme(x_7,?)

b. Output of vanilla Transformer: *bear(x_6); say.agent(x_3,Liam) ∧ say.theme(x_3,?) ∧ say.nmod(x_3,x_7) ∧ paint.agent(x_7,x_6) ∧ paint.theme(x_7,?)

c. Output of T5 and LLaMa: *bear(x_5); say.agent(x_3,Liam) ∧ say.ccomp(x_3,x_7) ∧ paint.agent(x_7,x_5)

The AM parser fails on all test instances in the case of *wh-questions with long movement*. We present a predicted AM dependency tree for such a sentence in Figure 5.7, contrasted with the corresponding gold standard AM dependency tree in Figure 5.6. Notably, for *wh*-questions with long movement, the required dependency trees are nonprojective, as illustrated in Figure 5.6: the edge from the embedded verb to the *wh*-pronoun (the edge snapped -> Who) crosses the matrix verb (root -> appreciate). However, the A* AM-Parser used in our study only supports projective dependency trees, leading to incorrect prediction of sentence structure as shown in Figure 5.7.[12]

Note that the A* AM-Parser's limitation to projective structures is shared by many other compositional semantic parsers. For instance, the LeAR model of Liu et al. (2021a) uses phrase-structure trees as compositional structures. Similarly, the CSL-T5 parser of Qiu et al. (2022a) uses phrase-structure trees during the data augmentation process. Since phrase structure trees are equivalent to projective dependency trees, these parsers are likely to

[12]Instead of the A* parser, one could instead use the fixed-tree decoder of Groschwitz et al. (2018), which is capable of predicting non-projective AM dependency trees. This parser achieves nonzero accuracy (36%) on *wh*-questions with long movement, confirming our hypothesis that the projectivity is the issue. However, the A* parser outperforms the fixed-tree decoder on most other generalization types, which is why we only report its results in the main body of the paper. The transition-based AM-Parser of Lindemann et al. (2020) can also predict non-projective trees, but uses a different probability model that is incompatible with the training algorithm of Groschwitz et al. (2021) that we use here.

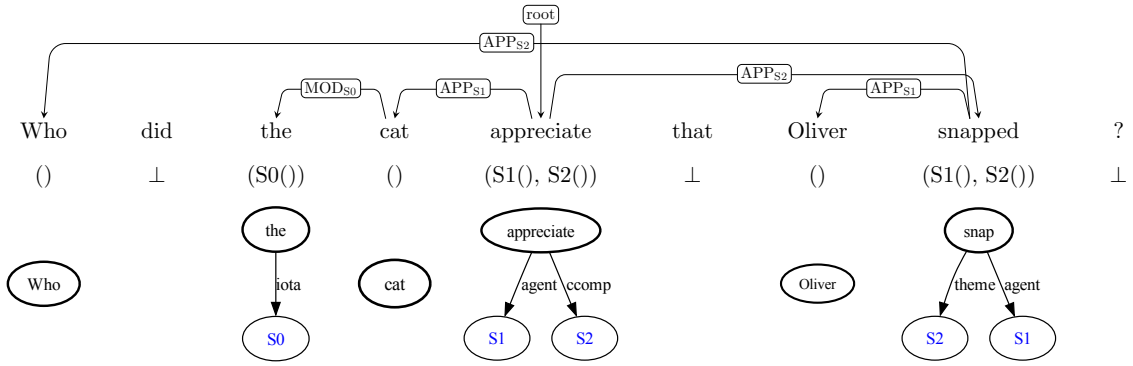


Figure 5.6: Example of gold AM dependency tree for *wh*-questions with long movement

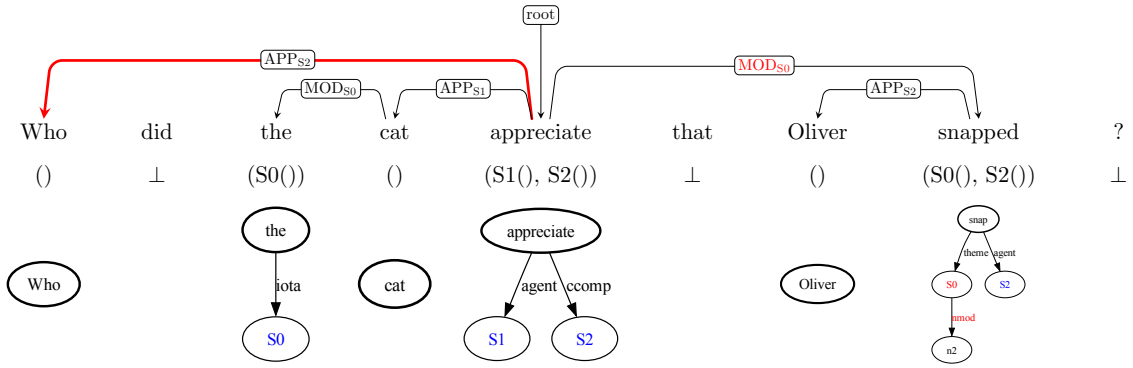


Figure 5.7: Example of predicted AM dependency tree for *wh*-questions with long movement

encounter similar difficulties on SLOG. Thus, this specific type of generalization can serve as a diagnostic tool to identify structural limitations in compositional semantic parsers.

5.6 Conclusion

Transformer-based models, despite lacking explicit symbolic representation, have demonstrated a remarkable ability to acquire nuanced syntactic representations, enabling them to handle structure-sensitive phenomena effectively, as discussed in Chapter 4. To further probe the extent to which their performance is driven by genuine syntactic generalization, aligned with symbolic compositional rules, as opposed to relying on structural similarity-based memorization derived from their training data, we introduced SLOG. This semantic parsing challenge set expands upon the COGS benchmark, and specifically targets structural generalization, which is often underrepresented in current compositional generalization benchmarks.

Using SLOG, we assessed the structural generalization capacity of Transformer models (both pretrained and trained from scratch), as well as AM-Parser, a structure-informed parsing model. While all models achieve good overall accuracy on COGS ($\geq$ 78%), their performance on SLOG is substantially lower. This was particularly evident for Transformer models, which scored below 41%, lagging behind the structure-informed parser (70.8%) by a wide margin. This performance discrepancy between SLOG and COGS illuminates the notable gap between models' lexical and structural generalization abilities.

Prior studies have shown that RNN models often struggle with learning complex long-range relations from simpler formal languages (Avcu et al., 2017; Mahalunkar and Kelleher, 2019). Our results on SLOG reveal that unseen long-distance predicate-argument dependencies pose considerable difficulty for Transformer-based models as well (§5.5.2). Additionally, these Transformer models struggle with deeper recursive constructions. Our results corroborate the observations of Hupkes et al. (2020) and Lakretz et al. (2021a), and further highlight challenges posed by unobserved deeper patterns, which persist beyond the recognized issue of length extrapolation (§5.5.1). On the other hand, the AM-Parser, despite its stronger overall performance (70.8%), displays categorical failures on gap generalization due to its inherent parser design limitations (§5.5.3).

These findings underscore the utility of SLOG in exposing the limitations of current semantic parsing models, which have previously been claimed to achieve good compositional generalization. SLOG thus can serve as a useful analytic tool for guiding future improvements. Furthermore, these results indicate that while Transformer-based models can approximate compositional behavior to a certain extent, they do not seem to rely on the kind of syntactic generalization rooted in systematic compositional rules. This insight lends support to the hypothesis that the Transformer model's ability to leverage hierarchical structures for nuanced syntactic generalization, as explored in Chapter 4, might be more attributable to

structural similarity-based analogies at the lexico-categorical abstraction level, rather than the internalization and application of systematic grammatical rules. This enables the models to handle a sophisticated form of language productivity; however, they falter when faced with novel linguistic structures that require the induction of systematic compositional rules.

The evaluation conducted with the SLOG challenge set represents only the first step — behavioral level — of our integrated three-level analysis framework as detailed in Chapter 4. This study thereby lays the groundwork for future research, particularly aimed at understanding what makes structural generalization so hard for Transformer models. The logical progression would be to advance to the representational and functional levels of analysis, using probing classifiers and causal intervention methodologies to delve into the model's difficulties with SLOG.

Part IV

Conclusion

Chapter 6

Conclusions and perspectives

6.1 Conclusions

This dissertation explored the abstraction capabilities of Transformer language models for syntactic processing. We sought to determine if these models rely mainly on surface-level patterns from their training data, or if they also implicitly construct abstract syntactic rules. Our research had two main objectives: first, to assess the potential of the autoregressive Transformer model as an explanatory tool for human syntactic processing; and second, to enhance interpretability methods for Transformer-based models.

Our research makes two main contributions. First, we have introduced an integrated framework for assessing the linguistic capacities of Transformer-based models. Second, we applied this framework to evaluate the models on two aspects of syntactic abstraction: the capacity to represent hierarchical structures and the capacity to compositionally generalize observed structures. These evaluations conducted align closely with the key prerequisites specified in Section 1.1, which are essential for a computational model to serve as a credible explanatory tool for human language processing. Our findings reveal that Transformers manage to represent hierarchical structures for nuanced syntactic generalization. However, instead of relying on systematic compositional rules, they seem to lean more towards lexico-categorical abstraction and structural similarity-based analogies (§5). This study both highlights the potential of autoregressive Transformer models as explanatory tools for human syntactic processing and provides a methodological framework for their analysis and interpretability.

From a methodological standpoint, we introduce a comprehensive linguistically-informed analysis framework that builds upon and enhances recent interpretability techniques. The framework operates on three interrelated levels. First, behavioral assessment, grounded in

challenge sets that target specific syntactic phenomena, serves as the foundational layer. This level assesses whether the model meets the requirement of reflecting human grammatical behavior. Although it reveals how the model behaves in response to certain inputs, it provides limited insight into its internal representations. Addressing this, the next level uses probing classifiers to locate the distribution of relevant syntactic information within the models. With these patterns identified, we introduce causal interventions as the third layer to decipher the underlying mechanisms driving a model's behavior and to evaluate their alignment with established linguistic analyses. This sets the stage for eventually modulating the model's behavior by tweaking the core components. In essence, our methodological framework serves two primary functions: it transforms linguistic theories into actionable, testable hypotheses, and enhances our ability to interpret and even guide Transformer-based models. In doing so, this framework takes a step toward fulfilling the 'interpretability and controllability' criteria, essential for using Transformers to explain human language processing (§ 1.1).

Our findings were twofold. First, our results in Chapter 4 indicate that the Transformer model acquires remarkably nuanced representations of sentence structure, as evidenced by its strong performance on both behavioral-level tasks and measures of representational adequacy. Specifically, we curated challenge sets for subject-verb agreement across relative clauses and object past-participle agreement, which differ fundamentally in linguistic analysis despite their surface similarity. We then assessed whether the Transformer forms distinct representations for resolving these two agreements. Our heuristic-based evaluation in Section 4.2 highlights the model's strong ability for nuanced, structure-dependent generalizations that go beyond mere surface heuristics. Further exploration using probing classifiers (§4.3) shows that syntactic information is mainly linearly encoded across all token representations between the two agreeing elements within a sentence. Despite this similar agreement information distribution pattern for both types of agreement, causal interventions experiments (§4.4) indicate that the model's predictions rely on linguistically relevant cues. These cues exhibit distinct patterns for different agreement phenomena, consistent with theoretical expectations. This evidence suggests that the autoregressive Transformer LM aligns with the key prerequisites for behavioral similarity and representational adequacy, as outlined in Section 1.1. Along with our interpretability framework, this makes the model a promising tool for studying human syntactic processing.

Our second set of findings in Chapter 5 presents a contrasting narrative. When subjected to the SLOG tasks, designed to assess compositional generalization, Transformer models encountered significant difficulties. SLOG involves a semantic parsing task (i.e., mapping linguistic expressions to meaning representations). The test is designed to have a systematic

shift between training and evaluation sets, ensuring that success in the latter demands a level of compositional generalization. In this phase, we evaluated various Transformer models as well as a symbolic neural parser. While all models excelled in the in-domain test set, Transformer models, even the recent pre-trained ones, struggle to generalize to sentences with longer dependency and deeper levels of recursion — areas where the symbolic parser performs much better. This divergence from human-like generalization, which allows for the interpretation of unfamiliar frames by systematically recombining known structures, suggests that Transformer models may rely on different or possibly insufficient underlying mechanisms.

The contrasting conclusions from Chapter 4, which highlights the model's proficiency in approximating hierarchical structures, and Chapter 5, which underscores its limited compositional generalization capacity, paint a nuanced picture of Transformer models' syntactic abstraction capabilities. This suggests that Transformer-based models primarily rely on lexico-categorical abstraction and structural similarity-based analogies for syntactic representation. While this enables them to generalize over unseen sentences with familiar structures, thus handling a sophisticated form of grammatical productivity, they struggle to handle novel linguistic structures that require inducing systematic compositional rules. These results corroborate previous findings with RNN (Baroni, 2020) and offer further empirical evidence that NLMs can achieve a certain level of abstraction for grammatical productivity without being truly compositional. Overall, this study highlights both the promise and potential limitations of autoregressive Transformer models as explanatory tools for human syntactic processing, and provides a methodological framework for its analysis and interpretability.

From a linguistic and cognitive science perspective, our positive results regarding the Transformer model's ability to represent hierarchical structures challenge the theory of syntactic nativism, which emphasizes innate structural properties. Our research reveals that an autoregressive Transformer language model, when exposed to human-scale learning data and trained merely to predict subsequent words, can grasp the intricacies of hierarchical-sensitive phenomena. This implies that the complexity of human syntactic competence could potentially be derived from exposure and general-purpose learning alone, without relying on innate linguistic priors. In this context, the Transformer model can set plausible lower bounds on the learnability of such abstractions, and provide a comparative baseline for understanding human syntactic processing.

These positive results, coupled with the limitations observed in models' compositional structural generalizations, indicate that Transformer models can achieve structure-dependent generalization without systematically following compositional rules. Instead, they seem to

rely mainly on lexico-categorical abstraction and analogies based on structural similarity. This provides constructive hypotheses about the learning and implementation of linguistic structure. On the other hand, while symbolic rules and recursive structures have traditionally been viewed as fundamental to our understanding of human language processing, they might not be the sole mechanisms for effective language processing. In particular, natural languages host many productive linguistic phenomena that follow less compositional, more complex principles, such as linguistic idiosyncrasies (Dankers et al., 2022), irregular inflections, and semi-lexicalized syntactic constraints (Goldberg and Jackendoff, 2004). Moving forward, investigating how Transformer models handle these phenomena could shed light on potential alternative cognitive strategies that remain underexplored in both human cognition and machine learning, presenting novel perspectives on computational approaches for linguistic productivity beyond the conventional rule-based compositionality.

From a deep learning perspective, our research highlights both the capabilities and limitations of data-driven neural models like Transformers. While excel in tasks where vast amounts of data guide them, they seem to struggle when faced with genuine structural and compositional challenges. This prompts the question: Can we enhance the compositional capabilities of these models to boost their learning efficiency without scarifying generality? Current research trends point in this direction, with a focus on harnessing the intrinsic nature of language to refine neural network architectures. Recent efforts, such as the study of Smolensky et al. (2022) on neurocompositional computing, suggest that by merging Compositionality and Continuity principles, there is potential to bridge the gap between symbolic and neural paradigms, pushing neural language models towards more robust compositional generalization. The SLOG test we developed (§5) can be a valuable tool to measure progress and guide model development. Crucially, by aligning models with compositional principles, we move closer to mirroring human cognitive processes, which could enhance their role as tools for understanding human language processing.

6.2 Future work

This dissertation highlights the potential of autoregressive Transformer language models as explanatory tools for the theoretical study of language and human linguistic processes. We introduced a methodological framework that facilitates testing linguistic hypotheses and conducting comparative studies between model behavior and human cognition. The next logical progression is to employ the model as an explanatory tool for human syntactic processing.

Autoregressive Transformer model as an explanatory tool for language processing: One direction that we initially aimed to explore in my thesis was the comparative analysis between human judgment and neural model behaviors on the two target agreement phenomena. While we have touched upon this topic preliminarily, a detailed study has not yet been conducted due to time limitations. Psycholinguistic studies have shown that humans also make agreement errors, with plural attractors being particularly error-prone. This is traditionally attributed to the markedness of plurals, whose features are more salient than the unmarked singular form during human language processing (Bock and Miller, 1991; Eberhard et al., 2005). Our evaluation of neural language models on number agreement tasks (§4.2.4) revealed that performance drops with increasing sentence complexity (quantified by the heuristic count). This leads to pertinent questions: Do the patterns of errors echo between models and humans? Are human judgments also influenced by surface-level heuristics? Moreover, as detailed in Section 4.2.5, while models exhibited the capacity to extrapolate syntactic generalization even in semantically implausible contexts, one wonders: To what extent are rules governing linguistic structures separate from those guiding linguistic meanings?

Additionally, there is an evident human tendency to produce more accurate agreement when dealing with singular controllers[1] (Villata, 2017). This asymmetry is often linked to an inherent human bias towards producing default singular forms (Greenberg et al., 1963; Corbett and Fraser, 2000). Notably, our observations in Section 4.2.5 reveal a similar singularity-plurality asymmetry within neural language models, with potential roots in the frequency-based biases of target verbs. This observation triggers a set of compelling inquiries: How does this resonate with established biases in the human cognitive system? How do humans navigate and extract generalities from their linguistic stimuli? And, importantly, can the autoregressive Transformer language model shed light on the origins of these behaviors, especially when we manipulate model training data — like data quantity, sentence structure complexity, and verb frequencies?

Another promising avenue for future research is to correlate model predictions with behavioral data. Number agreement tasks (§3) demonstrate how the outputs of the autoregressive language model can be directly used through minimal-pair comparisons. Another approach in the literature involves using the surprisal metric, calculated as the log of the inverse of the conditional word probability (Hale, 2001; Levy, 2008). Given that in psycholinguistics, a word's surprisal linearly affects the reading time of native speakers (Goodkind and Bicknell, 2018; Hale, 2001), using surprisal as a linking function allows comparisons between model output and human reading behavior, facilitating the testing of linguistic and

[1]Controllers correspond to "cues" in our dissertation.

cognitive theories. This could, for example, offer insights into the parallels between the Transformer model's attention mechanism and human working memory during reading tasks. Recent integrative modeling approaches, such as the one by Schrimpf et al. (2021), linked neuropsychological data, behavioral responses, and computational model predictions. This establishes connections between neural activations, human responses to linguistic stimuli, and model-based surprisal values. Such endeavors can enhance our understanding of human language processing and refine our computational models to align more closely with human cognition.

Model interpretability To effectively leverage Transformer language models for explaining human language processing, there is a pressing need to further illuminate their inner workings. Our framework, rooted in linguistic analysis, employs challenge sets, probing, and causal intervention methodologies. Yet, many interpretability methods lie outside the scope of our current exploration in this dissertation.

A notable direction for further exploration is the neural-level analysis techniques, as detailed in the survey by Sajjad et al. (2022). These techniques shed light on how models organize, specialize, and redundantly store knowledge, aligning well with our objectives. For instance, Bau et al. (2018) and Dai et al. (2022) have demonstrated how understanding neurons can help control the output of a model. Furthermore, such a granular understanding can guide the optimization of model architectures, possibly minimizing the required parameters (Voita et al., 2019; Sajjad et al., 2020; Dalvi et al., 2020). Aligning these capabilities with our objective can reinforce the potential of Transformer models as explanatory tools for human linguistic behaviors.

In Chapter 5, we highlighted the challenges faced by Transformer models in compositional structural generalization. As a future endeavor, we aim to understand what makes compositional generalization difficult for NLMs. Specifically, how do Transformer models combine token-based information into representations for larger linguistic structures? Additionally, exploring recent hybrid methodologies, which blend symbolic and neural network paradigms, appears promising, for instance, neurocompositional computing by Smolensky et al. (2022).

Bibliography

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Malin Ågren and Joost Van de Weijer. 2013. Input frequency and the acquisition of subject-verb agreement in number in spoken and written french. *Journal of French Language Studies*, 23(3):311–333.

Ekin Akyurek and Jacob Andreas. 2021. Lexicon learning for few shot sequence modeling. In *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)*, pages 4934–4946, Online. Association for Computational Linguistics.

Ben Ambridge, Evan Kidd, Caroline F Rowland, and Anna L Theakston. 2015. The ubiquity of frequency effects in first language acquisition. *Journal of child language*, 42(2):239–273.

Afra Amini, Tiago Pimentel, Clara Meister, and Ryan Cotterell. 2023. Naturalistic causal probing for morpho-syntax. *Transactions of the Association for Computational Linguistics*, 11:384–403.

Aixiu An, Chunyang Jiang, Maria A. Rodriguez, Vivi Nastase, and Paola Merlo. 2023. BLM-AgrF: A new French benchmark to investigate generalization of agreement in neural networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1363–1374, Dubrovnik, Croatia. Association for Computational Linguistics.

Cem Anil, Yuhuai Wu, Anders Johan Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Venkatesh Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems*.

Emily Atkinson, Matthew W. Wagers, Jeffrey Lidz, Colin Phillips, and Akira Omaki. 2018. Developing incrementality in filler-gap dependency processing. *Cognition*, 179:132–149.

Giusepppe Attardi. 2015. Wikiextractor. `https://github.com/attardi/wikiextractor`.

Enes Avcu, Chihiro Shibata, and Jeffrey Heinz. 2017. Subregular complexity and deep learning. *CLASP Papers in Computational Linguistics*, page 20.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Moshe Bar. 2007. The proactive brain: using analogies and associations to generate predictions. *Trends in cognitive sciences*, 11(7):280–289.

Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B*, 375(1791):20190307.

Marco Baroni. 2022. On the proper role of linguistically-oriented deep net analysis in linguistic theorizing. *Algebraic structures in natural language*, pages 1–16.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2018. Identifying and controlling important neurons in neural machine translation. *arXiv preprint arXiv:1811.01157*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Adriana Belletti. 2017. (past) participle agreement. *The Wiley Blackwell Companion to Syntax, Second Edition*, pages 1–29.

Emily M. Bender, Dan Flickinger, Stephan Oepen, and Yi Zhang. 2011. Parser evaluation over local and non-local deep dependencies in a large corpus. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 397–408, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

Jean-Phillipe Bernardy and Shalom Lappin. 2017. Using deep neural networks to learn syntactic agreement. *Linguistic Issues in Language Technology*, 15.

Robert C Berwick and Noam Chomsky. 2016. *Why only us: Language and evolution*. MIT press.

Tarek R. Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kuehnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. 2017. Neural-symbolic learning and reasoning: A survey and interpretation.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the Ability and Limitations of Transformers to Recognize Formal Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7096–7116, Online. Association for Computational Linguistics.

Kathryn Bock and J. Cooper Cutting. 1992. Regulating mental energy: Performance units in language production. *Journal of Memory and Language*, 31(1):99–127.

Kathryn Bock, Kathleen M. Eberhard, J. Cooper Cutting, Antje S. Meyer, and Herbert Schriefers. 2001. Some Attractions of Verb Agreement. *Cognitive Psychology*, 43(2):83–128.

Kathryn Bock and Carol A Miller. 1991. Broken agreement. *Cognitive Psychology*, 23(1):45–93.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Marc Brysbaert, Michaël Stevens, Paweł Mandera, and Emmanuel Keuleers. 2016. How many words do we know? practical estimates of vocabulary size dependent on word definition, the degree of language input and the participant's age. *Frontiers in psychology*, 7:1116.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Thea Cameron-Faulkner, Elena Lieven, and Michael Tomasello. 2003. A construction based analysis of child directed speech. *Cognitive science*, 27(6):843–873.

Rui Chaves. 2020. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 1–11, New York, New York. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. 2021. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. In *2020 25th International conference on pattern recognition (ICPR)*, pages 5482–5487. IEEE.

Noam Chomsky. 1957. *Syntactic Structures*. Janua linguarum (Mouton, Paris).: Series Minor. Mouton.

Noam Chomsky. 1965. Aspects of the theory of syntax. *Multilingual Matters: MIT Press*.

Noam Chomsky. 1986. *Knowledge of language: Its nature, origin, and use*. Greenwood Publishing Group.

Morten H. Christiansen and Nick Chater. 1999. Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, 23(2):157–205.

Andy Clark. 2015. *Surfing uncertainty: Prediction, action, and the embodied mind*. Oxford University Press.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Greville G Corbett and Norman M Fraser. 2000. Gender assignment: a typology and a model. In *Systems of Nominal Classification (Language, Culture and Cognition 4)*, pages 293–325. Cambridge University Press.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jillian Da Costa and Rui Chaves. 2020. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 12–21, New York, New York. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4908–4926, Online. Association for Computational Linguistics.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175, Dublin, Ireland. Association for Computational Linguistics.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2018. Universal transformers. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Iria de Dios-Flores, Juan Garcia Amboage, and Marcos Garcia. 2023. Dependency resolution at the syntax-semantics interface: psycholinguistic and computational insights on control dependencies. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 203–222, Toronto, Canada. Association for Computational Linguistics.

Andrew Drozdov, Nathanael Schärli, Ekin Akyürek, Nathan Scales, Xinying Song, Xinyun Chen, Olivier Bousquet, and Denny Zhou. 2022. Compositional semantic parsing with large language models. *arXiv:2209.15003*.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. 2022. Position information in transformers: An overview. *Computational Linguistics*, 48(3):733–763.

Kathleen M Eberhard, J Cooper Cutting, and Kathryn Bock. 2005. Making syntax of sense: number agreement in sentence production. *Psychological review*, 112(3):531.

Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. 2020. How can self-attention networks recognize Dyck-n languages? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4301–4306, Online. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Jeffrey L Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7:195–225.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany. Association for Computational Linguistics.

Martin B.H. Everaert, Marinus A.C. Huybregts, Noam Chomsky, Robert C. Berwick, and Johan J. Bolhuis. 2015. Structures, not strings: Linguistics as part of the cognitive sciences. *Trends in Cognitive Sciences*, 19(12):729–743.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *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)*, pages 1828–1843, Online. Association for Computational Linguistics.

Chaz Firestone. 2020. Performance vs. competence in human–machine comparisons. *Proceedings of the National Academy of Sciences*, 117(43):26562–26571.

Jerry A Fodor and Ernest Lepore. 2002. *The compositionality papers*. Oxford University Press.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Gottlob Frege. 1948. Sense and reference. *The Philosophical Review*, 57(3):209–230.

Felix A Gers and E Schmidhuber. 2001. Lstm recurrent networks learn simple context-free and context-sensitive languages. *IEEE transactions on neural networks*, 12(6):1333–1340.

Edward Gibson and James Thomas. 1999. Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes*, 14(3):225–248.

C Lee Giles, Clifford B Miller, Dong Chen, Hsing-Hen Chen, Guo-Zheng Sun, and Yee-Chun Lee. 1992. Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation*, 4(3):393–405.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Adele E Goldberg and Ray Jackendoff. 2004. The english resultative as a family of constructions. *language*, pages 532–568.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Johannes Gontrum, Jonas Groschwitz, Alexander Koller, and Christoph Teichmann. 2017. Alto: Rapid prototyping for parsing and translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32, Valencia, Spain. Association for Computational Linguistics.

Santiago González-Carvajal and Eduardo C Garrido-Merchán. 2020. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*.

Adam Goodkind and Klinton Bicknell. 2018. Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th workshop on cognitive modeling and computational linguistics (CMCL 2018)*, pages 10–18.

Joseph H Greenberg et al. 1963. Some universals of grammar with particular reference to the order of meaningful elements. *Universals of language*, 2:73–113.

Loïc Grobol and Benoit Crabbé. 2021. Analyse en dépendances du français avec des plongements contextualisés (French dependency parsing with contextualized embeddings). In

Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale, pages 106–114, Lille, France. ATALA.

Jonas Groschwitz, Meaghan Fowlie, and Alexander Koller. 2021. Learning compositional structures for semantic graph parsing. In *Proceedings of the 5th Workshop on Structured Prediction for NLP (SPNLP 2021)*, pages 22–36, Online. Association for Computational Linguistics.

Jonas Groschwitz, Matthias Lindemann, Meaghan Fowlie, Mark Johnson, and Alexander Koller. 2018. AMR dependency parsing with a typed semantic algebra. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1831–1841, Melbourne, Australia. Association for Computational Linguistics.

Raffaele Guarasci, Stefano Silvestri, Giuseppe De Pietro, Hamido Fujita, and Massimo Esposito. 2023. Assessing bert's ability to learn italian syntax: A study on null-subject and agreement phenomena. *Journal of Ambient Intelligence and Humanized Computing*, 14(1):289–303.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Jiaqi Guo, Zecheng Zhan, Yan Gao, Yan Xiao, Jian-Guang Lou, Ting Liu, and Dongmei Zhang. 2019. Towards complex text-to-SQL in cross-domain database with intermediate representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4524–4535, Florence, Italy. Association for Computational Linguistics.

Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Trans. Assoc. Comput. Linguistics*, 8:156–171.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Second meeting of the north american chapter of the association for computational linguistics*.

Alon Halevy, Peter Norvig, and Fernando Pereira. 2009. The unreasonable effectiveness of data. *IEEE intelligent systems*, 24(2):8–12.

Michael Hanna, Roberto Zamparelli, and David Mareček. 2023. The functional relevance of probed information: A case study. In *Proceedings of the 17th Conference of the European*

Chapter of the Association for Computational Linguistics, pages 835–848, Dubrovnik, Croatia. Association for Computational Linguistics.

Marc D Hauser, Noam Chomsky, and W Tecumseh Fitch. 2002. The faculty of language: what is it, who has it, and how did it evolve? *science*, 298(5598):1569–1579.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. 2022. Transformer language models without positional encodings still learn positional information. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1382–1390, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Jonathan Herzig, Peter Shaw, Ming-Wei Chang, Kelvin Guu, Panupong Pasupat, and Yuan Zhang. 2021. Unlocking compositional generalization in pre-trained models using intermediate representations. *arXiv preprint arXiv:2104.07478*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *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)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. Zenodo.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Dieuwke Hupkes. 2020. *Hierarchy and interpretability in neural models of language processing*. Ph.D. thesis, University of Amsterdam.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Frederick Jelinek. 1998. *Statistical methods for speech recognition*. MIT press.

Jaap Jumelet, Willem Zuidema, and Dieuwke Hupkes. 2019. Analysing neural language models: Contextual decomposition reveals default reasoning in number and gender assignment. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 1–11, Hong Kong, China. Association for Computational Linguistics.

Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. 2021. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*.

Katharina Kann, Alex Warstadt, Adina Williams, and Samuel R. Bowman. 2019. Verb argument structure alternations in word and sentence embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 287–297.

Fred Karlsson. 2007. Constraints on multiple center-embedding of clauses. *Journal of Linguistics*, 43(2):365–392.

Fred Karlsson. 2010. Syntactic recursion and iteration. *Recursion and human language*, pages 43–67.

Richard Kayne. 1972. Subject inversion in french interrogatives. In *Generative studies in Romance languages*, pages 70–126. Newbury House.

Richard Kayne and Paola Benincà. 1989. Facets of romance past participle agreement. *An Annotated Syntax Reader*, 220.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2007. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. 2023. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, 82(3):3713–3744.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Najoung Kim, Tal Linzen, and Paul Smolensky. 2022. Uncontrolled lexical exposure leads to overestimation of compositional generalization in pretrained models. *arXiv:2212.10769*.

Margaret King and Kirsten Falkedal. 1990. Using test suites in evaluation of machine translation systems. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Josef Klafka and Allyson Ettinger. 2020. Spying on your neighbors: Fine-grained probing of contextual embeddings for information about surrounding words. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4801–4811, Online. Association for Computational Linguistics.

Jordan Kodner and Nitish Gupta. 2020. Overestimation of syntactic representation in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics, pages 1757–1762, Online. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Adhiguna Kuncoro, Chris Dyer, John Hale, and Phil Blunsom. 2018a. The perils of natural behaviour tests for unnatural models: the case of number agreement. *Poster presented at Learning Language in Humans and in Machines, Paris, Fr., July*, pages 5–6.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018b. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Gina R Kuperberg and T Florian Jaeger. 2016. What do we mean by prediction in language comprehension? *Language, cognition and neuroscience*, 31(1):32–59.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253.

Yair Lakretz, Stanislas Dehaene, and Jean-Rémi King. 2020. What limits our capacity to process nested long-range dependencies in sentence comprehension? *Entropy*, 22(4):446.

Yair Lakretz, Théo Desbordes, Jean-Rémi King, Benoît Crabbé, Maxime Oquab, and Stanislas Dehaene. 2021a. Can rnns learn recursive nested subject-verb agreements? *arXiv preprint arXiv:2101.02258*.

Yair Lakretz, Dieuwke Hupkes, Alessandra Vergallito, Marco Marelli, Marco Baroni, and Stanislas Dehaene. 2021b. Mechanisms for handling nested dependencies in neural-network language models and humans. *Cognition*, page 104699.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. In *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)*, pages 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas K Landauer and Susan T Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Shalom Lappin and Jean-Philippe Bernardy. 2022. *Algebraic Structures in Natural Language*. CRC Press.

Karim Lasri, Alessandro Lenci, and Thierry Poibeau. 2022a. Does BERT really agree ? fine-grained analysis of lexical dependence on a syntactic task. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2309–2315, Dublin, Ireland. Association for Computational Linguistics.

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022b. Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2019. Flaubert: Unsupervised language model pre-training for french. *arXiv preprint arXiv:1912.05372*.

Sabine Lehmann, Stephan Oepen, Sylvie Regnier-Prost, Klaus Netter, Veronika Lux, Judith Klein, Kirsten Falkedal, Frederik Fouvry, Dominique Estival, Eva Dauphin, Herve Compagnion, Judith Baur, Lorna Balkan, and Doug Arnold. 1996. TSNLP - test suites for natural language processing. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Piyawat Lertvittayakumjorn and Francesca Toni. 2021. Explanation-based human debugging of nlp models: A survey. *Transactions of the Association for Computational Linguistics*, 9:1508–1528.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Bingzhi Li, Lucia Donatelli, Alexander Koller, Tal Linzen, Yuekun Yao, and Najoung Kim. 2023a. Slog: A structural generalization benchmark for semantic parsing. *arXiv:2310.15040*.

Bingzhi Li and Guillaume Wisniewski. 2021. Are neural networks extracting linguistic properties or memorizing training data? an observation with a multilingual probe for predicting tense. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3080–3089, Online. Association for Computational Linguistics.

Bingzhi Li, Guillaume Wisniewski, and Benoit Crabbé. 2021. Are Transformers a modern version of ELIZA? Observations on French object verb agreement. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4599–4610, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bingzhi Li, Guillaume Wisniewski, and Benoit Crabbé. 2022a. How distributed are distributed representations? an observation on the locality of syntactic information in verb agreement tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–507, Dublin, Ireland. Association for Computational Linguistics.

Bingzhi Li, Guillaume Wisniewski, and Benoît Crabbé. 2022b. Les représentations distribuées sont-elles vraiment distribuées ? observations sur la localisation de l'information syntaxique dans les tâches d'accord du verbe en français (how distributed are distributed representations ? an observation on the locality of syntactic). In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, pages 384–391, Avignon, France. ATALA.

Bingzhi Li, Guillaume Wisniewski, and Benoît Crabbé. 2023b. Assessing the Capacity of Transformer to Abstract Syntactic Representations: A Contrastive Analysis Based on Long-distance Agreement. *Transactions of the Association for Computational Linguistics*, 11:18–33.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2020. Fast semantic parsing with well-typedness guarantees. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3929–3951, Online. Association for Computational Linguistics.

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7:195–212.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Chenyao Liu, Shengnan An, Zeqi Lin, Qian Liu, Bei Chen, Jian-Guang Lou, Lijie Wen, Nanning Zheng, and Dongmei Zhang. 2021a. Learning algebraic recombination for compositional generalization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1129–1144, Online. Association for Computational Linguistics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *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)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Abhijit Mahalunkar and John Kelleher. 2019. Multi-element long distance dependencies: Using SPk languages to explore the characteristics of long-distance dependencies. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pages 34–43, Florence. Association for Computational Linguistics.

Christopher D. Manning. 2015. Last words: Computational linguistics and deep learning. *Computational Linguistics*, 41(4):701–707.

Alec Marantz. 2013. Words and rules revisited: Reassessing the role of construction and memory in language. In *27th Pacific Asia Conference on Language, Information, and Computation, PACLIC 2013*. National Chengchi University.

Ana Marasovic. 2018. Nlp's generalization problem, and how researchers are tackling it. *The Gradient*.

Gary Marcus. 2018. Deep learning: A critical appraisal.

Gary F Marcus. 1998. Rethinking eliminative connectionism. *Cognitive psychology*, 37(3):243–282.

Gary F Marcus. 2003. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. Association for Computational Linguistics.

Michael McCloskey. 1991. Networks and theories: The place of connectionism in cognitive science. *Psychological science*, 2(6):387–395.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2018. Revisiting the poverty of the stimulus: Hierarchical generalization without a hierarchical bias in recurrent neural networks. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 2093—2098, Austin, TX.

Richard Thomas McCoy, Jennifer Culbertson, Paul Smolensky, and Géraldine Legendre. 2021. Infinite use of finite means? evaluating the generalization of center embedding learned from an artificial grammar. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Warren S McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133.

Richard Meyes, Melanie Lu, Constantin Waubert de Puiseau, and Tobias Meisen. 2019. Ablation studies in artificial neural networks. *arXiv preprint arXiv:1901.08644*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari.

Richard Montague. 1974. English as a formal language. In Richmond H. Thomason, editor, *Formal Philosophy: Selected Papers of Richard Montague*, pages 188–222. Yale University Press, New Haven, London.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Aaron Mueller, Yu Xia, and Tal Linzen. 2022. Causal analysis of syntactic agreement neurons in multilingual language models. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 95–109, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. 2021. Refining targeted syntactic evaluation of language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3710–3723, Online. Association for Computational Linguistics.

Andrew Y Ng. 2004. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78.

OpenAI. 2023. Gpt-4 technical report.

Daniel W Otter, Julian R Medina, and Jugal K Kalita. 2020. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624.

Barbara H. Partee. 1984. Compositionality. In F. Landman and F. Veltman, editors, *Varieties of Formal Semantics*, pages 281–311. Dordrecht: Foris.

Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic books.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Steven Pinker and Alan Prince. 1988. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28(1-2):73–193.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R Bowman. 2020. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? *arXiv preprint arXiv:2005.00628*.

Peng Qian, Tahira Naseem, Roger Levy, and Ramón Fernandez Astudillo. 2021. Structural guidance for transformer language models. In *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)*, pages 3735–3745, Online. Association for Computational Linguistics.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. 2022a. Improving compositional generalization with latent structure and data augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4341–4362, Seattle, United States. Association for Computational Linguistics.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Tianze Shi, Jonathan Herzig, Emily Pitler, Fei Sha, and Kristina Toutanova. 2022b. Evaluating the impact of model scale for compositional generalization in semantic parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9157–9179, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

John C Raven. 1941. Standardization of progressive matrices, 1938. *British Journal of Medical Psychology*.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. 2018. Can LSTM learn to capture agreement? the case of Basque. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 98–107, Brussels, Belgium. Association for Computational Linguistics.

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online. Association for Computational Linguistics.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online. Association for Computational Linguistics.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101, Copenhagen, Denmark. Association for Computational Linguistics.

Douglas Roland, Frederic Dick, and Jeffrey L Elman. 2007. Frequency of basic english grammatical structures: A corpus analysis. *Journal of memory and language*, 57(3):348–379.

Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. 1985. Learning internal representations by error propagation.

David E Rumelhart, James L McClelland, and CORPORATE PDP Research Group. 1986. *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1: Foundations*. MIT press.

Ivan A Sag. 2010. English filler-gap constructions. *Language*, pages 486–545.

H Sajjad, F Dalvi, N Durrani, and P Nakov. 2020. Poor man's bert: Smaller and faster transformer models. arxiv 2020. *arXiv preprint arXiv:2004.03844*.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2022. Neuron-level interpretation of deep NLP models: A survey. *Transactions of the Association for Computational Linguistics*, 10:1285–1303.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

H Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop (1995)*.

Martin Schrimpf, Idan Asher Blank, Greta Tuckute, Carina Kauf, Eghbal A Hosseini, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. 2021. The neural architecture of language: Integrative modeling converges on predictive processing. *Proceedings of the National Academy of Sciences*, 118(45):e2105646118.

Luzi Sennhauser and Robert Berwick. 2018. Evaluating the ability of LSTMs to learn context-free grammars. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 115–124, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Vighnesh Shiv and Chris Quirk. 2019. Novel positional encodings to enable tree-based transformers. *Advances in neural information processing systems*, 32.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Paul Smolensky, Richard McCoy, Roland Fernandez, Matthew Goldrick, and Jianfeng Gao. 2022. Neurocompositional computing: From the central paradox of cognition to a new generation of ai systems. *AI Magazine*, 43(3):308–322.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Taiga Someya and Yohei Oseki. 2023. JBLiMP: Japanese benchmark of linguistic minimal pairs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1581–1594, Dubrovnik, Croatia. Association for Computational Linguistics.

Mirac Suzgun, Yonatan Belinkov, Stuart Shieber, and Sebastian Gehrmann. 2019. LSTM networks can perform dynamic counting. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pages 44–54, Florence. Association for Computational Linguistics.

Whitney Tabor. 1994. *Syntactic innovation: A connectionist model*. stanford university.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

R. Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Sara Veldhoen, Dieuwke Hupkes, Willem H Zuidema, et al. 2016. Diagnostic classifiers revealing how neural networks process hierarchical structure. In *CoCo@ NIPS*, pages 69–77. Barcelona.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *arXiv preprint arXiv:2004.12265*.

Sandra Villata. 2017. *Intervention effects in sentence processing*. Ph.D. thesis, éditeur non identifié.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019. Self-attention with structural position representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1403–1409, Hong Kong, China. Association for Computational Linguistics.

Yumeng Wang, Lijun Lyu, and Avishek Anand. 2022. Bert rankers are brittle: a study using adversarial document perturbations. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 115–120.

Alex Warstadt and Samuel R Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? In *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. On the practical computational power of finite precision rnns for language recognition. *arXiv preprint arXiv:1805.04908*.

Pia Weißenhorn, Lucia Donatelli, and Alexander Koller. 2022. Compositional generalization with a broad-coverage semantic parser. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 44–54, Seattle, Washington. Association for Computational Linguistics.

Gordon Wells and Allayne Bridges. 1981. *Learning through interaction: volume 1: the study of language development*, volume 1. Cambridge University Press.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019. Structural supervision improves learning of non-local grammatical dependencies. In *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)*, pages 3302–3312, Minneapolis, Minnesota. Association for Computational Linguistics.

Rodrigo Wilkens, Leonardo Zilio, and Aline Villavicencio. 2023. Assessing linguistic generalisation in language models: a dataset for brazilian portuguese. *Language Resources and Evaluation*, pages 1–27.

Guillaume Wisniewski, Lichao Zhu, Nicolas Bailler, and François Yvon. 2021. Screening gender transfer in neural machine translation. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 311–321, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Wolf. 2019. Some additional experiments extending the tech report" assessing berts syntactic abilities" by yoav goldberg. Technical report, Technical report.

Yuk Wah Wong and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 960–967, Prague, Czech Republic. Association for Computational Linguistics.

Zhengxuan Wu, Christopher D Manning, and Christopher Potts. 2023. Recogs: How incidental details of a logical form overshadow an evaluation of semantic interpretation. *arXiv preprint arXiv:2303.13716*.

Yuekun Yao and Alexander Koller. 2022. Structural generalization is hard for sequence-to-sequence models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5048–5062, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aditya Yedetore, Tal Linzen, Robert Frank, and R. Thomas McCoy. 2023. How poor is the stimulus? evaluating hierarchical generalization in neural networks trained on child-directed speech. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9370–9393, Toronto, Canada. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Hao Zheng and Mirella Lapata. 2022. Disentangled sequence to sequence learning for compositional generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4256–4268, Dublin, Ireland. Association for Computational Linguistics.

Zining Zhu and Frank Rudzicz. 2020. An information theoretic view on selecting linguistic probes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9251–9262, Online. Association for Computational Linguistics.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

Résumé

Les théories linguistiques supposent que la compétence linguistique humaine est fondée sur des structures innées et des représentations symboliques (Chomsky, 1965, 1986). Cependant, les modèles de langues basés sur le Transformeur, sans intégrer explicitement ces principes, ont atteint des performances comparables à celles de l'être humain dans de nombreuses tâches de traitement automatique de langues (TAL) (Lertvittayakumjorn and Toni, 2021; Bubeck et al., 2023). Contrairement aux modèles traditionnels basés sur l'apprentissage supervisé et des représentations symboliques tels que les arbres syntaxiques, les Transformeurs apprennent leur représentation du langage directement à partir de textes bruts, sans guidance grammaticale. Ce succès remet en question l'importance des structures hiérarchiques en traitement du langage, et suscite également des interrogations sur les mécanismes qui sous-tendent la compétence linguistique des Transformeurs. Une question clé, qui est également le cœur de cette thèse, est de savoir si les Transformeurs construisent implicitement une forme de représentation hiérarchique abstraite. La complexité de ces modèles, avec leurs nombreux paramètres, rend difficile la compréhension de leur fonctionnement interne. Bien que la recherche dans ce domaine soit en plein essor, l'étendue de la capacité d'abstraction linguistique des Transformeur reste une question ouverte. Certains travaux soulignent la compétence du modèle à capturer des nuances syntaxiques complexes, tandis que d'autres suggèrent une possible dépendance excessive à des régularités statistiques ou une simple mémorisation des données. Cette thèse vise à éclaircir si les Transformeurs représentent principalement des structures syntaxiques à travers des motifs de surface ou s'ils forment également des règles plus abstraites. En abordant cette question, nous cherchons à explorer les niveaux d'abstraction syntaxique que ces modèles peuvent atteindre et les mécanismes qui guident leurs prédictions. L'étude poursuit deux objectifs principaux : i) évaluer le potentiel d'un modèle de langue Transformeur autorégressif comme outil explicatif pour le traitement syntaxique humain ; ii) améliorer l'interprétabilité du modèle.

Nous abordons ces objectifs en examinant les abstractions syntaxiques des modèles Transformeur sur deux niveaux : leur capacité à modéliser des structures hiérarchiques, présentée dans le Chapitre 4, et leur capacité à généraliser de manière compositionnelle les structures apprises, exposée dans le chapitre 5. Ces deux aspects sont essentiels à la cognition linguistique humaine. Notre étude se concentre sur le modèle de langue Transformeur autorégressif, car son objectif de modélisation du langage est en phase avec la prédiction incrémentale des mots, caractéristique fondamental du traitement linguistique humain (Hale, 2001; Kuperberg and Jaeger, 2016; Levy, 2008)

Nous avons introduit un cadre d'analyse intégré comprenant trois niveaux interdépen-

dants: évaluation comportementale à travers des ensembles de test de défis, analyse représentationnelle à l'aide de sondes linguistiques, et analyse fonctionnelle par intervention causale. Dans le chapitre 4, nous avons déployé ce cadre pour mener une étude contrastive sur la capacité du modèle Transformer à représenter des structures hiérarchiques. L'étude se focalise sur deux phénomènes d'accord à longue distance en français : l'accord sujet-verbe à travers des propositions relatives (S-V désormais) illustré par (1), et l'accord objet-participe passé (O-PP désormais) illustré par (2).

(1) Les **chat·s** [que Noûr aime bien]$_{RC}$ **jou·ent** dans le jardin.

(2) Les **chat·s** [que Noûr a **adopté·s**]$_{RC}$ sont mignons.

Bien que les phrases (1) et (2) semblent similaires en surface, elles diffèrent fondamentalement en théorie linguistique. La première concerne un accord sujet-verbe à travers une proposition relative, tandis que la seconde met en jeu un accord entre un antécédent dans la proposition principale et un participe passé dans la relative, ce qui la résolution d'anaphore et un mouvement. Nous cherchons à évaluer la capacité du Transformer à réaliser ces deux types d'accord et si ses représentations internes reflètent cette distinction linguistique.

Partie 1: La capacité du Transformeur à modéliser des structures hiérarchiques

Évaluation comportementale Pour évaluer le comportement syntaxique du modèle, nous avons extrait, à l'aide d'heuristiques simples, deux jeux de données d'évaluation à partir de corpus Gutenberg : un pour l'accord S-V (27 582 phrases) et un pour l'accord O-PP (68 794 phrases). Dans ces phénomènes, le nom qui détermine l'accord (soit le sujet, soit l'antécédent) est nommé *indice*. Le verbe s'accordant avec cet *indice* est la *cible*. Le segment de phrase allant de l'*indice* (y compris ses dépendants) jusqu'à la *cible* (non incluse) est désignée comme le *contexte*. Ces deux jeux de données contiennent des phrases avec au moins une proposition relative entre l'*indice* et la *cible*. Après avoir pré-entraîné le modèle de langue Transformeur sur un sous-ensemble de Wikipedia, nous avons évalué sa capacité syntaxique à l'aide de tâches d'accord en nombre. La tâche demande au modèle de prédire le mot suivant à partir d'un préfixe de phrase, comme dans l'exemple "Les chats que Noûr a __". Le modèle calcule alors une distribution de probabilités pour chaque mot du vocabulaire. Nous évaluons sa performance en comparant la probabilité qu'il donne au verbe correct "adoptés" par rapport à la variante au singulier "adopté". Si la forme correct a une probabilité plus élevée, le modèle est considéré comme ayant correctement accordé l'exemple.

Alors que le modèle obtient une performance globale élevée (> 94% de précision) pour les deux tâches d'accord, notre analyse (Section 4.2.4) montre que des heuristiques simple (p. ex., accorder le verbe systématiquement avec le premier nom) peuvent produire des résultats comparables. Il est donc difficile de déterminer si le modèle s'appuie sur la structure syntaxique de la phrase ou sur des motifs superficiels. Pour évaluer cette capacité syntaxique

au-delà des motifs superficielles, nous introduisons un protocole d'évaluation basé sur des heuristiques. D'abord, nous définissons cinq heuristiques de surface qu'un modèle statistique pourrait exploiter pour prédire le nombre du verbe à partir d'indices superficiels. Chacune suppose que le verbe cible s'accorde systématiquement en nombre avec :

h1. le mot le plus proche marqué pour le nombre grammatical ;
h2. le nom le plus proche ;
h3. le premier nom de la phrase ;
h4. le nombre majoritaire exprimé dans la séquence fournie au modèle ;
h5. le nom qui précède le *que* le plus proche;

Ensuite, nous utilisons ces heuristiques pour mesurer la difficulté de la prédiction d'accord. Pour chaque phrase de test, nous comptons combien d'heuristiques prédisent correctement la forme du verbe cible, puis nous répartissons notre ensemble de test en six sous-ensembles selon ce nombre. Plus il y a d'heuristiques qui correspondent, plus la tâche de prédiction est considérée comme facile. Dans la suite de cette thèse, nos analyses se focalisent sur les cas les plus complexes (sous-groupes d'heuristiques 0 et 1).

Nos résultats montrent que la performance des modèles pour les deux tâches d'accord diminue avec la difficulté de la tâche. Toutefois, le Transformeur maintient une précision de 94% dans le cas le plus difficile pour l'accord S-V et de 76% pour l'accord O-PP. Cela met en évidence sa capacité à généraliser des informations syntaxiques au-delà des simples heuristiques. De plus, des expériences de contrôle confirment que le Transformer présente des généralisations grammaticales robustes, même en absence d'indices sémantiques et malgré un fort biais de fréquence, ce qui suggère qu'il satisfait au premier critère, la similarité comportementale, en tant qu'outil explicatif du traitement syntaxique humain.

Analyse représentationnelle Les évaluations comportementales montrent comment le modèle réagit à certains stimulis, mais elles offrent une vision limitée de ses représentations internes. Pour approfondir cette compréhension, dans la section 4.3, nous avons utilisé des sondes linguistiques pour déterminer où les informations syntaxiques sont encodées dans le modèle et s'il utilise des représentations distinctes qui reflètent les nuances théoriques de chaque phénomène d'accord. Une sonde est un classifieur entraînés à prédire des propriétés linguistiques à partir des représentations générées par le modèle. Si le Transformer a bien capturé l'information sur l'accord, alors une sonde devrait pouvoir l'identifier dans ses représentations internes.

Notre objectif est de déterminer quelles représentations de mots dans une phrase encodent le nombre grammatical de *l'indice*. Pour ce faire, chaque phrase de nos jeux de données est associée à une étiquette indiquant le nombre de *l'indice* (le sujet ou l'antécédent).

La tâche est de prédire cette étiquette à partir des représentations de mots extraites de la dernière couche du Transformeur, en utilisant un classifieur de régression logistique. Nos résultats montrent que, pour les deux types d'accord étudiés, les informations requises pour prédire la forme correcte de la *cible* sont principalement encodées dans tous les mots entre l'*indice* (où le nombre de la *cible* est spécifié) et la *cible* (où l'information est « utilisée »).

De plus, nous avons exploré la localisation de cette information sur l'accord dans l'espace de représentation du Transformeur. Pour ce faire, nous avons refait l'expérience de sondes linguistiques en utilisant cette fois des classifieurs logistiques régularisés ℓ_1 pour sélectionner les caractéristiques pertinentes. Les résultats révèlent que, pour les deux phénomènes étudiés, l'information sur l'accord est principalement encodée dans quelques dimensions fortement corrélées (moins de 10 sur 768). En outre, cette information est aussi diffusément présente, de manière redondante, dans les autres dimensions.

Analyse fonctionnelle L'approche des sondes linguistiques révèle que le modèle encode de manière similaire des informations syntaxiques pour les deux types d'accord, mais elle ne met en évidence qu'une corrélation sans établir de causalité. Elle ne clarifie donc pas la manière dont le modèle mobilise ces informations encodées pour effectuer des prédictions d'accord. Pour combler cette lacune, dans la section 4.4, nous avons introduit des interventions causales pour comprendre comment le Transformer utilise ces informations lors des prédictions d'accord et vérifier leur alignement avec les théories linguistiques établies.

Les Transformeurs utilisent un mécanisme d'auto-attention pour construire une représentation contextualisée de chaque mot, en réalisant une somme pondérée des représentations des mots précédents. Pour étudier l'impact causal de mots spécifiques sur la prédiction d'accord à la position *cible*, nous avons neutralisé leur contribution en coupant l'attention directe depuis la position *cible* vers ces mots. En comparant les prédictions avant et après ces interventions, nous mesurons l'influence causale de certains mots sur la décision du modèle dans les tâches d'accord. Nous reproduisons les mêmes tâches d'accord (§4.2.4) avec le Transformeur, mais cette fois, lors de la prédiction de la cible (et seulement à ce moment) nous supprimons l'attention directe depuis la cible vers :

i1. L'*indice* et ses dépendants;
i2. le pronom relatif *que* dans le *contexte*;
i3. i1 et i2;
i4. tous les mots dans le *contexte* sauf i1 et i2.

Bien que les sondes linguistiques révèlent une distribution similaire de l'information syntaxique pour les deux phénomènes d'accord (4.3), le Transformeur utilise ces informations de manière différente pour réaliser l'accord selon la tâche. Pour l'accord O-PP, l'antécédent

et le pronom relatif “que” sont déterminants dans la prédiction du modèle. Tandis que pour l’accord S-V, le sujet est important, mais le “que" influence peu la prédiction. Cette distinction concorde avec les analyses linguistiques théoriques, soulignant l’adéquation représentationnelle du Transformeur pour la modélisation de l’information syntaxique.

De plus, pour analyser l’impact des plongements positionnels sur la capacité d’abstraction syntaxique du modèle, nous avons réalisé des expériences d’ablation où nous avons retiré ces plongements de modèles Transformeur (autorégressif et bidirectionnel). Les résultats indiquent que ces plongements positionnels n’ont qu’un impact très limité sur la performance générale et la capacité d’abstraction syntaxique du Transformeur autorégressif. Cela est probablement dû à la capacité du modèle à inférer l’information sur l’ordre des mots via le masque d’attention incrémental.

En conclusion du Chapitre 4, nos analyses montrent que le modèle réussit à capturer les structures hiérarchiques nécessaires à une généralisation fine, basée sur la grammaire. Son excellente performance, aussi bien dans les tâches comportementales que dans l’évaluation de l’adéquation de ses représentations, suggère que le Transformer autorégressif remplit les critères essentiels (§1.1) pour servir de modèle explicatif. En combinant cela avec notre cadre d’interprétabilité, le modèle Transformer se présente comme un outil prometteur pour étudier le traitement syntaxique humain.

Partie 2: La capacité du Transformeur à généraliser de manière compositionnelle les structures observées

Le Chapitre 4 a démontré la capacité du Transformer à exploiter des relations syntaxiques pour prédire des dépendances à longue distance. Toutefois, il reste à élucider si cette généralisation syntaxique découle de sa capacité à combiner de manière compositionnelle des constituants vus lors de l’entraînement ou d’une mémorisation fondée sur des similarités structurelles. Le Chapitre 5 évalue si les Transformeurs peuvent appliquer les règles syntaxiques de manière compositionnelle pour interpréter de nouvelles structures linguistiques. Nous introduisons un test en parsing sémantique où les modèles doivent convertir des phrases anglaises en représentations sémantiques. Ce test présente une variation systématique entre les ensembles d’entraînement et d’évaluation, mettant en évidence la capacité du modèle à interpréter des structures non vues en recombinant des composants déjà rencontrés à l’entraînement. Par exemple, comme illustré dans (3), l’ensemble d’entraînement contient des propositions relatives (RC) modifiant des phrases nominales (NP) uniquement en position d’objet , tandis que l’ensemble d’évaluation teste la capacité à interpréter des RC modifiant le NP en position de sujet, comme dans (4).

(3) Entrînement
   a. Emma saw [the cat that the man held]$_{obj}$.
   b. The dog ran.
   c. the cat that the man held

(4) Généralisation

   [The cat that Emma saw]$_{subj}$ ran.

En utilisant cet ensemble de tests, nous avons évalué la capacité de généralisation structurelle de divers modèles Transformer ainsi que d'un parser symbolique. Alors que tous les modèles interprètent correctement des phrases non vues mais avec des structures familières, les Transformers, y compris les plus récents, ont des difficultés avec des phrases présentant des dépendances plus longues ou une récursion plus profonde — cas dans lesquels le parser symbolique est beaucoup plus performant. Cette différence, notable par rapport à la capacité humaine de généraliser en recombinant des structures familières, suggère que les modèles Transformeur pourraient s'appuyer sur des mécanismes sous-jacents différents ou insuffisants.

Les conclusions contrastées du Chapitre 4, qui met en avant la compétence du modèle à approximer les structures hiérarchiques, et du Chapitre 5, qui souligne sa capacité limitée à la généralisation compositionnelle, offrent une vision nuancée des capacités d'abstraction syntaxique des modèles Transformeur. Cela suggère qu'ils s'appuient principalement sur l'abstraction lexico-catégorielle et des analogies basées sur des similarités structurelles. Bien que cela leur permette de généraliser sur des phrases non vues avec des structures familières, gérant ainsi une forme sophistiquée de productivité grammaticale, ils peinent face à de nouvelles structures linguistiques qui nécessitent l'induction de règles compositionnelles systématiques. Dans l'ensemble, cette étude met en lumière à la fois les promesses et les limites potentielles des modèles Transformeur autoregressifs comme outils explicatifs pour le traitement syntaxique humain, tout en proposant un cadre méthodologique pour leur analyse et interprétabilité.

Appendix A

A.1 Neural language models

A.1.1 Hyperparameters

The LSTM model have a total of 47,900,241 parameters. A grid search was conducted for the optimal hyperparameters in the following range: batch size from {32,64}, dropout rate from { 0.0, 0.1, 0.2, 0.3 }, and learning rate from { 0.1, 0.01, 0.001, 0.0001 }. The configuration yielding the lowest perplexity score (37.1) comprises a batch size of 64, a dropout rate of 0.1, and a learning rate of 0.0001. Subsequent training of four additional LSTM models, using this optimal hyperparameter combination, yielded perplexity scores of 36.8, 36.8, 36.9, and 37.0.

Each Transformer model $\in \{\mathcal{M}, \mathcal{M}_{shallow}, \mathcal{M}_{shared}, \mathcal{M}_{nopos}$, MLM, MLM$_{nopos}\}$ has an architecture with 16 attention heads, a hidden size of 768, and feed forword dimensions of 2048. Model $\mathcal{M}$, the main focus of our study, has a total of 126,674,513 parameters. To identify optimal hyperparameters, we conducted a grid search in the range of learning rates { 0.01, 0.01, 0.02, 0.03 } and dropout rates { 0.0, 0.1, 0.2, 0.3, 0.4 }, yielding 16 combinations. The combination with the lowest perplexity of 27.0 had a learning rate of 0.02 and a dropout rate of 0.2. We further trained four more Transformer models with these parameters, achieving perplexities of 26.8, 27.0, 27.1, and 27.2.

Training was performed with stochastic gradient descent with a fixed initial learning rate of 0.02 and cosine scheduling across 100 epochs without annealing. The first epoch was dedicated to warmup with a linear incremental schedule for the learning rate. We used

	PPL	# params	# layers	lr	dropout	tied layers	use positional embedding
LSTM	$36.9_{\pm 0.1}$	47.9M	2	0.0001	0.1	False	–
$\mathcal{M}$	$27.0_{\pm 0.2}$	126.7M	64	0.02	0.2	False	True
$\mathcal{M}_{shallow}$	$37.8_{\pm 0.7}$	49.5M	2	0.002	0.0	False	True
$\mathcal{M}_{shared}$	$30.7_{\pm 0.6}$	47.8M	16	0.002	0.0	True	True
$\mathcal{M}_{nopos}$	$27.4_{\pm 0.3}$	126.7M	16	0.01	0.1	False	False
MLM	$^{\dagger}\ 5.6_{\pm 1.2}$	130.5M	16	0.02	0.2	False	True
MLM$_{nopos}$	$^{\dagger}\ 57.2_{\pm 2.3}$	130.5M	16	0.02	0.2	False	False

Table A.1: Hyperparameter configurations for each model and their corresponding average perplexity scores. † denotes pseudo-perplexity scores used for MLM evaluation (§4.5.2), not comparable with conventional perplexity scores.

batch sizes of 64 and bptt_chunk of 150, running on 8 GPUs, except during warmup when we fixed the batch size to 8. Each model was trained up to 72 hours. We trained five seeds for each best hyperparameter configuration. All experiments results are averaged across these five instances. All other Transformer-based LMs followed the same training procedure as the model $\mathcal{M}$.

A.1.2 Perplexities in language model evaluation

As discussed in Section 2.2.1, perplexity is a common metric for evaluating conventional language models, which predict the next word in a sequence based on the preceding context.

The perplexity of a LM on a word sequence $S = w_1, w_2, ..., w_t$ is computed using the preceding tokens $w_{1:i-1}$ and applying the chain rule $\sum_{i=1}^{N} \log_2 \mathbb{P}_{LM}(w_i|w_{1:i-1})$, as shown:

$$PPL(S) = 2^{-\frac{1}{N}\sum_{i=1}^{N} \log_2 \mathbb{P}_{LM}(w_i|w_{1:i-1})} \tag{A.1}$$

However, this metric doesn't apply to models trained with a masked language modeling objective. MLM predicts a masked token w_i based on its surrounding context $S_{\setminus i}$, rather than directly modeling the conditional probability $\mathbb{P}(w_i|w_{1:i-1})$. To evaluate MLMs, we use the pseudo-perplexity, computed as the average of the conditional log probabilities $log\mathbb{P}_{MLM}(w_i|S_{\setminus i})$ for each token, as proposed by Salazar et al. (2020). Given a pretrained MLM with parameters denoated as Θ, and a sequence S, we mask each token in the sequence iteratively and compute the log-probability for each word in S. The pseudo-perplexity score is defined as:

$$PPPL(S) = \frac{1}{|S|} \sum_{w \in S} \log \mathbb{P}_{MLM}(w|S_{\setminus w}; \Theta) \tag{A.2}$$

To estimate the PPPL score of our validation set $\mathcal{D}_v$, we use bootstrap sampling following Sinha et al. (2021). We draw 1000 samples five times with replacement and compute the bootstrap perplexity (BPPL):

$$BPPL_{\mathcal{D}_v} = \exp\left(-\frac{1}{N} \sum_{S \in W} PPPL(S)\right) \tag{A.3}$$

We use BPPL as the final pseudo-perplexity score for evaluating MLMs. While it's not equivalent to the perplexity metric for autoregressive language models, it allows for a direct comparison between MLMs.

A.2 Error analyses

A.2.1 Qualitative error pattern analysis

For each agreement task, we sampled 100 sentences for which the Transformer made incorrect number predictions in the majority of its 5 runs. We present common error patterns in these results to inform future experiments. In the following examples from the evaluation set, we bold the **cue** (subject or antecedent) and target verb; in each case, the Transformer predicted the opposite number from the target verb.

S-V agreement across relative clauses Only about 1.1% of the sentences (303 in total) received incorrect number predictions from the Transformer. The majority of these errors align with the five heuristics outlined in Section 4.2.4, particularly those related to local nouns or pronoun attractions. Here, we examine three main categories of errors.

The model seems to associate the conjunction "et" (*and*) with plural verb forms. For instance, in examples (5) and (6), the model incorrectly chose the plural form for the target verbs. Notably, none of these examples feature plural nouns, and all 5 heuristics defined earlier (§4.2.4) predict a singular form.

(5) Le **sentier**$_{Sg}$ qu' ils suivaient , lui et la fée , **descendait** ...
The **path** they followed, he and the fairy, **descended**$_{Sg}$ ...

(6) le **charm**$_{Sg}$ que la manière , la cadence et l'accent peuvent ajouter à un organe **apparaît**$_{Sg}$ ...
the **charm** that manner, cadence and accent can add to an organ **appears**...

In most cases of long-distance subject-verb agreement, the Transformer accurately predicted the number of the target verb, resisting the influence of local attractor nouns. However, it struggles with non-canonical constructions like the inversion of noun subjects and verbs in object relative clauses. In French, the standard word order places the noun subject before the predicate. Stylistic inversion (Kayne, 1972) allows users to reverse this order, as shown in examples (7) and (8), where the predicate is in blue and the subject in orange. When faced with these inversions, the Transformer tends to make much more errors, basing its agreement on the most recent nouns, which are the inverted subjects of the embedded relatives. This suggests that the model struggles with handling such less frequent, optional stylistic inversions phenomena.

(7) les **colons**$_{Pl}$, craignant la concurrence commerciale **qu'**allait leur faire la compagnie, **déclarèrent**$_{Pl}$ ...
the **colonists**$_{Pl}$, fearing the commercial competition that the company was going to bring them, **declared**$_{Pl}$

(8) afin que la **liqueur**$_{Sg}$, en suivant les sillons que forment les plis , **pût**$_{Sg}$ arriver à la pointe du cône
so that the **liquor**$_{Sg}$, following the grooves that the folds form, **could**$_{Sg}$ reach the tip of the cone.

Sentences with multiple instances of "que" (*that*) in the prefix seem to be particularly error-prone for the model. This holds true whether it is two relative pronouns "que", as in example (9), or a conjunction "que" followed by a relative pronoun "que", as in example (10). In the first example, the model incorrectly predicts a singular form despite the absence of any singular nouns in the prefix. In the second example, the model appears to misidentify "papiers" (*papers*) as the subject of the target verb, instead of the closer noun "lettre" (*letter*), leading to an incorrect plural prediction. These errors could be indicative of the model's difficulty in managing nested or complex syntactic structures that require a nuanced understanding of contextual and hierarchical relationships.

(9) Les **mots**$_{Pl}$ que je devine , que je sens tout près de vous **sont**$_{Pl}$ très beaux
The **words**Pl that I guess, that I feel so close to you, **are**$_{Pl}$ very beautiful.

(10) les papiers ne me paraissent pas si terribles que la **lettre**$_{Sg}$ que vous m'avez envoyée **semblait**$_{Sg}$ le faire craindre
The papers don't seem as terrible as the **letter**$_{Sg}$ that you sent me **made**$_{Sg}$ it appear to be.

O-PP agreement About 5.4% of the sentences (3 698 in total) received incorrect number predictions from the Transformer. Here, we examine two main categories of errors.

The model frequently struggled with identifying the correct head nouns in prepositional phrases. For instance, in example (11), the model incorrectly agreed the past participle with "chevalerie" (*chivalry*), the closer but incorrect noun, rather than the correct head noun "exploits". Conversely, in example (11), the model wrongly predicted a plural form, aligning with the more distant noun "personnes" (*people*$_{Pl}$) instead of the correct, closer noun "compagnie" (*company*). These errors indicate that the model has not fully grasped the structure of French prepositional phrases.

(11) Je ferai les plus fameux **exploits**$_{Pl}$ de chevalerie$_{Sg}$ qu'on ait **vus**$_{Pl}$
I'll make the most famous **exploits** of chivalry that one has **seen**$_{Pl}$

(12) Il y avait deux ou trois cents personnes$_{Pl}$ de la meilleure **compagnie**$_{Sg}$ que j'aie **vue**$_{Sg}$ en Italie.
There were two or three hundred people$_{Pl}$ from the best **company**$_{Sg}$ I've **seen**$_{Sg}$ in Italy.

(13) L'un des **instruments** les plus puissants que Dieu ait **confiés** ...
One of the most powerful **instruments** that God has **configured**...

(14) Aucun des **sentiments**$_{Pl}$ que j'ai **éprouvés**$_{Pl}$ jusque-là ne mérite le nom d'amour.
None of the **feelings**$_{Pl}$ I've **experienced**$_{Pl}$ so far deserves the name of love.

Another systematic error observed in the Transformer's predictions involves constructions "l'un des" (*one of the*) and "aucun des" (*none of the*), as illustrated in (13) and (14). In both cases, the model incorrectly matched the past participle with the quantifiers, which semantically imply singularity — either 'one' or 'none'. However, past participles should actually agree with the plural nouns that these quantifiers modify. This indicates that the model has not fully understood the intricate interplay between semantics and morphosyntax in the context of quantifier agreement.

Much like the difficulties encountered in S-V agreement, the stylistic inversion of the noun subjects (highlighted in blue) within relative clauses also complicates O-PP agreement cases for the models. They tend to leverage the grammatical number of the auxiliary verb (underlined) immediately preceding the target to predict its number. For example, in (15), all preceding nouns are plural, yet both LSTM and Transformer models predicted a singular form. A detailed analysis of this non-canonical construction (in total 1,599 sentences) shows that when the number of the intervening auxiliary differs from that of the past participle, the LSTM's accuracy drops to 42%, while the Transformer maintains an 80% accuracy rate.

(15) J'étudiai les sculptures symboliques dans les **chambres** intérieures des pagodes que n'a$_{Sg}$ **vues** nul œil profane et où une robe de brahme me permettait de pénétrer.
I studied the symbolic sculptures in the inner **chambers**$_{Pl}$ of the pagodas that no profane eye has **seen**$_{Pl}$, and where a Brahmin robe allowed me to enter.

(16) ... qu'il faut attribuer tous les **malheurs** qu'a$_{Sg}$ **éprouvés** notre belle France.
that we must attribute all the **misfortunes**$_{Pl}$ that our beautiful France has **experienced**$_{Pl}$.

A.2.2 Lexical variation

There is a noticeable lexical variation in the results. Table A.2 highlights this disparity: the top-performing verbs in both agreement tasks achieved 100% accuracy. Conversely, while

the least accurate past participles consistently scored 0%, the lowest-scoring verbs in S-V agreement still attained an accuracy rate of over 81%.

This disparity might be attributed to frequency effects. Both the occurrence (absolute frequency) and the frequency ratio of a target form relative to its competing form play roles. Typically, the more frequent a verb or past participle is, the more likely it is to be predicted correctly. In contrast, less frequent lexical items, predominantly in their plural forms, often lag behind. For instance, the verb "dits" posed challenges; its singular counterpart is 11 times more prevalent, leading the model to consistently predict the more frequent, but incorrect form. Such variations indicate that the Transformer language model may form less robust number representations for infrequent verbs and struggle when the frequency bias heavily favors one form over another.

Form	Accuracy(%)	Total Sentences	Target Occurrences	Ratio($\frac{\text{Target form}}{\text{Competing form}}$)
S-V across relatives				
Best-performing V				
serait	100	196	11426	3.3
fit	100	128	8361	5.2
eut	100	126	6101	4.2
vient	100	122	7682	2.6
furent	100	94	17318	0.3
worst-performing V				
mettaient	90	10	188	0.4
suffisent	89.3	28	270	0.2
contenait	88.2	17	698	3.2
auront	87.5	24	1363	0.3
disaient	81.8	11	150	0.2
Object-past participle				
Best-performing PPs				
fait	100	2030	112263	20.8
eu	100	961	14544	193.9
envoyé	100	272	3499	23.8
laissées	100	270	389	0.7
dit	100	246	15369	11.0
Worst-performing PPs				
dits	0.0	60	1398	0.09
mérités	0.0	29	704	4.5
crus	0.0	18	238	0.2
éveillés	0.0	11	38	0.3
désirés	0.0	10	45	0.2

Table A.2: Verbs (at least 10 sentences) yielding the highest and lowest accuracy for the Transformer-LM. 'Target Occurrences' refers to the frequency of the target form in pretraining data. 'Ratio' signifies the frequency ratio of the target form to its competing form (i.e., with the opposite number) in the pretraining data.

S-V across relatives
Worst-performing verbs
(1) Sans doute en étant sans cesse auprès de l'Empereur témoin ou collaborateur, l'on pouvait bien deviner ou préjuger les intentions qui le maîtrisaient, et les **conséquences** que l'on tirait de ce que l' on croyait à peu près savoir **mettaient** sur les traces mêmes de ce qu'il pouvait y avoir d'occulte dans sa conduite apparente . (2) Sous un ciel toujours clément , quelques aunes de toile suffisent pour vêtir le Napolitain , comme quelques **pièces** de basse monnaie qu'il gagne sans fatigue lui **suffisent** pour se procurer la nourriture (3) Toutes les **phrases** qu'elle me disait, discrètes à la fois et vives, **contenaient** autant d'interrogations sur ma vie depuis que je l'avais quittée ... (4) Une fois que le médecin aura ainsi pris position, les **conseils** qu'il donnera , non seulement sur l'hygiène mentale , mais sur l'hygiène alimentaire , musculaire , **auront** toutes chances d' être suivis; (5) Son génie éclatait, austère et convulsif, Comme celui de Dante ou de Savonarole, Les **bouches** qu'il ouvrait **disaient** d'autres paroles...
O-PP agreement
Worst-performing verbs
(1) voici les **mots** mêmes qu'il m'a **dits**, je vous les répète. (2) Votre affectation à n'en pas parler aura fait naître ces **soupçons** que j'ai si peu **mérités**, et dont je ne me consolerai jamais. (3) La belle-sœur du prince de Schwartzenberg, entendant sortir de la salle embrasée des **cris** qu'elle a **crus** poussés par sa fille aînée, ... (4) Je te l'ai dit , il faut aller vers le nord pour échapper aux **soupçons** qu'a **éveillés** ton absence. (5) Elle ne me donna pas sur cette affaire tous les **renseignements** que j'aurais **désirés**.

Table A.3: Examples of sentences for the worst performing verbs and past participles in Table A.2, the words in bold indicate the **cue-target** pairs.

A.3 Sample sentences from evaluation sets for long-distance S-V and O-PP agreements

In this section, we provide an extract of sentences used in the experiments of Chapter 4, aimed at assessing model capacity to process structure-sensitive phenomena. These sentences are organized into subsets following the heuristic-based evaluation protocol established in Section 4.2.4. All sentences are sampled directly from the evaluation sets and are presented as they are after tokenization. This format is specifically tailored for our word-based neural language models and may not adhere to standard French writing rules, for instance, elisions are separated into two words, 'l'esprit' appears as 'l' esprit'.

A.3.1 Long-distance S-V agreement

Sentences exhibiting long-distance S-V agreement in Chapter 4 refers to sentences where the main verb and its syntactic subject are separated by one or more object relative clauses. The entire evaluation set consists of 27,582 sentences (§4.2.2). The main verb and the head of its subject are highlighted in bold.

5-heuristic subset

1). Monvel , me dit-il enfin , vous avez raison , le **mariage** que je vous avais proposé **est** impossible .

2). L' **esprit** de parti qui règne ici et qui augmente par la faiblesse du gouvernement , lequel cependant fait ce qu' il peut , **rend** ce séjour de plus en plus odieux .

3). Adieu , je vous bénis , ne maudissez jamais ma mémoire ; rappelez-vous que la plus grande **douleur** que j' éprouve dans mon supplice **est** celle de mourir loin de mes enfants.

4). Ainsi donc le **mouvement** de substance que nous appelons génération , ne **doit** être attribué qu' à Dieu .

5). Et puis , il faut bien le dire , les **paroles** que répètent les perroquets **tombent** quelquefois avec tant d' à-propos , qu' ils vous ont l' air d' avoir une intelligence surprenante.

4-heuristic subset

1). Les **oeuvres** extraordinaires que ces hommes produisent , dit Goethe , **supposent** une organisation très-délicate.

2). A trois heures précises , le **cercueil** qu' on m' avait réservé **reçut** ma très viable et très vitale dépouille ,

3). Ainsi , répondit-il , le **motif** que vous me donnez **est** le seul qui vous pousse à me quitter ?

4). Monsieur , cet **ouvrage** que je vous présente vous **appartient** , puisque tout ce qui est à moi est à vous .

5). À trois heures du matin tout était en mouvement par un temps sombre et pluvieux , et les **caissons** qu' on brûlait ou qu' on faisait sauter faute de les pouvoir atteler , **ajoutaient** de sinistres lueurs et de plus sinistres détonations à cette retraite .

3-heuristic subset

1). Ainsi , Monseigneur , la **demande** que le roi d' Espagne aura faite au roi de ces quatre vaisseaux **devient** aussi inutile que le projet du Conseil des Indes

2). Il restera à savoir si les six cents **hommes** qu' on pourrait laisser dans ce fort **pourraient** s' y défendre trente ou quarante jours et attendre le retour de l' armée ;

3). Au surplus , comme le premier **but** que je me propose , le plus ardent de tous mes désirs , **est** de suivre précisément les intentions de Sa Majesté.

4). Évidemment ses adversaires l' appréciaient à sa juste valeur : depuis le commencement de la guerre française , il s' était distingué parmi les plus braves ; les précieux **services** qu' il avait rendus aux forts anglais établis sur les frontières l' **avaient** rendu légendaire parmi les Indiens .

5). À ma grande surprise , j' ai été nommé membre de l' Académie des beaux-arts de l' Institut , et si , quand j' y prends la parole de temps en temps , les **observations** que je fais sur nos usages académiques **sont** assez inutiles et restent sans résultats , je n' ai pourtant avec mes confrères que des relations amicales et de tout point charmantes .

2-heuristic subset

1). Ainsi , les **devoirs** que nous impose la famille **sont** en contradiction avec ceux que nous impose l' humanité .

2). Assurément , les **effets** qu' elle a produits jusqu' à présent **sont** relativement faibles ;

3). Depuis quelques jours , le bruit du départ de don Carlos pour l' Espagne s' était répandu à Londres , mais les **détails** qu' on donnait sur cet événement **étaient** tellement vagues et contradictoires qu' il était difficile d' y ajouter foi .

4). Herr von BethmannHollweg affirme que les **papiers** que nous avons trouvés dans les archives du ministère des Affaires étrangères à Bruxelles , **montrent** que l' Angleterre , en 1911 , était déterminée à jeter des troupes en Belgique ...

5). Mes enfants , pensez toujours que l' **homme** que vous aurez en face de vous **peut** être le père , le frère de votre camarade Brussanes , et cela retiendra , j' en suis certain , les mains trop promptes .

1-heuristic subset

1). Quand j' entrai chez Éliane , elle était seule , couchée sur une chaise longue ; ses longs **cheveux** noirs , que j' avais toujours vus bouclés avec le plus grand soin , **tombaient** en désordre sur ses épaules ;

2). À mon âge , les **promesses** que l' on fait à la raison ne **tiennent** guère .

3). Toutes les portes étaient ouvertes et sans gardes ; mais le **respect** qu' inspirait la présence des princes **suffit** seul pour empêcher le désordre et la confusion .

4). Sibylle était allée au-devant de cette recommandation , et les **instructions** que la duchesse lui transmit , en se gardant bien de lui en révéler l' origine , se **trouvèrent** superflues .

0-heuristic subset

1). Ça le réjouissait de savoir qu' on fêtait la République , et les **souvenirs** de la Révolution qu' il tenait de son père et de son grand-père , lui **revenaient** à la mémoire.

2). Les douaniers de Néphélococcygie font bonne garde : toute la **fumée** des sacrifices que les hommes offrent aux anciens dieux **est** interceptée .

3). Le public intelligent et lettré verra bien , de son côté , que les **arcanes** de l' érudition qu' il craint , respecte et méprise à la fois , ne **sont** pas si mystérieux ni si redoutables lorsque les questions sur lesquelles s' exercent les érudits sont mises au point et discutées avec simplicité .

4). Le pauvre diable avait beau faire des efforts , il ne pouvait avancer , car il était pris entre deux arbres , et les deux **bottes** de paille qu' il avait de chaque côté , l' **empêchaient** de passer .

5). Le goût italien moderne nous gagne , et la contagion est telle que les **coins** réservés aux artistes , dans ce grand bazar populaire et bourgeois qu' on vient de fermer , y **prenaient** aussi des aspects de réclame et d' étalage forain .

A.3.2 O-PP agreement

The entire evaluation set consists of 68,497 sentences (§4.2.2). The antecedent and the target past participle are highlighted in bold.

5-heuristic subset

1). À onze heures du soir , ils hallèrent en effet le Ouest-Sud-Ouest , et puis après ils tournèrent au Sud : désespérés de ne rien trouver sur les **vagues** qu' ils avaient **battues** pendant plusieurs heures , nos pilotes se décidèrent à gouverner sur Ouessant , où leurs familles devaient s' inquiéter de ne pas les avoir vus rentrer à l' heure accoutumée.

2). À part l' enivrement des premiers regards , Maurice s' était trouvé au-dessous de son attente dans la **réception** que lui avait **faite** Geneviève , et il comptait sur la solitude pour regagner le chemin qu' il avait perdu , ou du moins qu' il paraissait avoir perdu dans la route de ses affections .

3). À midi , mon excellent montagnard était de retour avec la **réponse** que le capitaine avait **écrite** devant lui , dans son bureau du quartier général où il doit , soit dit en passant , terriblement peiner , lui qui est seul là-bas pour recevoir , répondre , et parer à l' imprévu !

4). À mesure que j' ai appris à connaître le terrible et singulier gouvernement , régularisé , pour ne pas dire fondé par Pierre Ier , j' ai mieux compris l' importance de la **mission** que le hasard m' avait **confiée** .

5). toujours est-il que le visage de Bessie se couvrait d' un **voile** de tristesse que John ne lui avait jamais **vu** , et qui ajoutait à son charme , comme l' ombre ajoute au charme de la lumière .

4-heuristic subset

1). "tous ces gens-là sont venus au monde par une **incision** que l' art a **faite** "

2). serait-il possible que des cœurs brûlant d' un zèle aussi pur pour votre prospérité , pour votre gloire , eussent renoncé à des sentiments plus chers que leur **vie** , qu' ils ont tant de fois **exposée** pour vous ?

3). ne serait-ce pas plutôt une intrigue avec **quelqu'une** de la ferme que tu aurais **prise** pour Tatiana ?

4). Vous êtes en train de perdre dans ce désert les belles **manières** que vous avez **apprises** à l' Université de Californie .

5). Vous verrez que je suis digne de mettre à vos pieds le plus magnifique **amour** que jamais homme ait **offert** à une femme .

3-heuristic subset

1). Vous verrez les **mûriers** que mon grand-père a **plantés** , et le gros figuier qui est sous ma fenêtre , tout peuplé de nids de tourterelles !

2). Quant aux **impressions** que vous m' avez **confiées** , tout ce que je puis vous dire , c' est que vous êtes , je crois , dans un cas où l' on a plus besoin de conseils s' appliquant à l' âme qu' au corps .

3). Si nous sommes en paix et que notre état actuel ne soit qu' un état de mésintelligence , la France doit liquider tout le **tort** que ses corsaires vous auront **fait** .

4). –Un de vos amis , repris-je avec ironie , le meilleur même de vos amis ; je suis reconnaissant de la **place** que vous m' avez **faite** , mais cette place , je ne m' en sens pas digne .

5). À cette époque-là , les jeunes gens de la bourgeoisie tiraient une grande vanité de pouvoir montrer un **sabre** de gendarme qu' ils avaient **acheté** à quelque voyou après la fête , ou une égratignure qu' ils s' étaient faite en se mettant à la fenêtre précipitamment , pour regarder .

2-heuristic subset

1). À cette voix , qui lui rappelait les seules **affections** qu' elle eût jamais **connues** , Jane rouvrit les yeux , regarda Dolly , et lui sourit en murmurant

2). Biscarre , surpris par des **poursuites** que son imprudence lui avait **attirées** , a disparu depuis plus de trois semaines sans fairr e connaître sa résidence actuelle ;

3). À Rome , au temps de Néron , certain tribun des soldats , fils d' un honnête publicain , montrait dans l' administration militaire des **talents** qu' il avait précédemment **exercés** dans l' administration civile .

4). vous étiez prêt à renoncer à Pénélope qui vous attend , à Ulysse que vous verrez , à Ithaque où vous devez régner , à la gloire et à la haute **destinée** que les dieux vous ont **promise** par tant de merveilles qu' ils ont faites en votre faveur : vous renonciez à tous ces biens pour vivre déshonoré auprès d' Eucharis !

5). Un de nos blessés mourut ; mais je ne crois pas que sa mort fut la suite de la blessure qui l' avait alité , ce fut la puissance narcotique de la **drogue** que les natifs avaient **mise** dans le café .

1-heuristic subset

1). Tout cela , ce sont les restes de la **montagne** , que les eaux ont **réduite** en menus fragments , transportée en détail et déversée en énormes alluvions à l' issue des grandes vallées .

2). Étudiant à Paris , c' est là qu' il avait traversé les dernières années de la Restauration et les premières qui suivirent la révolution de 1830 , belles **années** que le siècle n' a pas **revues** depuis , qu' il ne reverra pas .

3). non , je ne me plains pas de vous , car je vous dois les quelques **jours** de bonheur que j' ai **passés** auprès de Blanche .

4). À cette lettre noble et touchante , qu' appuyaient auprès du roi les éminents **services** que , depuis son avènement , lui avait **rendus** le prince de Condé , il ne pouvait ne pas répondre par une adhésion sans réticences .

5). vous ne laisserez donc pas tomber ce masque d' hypocrisie dont vous avez couvert des **forfaits** qu' aucune langue humaine n' a **décrits** .

0-heuristic subset

1). Si elle vous coûte trop à dire , renvoyez-moi seulement mes billets et la **boucle** de cheveux que vous avez **emportée** ; je vous comprendrai et ..... Ah !

2). Seule , la pénurie de matériel et le manque d' information ont pu , au début de la guerre , permettre les **opérations** sans examen radiologique préalable , que , plus tard, on eût **considérées** comme criminelles .

3). voici l' instant de t' appliquer quelques-uns de ces **coups** de fouet qu' on t' a **ordonnés** pour le désenchantement de Dulcinée !

4). Un vif rayon de soleil perçant les nuages vint resplendir sur deux grands **portraits** placés de chaque côté de la cheminée , que le juif n' avait pas encore **remarqués** , et qui , peints en pied et de grandeur naturelle , représentaient , l' un une femme , l' autre un homme .

5). Quand je pense à la date de ces traités de 1814 , aux **difficultés** de tout genre que j' ai **éprouvées** ...

A.4 Additional figures and tables

Models	S-V	O-PP
LSTM	$94.3_{\pm 0.3}$	$82.1_{\pm 1.1}$
$\mathcal{M}$	$98.9_{\pm 0.04}$	$94.6_{\pm 0.2}$
$\mathcal{M}_{shallow}$	$90.8_{\pm 0.4}$	$84.7_{\pm 0.7}$
$\mathcal{M}_{shared}$	$97.8_{\pm 0.3}$	$89.0_{\pm 0.3}$
Majority class	69.7	65.1
(1) first noun	83.7	69.5
(2) Most recent noun	77.5	88.6
(3) Most recent token	66.9	60.3
(4) Majority number in prefix	75.9	70.0
(5) Noun before "que"	91.6	95.7

Table A.4: Accuracy (%) achieved by our models (averaged across 5 models for each architecture), compared to accuracies predicted by the 5 surface heuristics considered in this work on long-distance agreement tasks.

Constructions		Original		Nonce	
		LSTM	Transformer	LSTM	Transformer
S-V agreement					
	overall	$94.3_{\pm 0.3}$	$98.9_{\pm 0.04}$	$87.0_{\pm 0.4}$	$95.5_{\pm 0.2}$
	5 heuristics	$98.6_{\pm 0.1}$	$99.6_{\pm 0.05}$	$94.9_{\pm 0.6}$	$98.2_{\pm 0.1}$
	4 heuristics	$95.2_{\pm 0.5}$	$99.0_{\pm 0.1}$	$87.9_{\pm 0.8}$	$95.5_{\pm 0.3}$
	3 heuristics	$91.3_{\pm 0.8}$	$98.4_{\pm 0.1}$	$80.8_{\pm 1.2}$	$93.4_{\pm 0.3}$
	2 heuristics	$84.8_{\pm 1.0}$	$97.7_{\pm 0.1}$	$70.4_{\pm 2.2}$	$90.3_{\pm 0.5}$
	1 heuristic	$81.2_{\pm 1.8}$	$96.8_{\pm 0.1}$	$63.6_{\pm 1.8}$	$88.5_{\pm 0.7}$
	0 heuristic	$74.7_{\pm 2.2}$	$94.1_{\pm 0.5}$	$58.0_{\pm 3.0}$	$82.5_{\pm 1.6}$
O-PP agreement					
	overall	$82.1_{\pm 1.1}$	$94.6_{\pm 0.2}$	$77.1_{\pm 2.3}$	$93.9_{\pm 0.2}$
	5 heuristics	$95.3_{\pm 0.6}$	$99.2_{\pm 0.1}$	$91.1_{\pm 1.5}$	$98.7_{\pm 0.1}$
	4 heuristics	$85.9_{\pm 1.0}$	$96.5_{\pm 0.1}$	$80.5_{\pm 1.7}$	$95.7_{\pm 0.3}$
	3 heuristics	$71.9_{\pm 1.6}$	$91.6_{\pm 0.4}$	$65.9_{\pm 3.1}$	$91.3_{\pm 0.4}$
	2 heuristics	$62.2_{\pm 2.4}$	$87.6_{\pm 0.4}$	$56.6_{\pm 4.6}$	$87.3_{\pm 0.7}$
	1 heuristic	$37.4_{\pm 4.1}$	$77.9_{\pm 0.8}$	$30.3_{\pm 5.5}$	$73.1_{\pm 1.0}$
	0 heuristic	$40.2_{\pm 2.7}$	$76.1_{\pm 1.0}$	$35.8_{\pm 5.0}$	$69.2_{\pm 0.5}$

Table A.5: Average accuracy (%) of LSTM and Transformer models on the *Nonce set* versus the *Original set* by prediction difficulty.

Corpus	Target verb eval	Top3 eval
S-V agreement		
overall	$99.5_{\pm 0.1}$	$99.2_{\pm 0.1}$
5 heuristics	$99.8_{\pm 0.05}$	$99.6_{\pm 0.1}$
4 heuristics	$99.3_{\pm 0.1}$	$98.5_{\pm 0.2}$
3 heuristics	$99.3_{\pm 0.2}$	$98.7_{\pm 0.3}$
2 heuristics	$99.2_{\pm 0.2}$	$99.1_{\pm 0.2}$
1 heuristic	$98.7_{\pm 0.5}$	$98.4_{\pm 0.3}$
0 heuristic	$97.6_{\pm 0.6}$	$97.2_{\pm 0.5}$
O-PP agreement		
overall	$94.4_{\pm 0.6}$	$91.7_{\pm 1.0}$
5 heuristics	$99.2_{\pm 0.1}$	$99.0_{\pm 0.2}$
4 heuristics	$96.6_{\pm 0.5}$	$95.3_{\pm 0.6}$
3 heuristics	$91.3_{\pm 0.6}$	$87.0_{\pm 0.9}$
2 heuristics	$86.9_{\pm 1.1}$	$80.2_{\pm 1.8}$
1 heuristic	$76.6_{\pm 0.8}$	$62.2_{\pm 1.5}$
0 heuristic	$74.8_{\pm 0.4}$	$77.1_{\pm 1.2}$

Table A.6: Comparison of Transformer LM's accuracy in two agreement tasks using *top3* evaluation metric and target verb evaluation metric (§4.2.3). For a fair comparison, sentences were excluded where the top ten predicted words do not include any verbs, which account for 7.9% of sentences in S-V agreement and 0.3% in O-PP agreement.

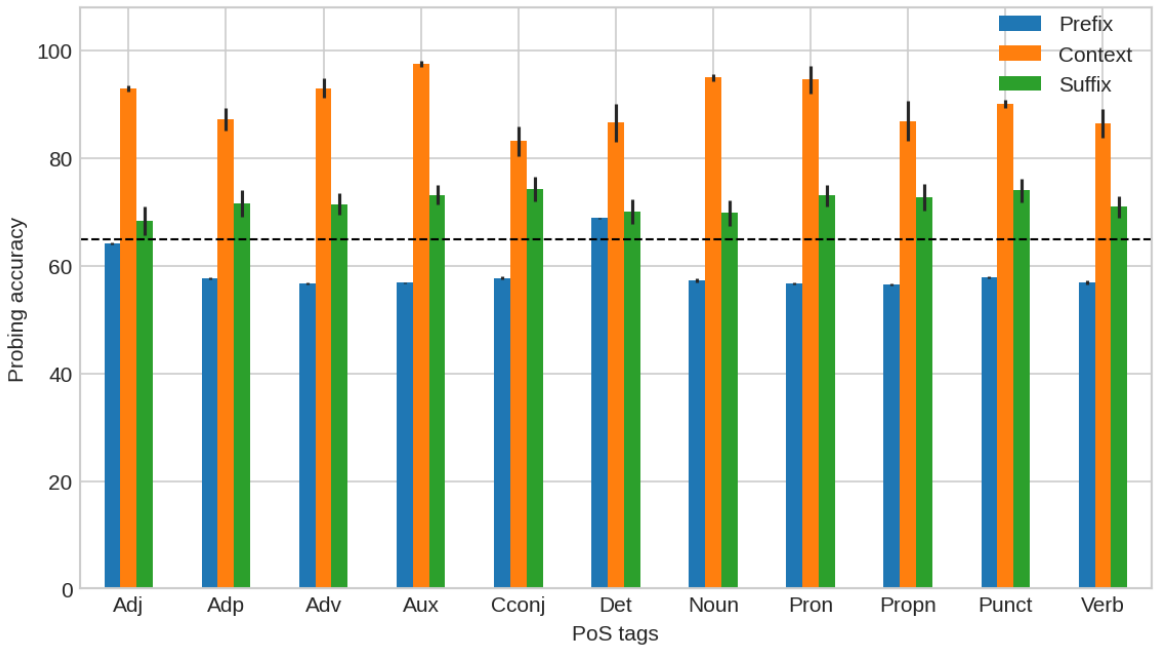


Figure A.1: Probing accuracy based on tokens PoS tags and their positions in the sentences, from left to right: *prefix*, *context*, *suffix*

Corpus	Target verb eval	Top3 eval
S-V agreement		
overall	$95.3_{\pm 0.4}$	$90.2_{\pm 0.6}$
5 heuristics	$98.6_{\pm 0.2}$	$94.5_{\pm 0.1}$
4 heuristics	$95.6_{\pm 0.3}$	$86.4_{\pm 1.2}$
3 heuristics	$92.6_{\pm 1.1}$	$87.0_{\pm 0.8}$
2 heuristics	$87.8_{\pm 1.2}$	$82.7_{\pm 0.9}$
1 heuristic	$86.3_{\pm 0.9}$	$81.7_{\pm 1.0}$
0 heuristic	$79.3_{\pm 1.6}$	$76.2_{\pm 2.1}$
O-PP agreement		
overall	$81.9_{\pm 1.8}$	$67.3_{\pm 2.3}$
5 heuristics	$96.1_{\pm 0.5}$	$83.7_{\pm 0.2}$
4 heuristics	$87.5_{\pm 1.1}$	$73.5_{\pm 0.6}$
3 heuristics	$71.1_{\pm 1.6}$	$51.3_{\pm 0.9}$
2 heuristics	$59.0_{\pm 2.4}$	$34.2_{\pm 1.8}$
1 heuristic	$31.7_{\pm 3.7}$	$22.2_{\pm 1.5}$
0 heuristic	$37.7_{\pm 2.5}$	$34.3_{\pm 1.2}$

Table A.7: Comparison of LSTM LM's accuracy in two agreement tasks using *top3* evaluation metric and target verb evaluation metric (§4.2.3). For a fair comparison, sentences were excluded where the top ten predicted words do not include any verbs, which account for 29.8% of sentences in S-V agreement and 45.7% in O-PP agreement.

	Transformer	LSTM
S-V agreement	$99.5_{\pm 0.1}$	$88.5_{\pm 0.3}$
O-PP agreement	$92.7_{\pm 0.2}$	$70.4_{\pm 0.5}$

Table A.8: Inter-agreement (%) between the target verb evaluation metric and the *top3* evaluation metric.

Subject-verb across object relative				
Subsets	Mask *context* tokens except cue que	Mask cue	Mask que	Mask cue⊕que
Overall	$16.7_{\pm 0.7}$	$10.6_{\pm 1.3}$	$2.6_{\pm 0.3}$	$13.5_{\pm 0.6}$
5 heuristics	$5.0_{\pm 0.5}$	$2.4_{\pm 0.4}$	$0.5_{\pm 0.1}$	$4.0_{\pm 0.3}$
4 heuristics	$15.3_{\pm 1.0}$	$10.1_{\pm 1.6}$	$3.0_{\pm 0.4}$	$12.5_{\pm 0.6}$
3 heuristics	$28.1_{\pm 0.8}$	$19.8_{\pm 3.0}$	$5.7_{\pm 0.4}$	$24.3_{\pm 1.1}$
2 heuristics	$44.1_{\pm 1.4}$	$21.1_{\pm 3.2}$	$7.0_{\pm 0.8}$	$35.4_{\pm 1.3}$
1 heuristics	$44.7_{\pm 1.9}$	$25.3_{\pm 2.3}$	$3.2_{\pm 0.6}$	$30.8_{\pm 1.9}$
0 heuristics	$42.6_{\pm 2.0}$	$31.1_{\pm 1.9}$	$6.7_{\pm 1.0}$	$34.3_{\pm 1.9}$
Object-past participle				
Overall	$8.4_{\pm 1.0}$	$25.6_{\pm 0.8}$	$17.9_{\pm 0.5}$	$30.1_{\pm 0.3}$
5 heuristics	$3.6_{\pm 0.1}$	$7.8_{\pm 0.3}$	$6.7_{\pm 0.2}$	$10.5_{\pm 0.5}$
4 heuristics	$7.9_{\pm 1.1}$	$24.0_{\pm 0.7}$	$15.3_{\pm 0.4}$	$27.6_{\pm 0.4}$
3 heuristics	$11.6_{\pm 1.7}$	$24.8_{\pm 1.4}$	$19.4_{\pm 1.0}$	$28.5_{\pm 0.4}$
2 heuristics	$12.6_{\pm 3.3}$	$40.5_{\pm 1.6}$	$39.8_{\pm 1.4}$	$53.8_{\pm 0.5}$
1 heuristic	$15.8_{\pm 3.3}$	$67.5_{\pm 1.4}$	$57.7_{\pm 1.0}$	$80.7_{\pm 0.4}$
0 heuristic	$24.0_{\pm 3.4}$	$59.0_{\pm 3.1}$	$64.0_{\pm 1.1}$	$88.4_{\pm 1.3}$

Table A.9: Average causal effect of interventions on Transformer's NA task performance, quantified by **drop** in accuracy before and after different interventions, and further broken down based on prediction difficulty measured by the number of heuristics. The term *cue* here refers to the antecedent and its modifiers (determiners and adjectives) in O-PP agreement, and to the subject and its modifiers in S-V agreement.

Constructions	$\mathcal{M}$	$\mathcal{M}_{NoPos}$
Perplexity	27.0	27.4
S-V agreement		
overall	$98.9\%_{\pm 0.04}$	$98.8\%_{\pm 0.1}$
5 heuristics	$99.6\%_{\pm 0.05}$	$99.6\%_{\pm 0.1}$
4 heuristics	$99.0\%_{\pm 0.1}$	$98.5\%_{\pm 0.2}$
3 heuristics	$98.4\%_{\pm 0.1}$	$98.1\%_{\pm 0.2}$
2 heuristics	$97.7\%_{\pm 0.1}$	$97.6\%_{\pm 0.1}$
1 heuristic	$96.8\%_{\pm 0.1}$	$97.2\%_{\pm 0.2}$
0 heuristic	$94.1\%_{\pm 0.5}$	$94.8\%_{\pm 0.7}$
O-PP agreement		
overall	$94.6\%_{\pm 0.2}$	$94.3\%_{\pm 0.3}$
5 heuristics	$99.2\%_{\pm 0.1}$	$99.0\%_{\pm 0.1}$
4 heuristics	$96.5\%_{\pm 0.1}$	$95.9\%_{\pm 0.2}$
3 heuristics	$91.6\%_{\pm 0.4}$	$91.3\%_{\pm 0.6}$
2 heuristics	$87.6\%_{\pm 0.4}$	$87.5\%_{\pm 0.6}$
1 heuristic	$77.9\%_{\pm 0.8}$	$78.1\%_{\pm 1.0}$
0 heuristic	$76.1\%_{\pm 1.0}$	$75.6\%_{\pm 1.3}$

Table A.10: Autoregressive Transformer LM's accuracy on two NA tasks with and without positional embeddings.

Constructions	$\mathcal{M}^{MLM}$	$\mathcal{M}^{MLM}_{NoPos}$
S-V agreement		
overall	$99.3_{\pm 0.2}$	$84.9_{\pm 0.8}$
5 heuristics	$99.7_{\pm 0.1}$	$96.7_{\pm 0.1}$
4 heuristics	$99.3_{\pm 0.1}$	$87.3_{\pm 0.3}$
3 heuristics	$99.0_{\pm 0.2}$	$77.1_{\pm 0.5}$
2 heuristics	$98.6_{\pm 0.1}$	$60.1_{\pm 1.1}$
1 heuristic	$98.1_{\pm 0.3}$	$46.5_{\pm 2.1}$
0 heuristic	$95.5_{\pm 0.3}$	$29.4_{\pm 1.8}$
O-PP agreement		
overall	$95.1_{\pm 0.2}$	$72.5_{\pm 2.3}$
5 heuristics	$99.4_{\pm 0.05}$	$92.2_{\pm 0.1}$
4 heuristics	$96.7_{\pm 0.1}$	$79.3_{\pm 0.7}$
3 heuristics	$92.2_{\pm 0.3}$	$55.6_{\pm 1.6}$
2 heuristics	$88.4_{\pm 0.5}$	$35.4_{\pm 1.8}$
1 heuristic	$82.2_{\pm 0.7}$	$30.3_{\pm 4.1}$
0 heuristic	$75.1_{\pm 1.1}$	$22.1_{\pm 2.5}$

Table A.11: Masked Transformer LM's accuracy on two NA tasks with and without positional embeddings.

A.5 Grammar and sampling details

SLOG expands upon the COGS vocabulary, which consists of 503 nouns and 113 verbs, to additionally include *wh*-words (*who, what*) and *that* used as a relative pronoun. In SLOG, for the sake of simplicity, we only consider restrictive relative clauses introduced by *that* regardless of the animacy of the head NPs. For indirect object-extracted instances, we use the preposition stranding structure, such as *the boy that Emma give a cake to*, rather than *the boy to whom Emma gave a cake*.

The dataset includes the 30,000 examples from the initial COGS training set, and new examples that fall into one of the following categories:

- Relative clauses within object NPs, equal in number to instances with PP modifications
- Subject and object *wh*-questions matching the quantity of their corresponding declarative sentences
- An equal number of four-level-nesting recursion constructions as the depth-2 instances in initial COGS
- A primitive example for each ditransitive verbs and verbs accepting complement clause (CP) arguments

Finally, the SLOG sampling process excludes sentences with duplicate nouns (e.g. *Emma saw Emma.*).

A.6 SLOG Full results and additional analyses

We report the full results of the experiments discussed in Section 5.5 in Table A.12.

A.6.1 Effect of the reformatted exact-match metric

All models exhibit higher overall accuracy with the reformatted exact-match evaluation compared to the initial metric, notably pretrained models with an over 10 percentage point increase (Table A.12). This suggests that the initial exact-match metric may have underestimated model performance.

A.6.2 RC Modifiers in unseen positions

Generalizing RC modifiers to unseen positions presents a similar challenge as PP modification cases, due to unobserved long-distance dependencies. As shown in Table A.13, all models

Generalization cases	Vanilla Transformer		T5		LLaMa		AM-Parser
Deeper PP recursion	13.1$_{\pm 1.5}$	13.1$_{\pm 1.5}$	15.7$_{\pm 0.7}$	16.6$_{\pm 1.0}$	19.8$_{\pm 1.1}$	20.6$_{\pm 1.0}$	100.0$_{\pm 0.0}$
Deeper tail CP recursion	0.2$_{\pm 0.1}$	0.9$_{\pm 0.3}$	0.8$_{\pm 0.2}$	5.3$_{\pm 0.4}$	3.9$_{\pm 0.4}$	12.1$_{\pm 0.7}$	100.0$_{\pm 0.0}$
Deeper center embedding	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	99.5$_{\pm 0.4}$
Shallower PP recursion	98.7$_{\pm 0.8}$	98.7$_{\pm 0.8}$	90.2$_{\pm 2.2}$	93.1$_{\pm 1.9}$	97.3$_{\pm 0.9}$	98.9$_{\pm 0.6}$	100.0$_{\pm 0.0}$
Shallower tail CP recursion	32.6$_{\pm 3.6}$	55.2$_{\pm 4.2}$	44.8$_{\pm 2.8}$	60.9$_{\pm 2.1}$	85.4$_{\pm 3.6}$	98.1$_{\pm 0.7}$	100.0$_{\pm 0.0}$
Shallower center embedding	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	64.1$_{\pm 19.1}$	0.0$_{\pm 0.0}$	50.7$_{\pm 5.7}$	100.0$_{\pm 0.0}$
PP in subject NPs	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.8$_{\pm 0.5}$	12.3$_{\pm 4.4}$	28.9$_{\pm 3.5}$	57.6$_{\pm 8.1}$
PP in indirect object NPs	42.5$_{\pm 2.2}$	42.5$_{\pm 2.2}$	50.1$_{\pm 1.7}$	53.8$_{\pm 1.4}$	55.0$_{\pm 3.9}$	71.2$_{\pm 4.2}$	90.4$_{\pm 8.1}$
RC in subject NPs	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.2$_{\pm 0.2}$	3.4$_{\pm 1.6}$	29.5$_{\pm 3.4}$	55.8$_{\pm 8.4}$
RC in indirect object NPs	34.4$_{\pm 6.0}$	34.8$_{\pm 6.1}$	35.1$_{\pm 1.9}$	36.6$_{\pm 2.1}$	48.6$_{\pm 1.9}$	55.0$_{\pm 2.1}$	74.4$_{\pm 6.4}$
Indirect object-extracted RC	4.7$_{\pm 5.6}$	4.7$_{\pm 5.7}$	0.0$_{\pm 0.0}$	0.0$_{\pm 0.0}$	0.1$_{\pm 0.3}$	2.5$_{\pm 3.2}$	0.0$_{\pm 0.0}$
Indirect object *wh*-questions	35.9$_{\pm 8.3}$	42.4$_{\pm 13.5}$	0.0$_{\pm 0.0}$	0.4$_{\pm 0.7}$	27.9$_{\pm 9.3}$	73.5$_{\pm 18.4}$	41.4$_{\pm 42.4}$
Active subject *wh*-questions	96.7$_{\pm 2.6}$	97.1$_{\pm 2.4}$	90.5$_{\pm 4.0}$	98.1$_{\pm 1.7}$	92.8$_{\pm 6.4}$	93.3$_{\pm 6.0}$	99.8$_{\pm 0.6}$
Passive subject *wh*-questions	27.4$_{\pm 1.7}$	31.9$_{\pm 5.4}$	20.3$_{\pm 3.8}$	100.0$_{\pm 0.0}$	4.8$_{\pm 4.5}$	15.3$_{\pm 17.5}$	100.0$_{\pm 0.1}$
Direct object *wh*-questions	2.8$_{\pm 3.4}$	16.0$_{\pm 12}$	47.2$_{\pm 1.0}$	98.5$_{\pm 0.9}$	0.5$_{\pm 0.5}$	8.6$_{\pm 5.7}$	29.4$_{\pm 33.5}$
Wh-questions with modified NPs	17.6$_{\pm 0.9}$	17.8$_{\pm 1.3}$	20.5$_{\pm 1.0}$	36.8$_{\pm 0.4}$	15.8$_{\pm 0.6}$	20.8$_{\pm 2.4}$	55.6$_{\pm 12.5}$
Wh-questions long movement	4.0$_{\pm 7.8}$	4.9$_{\pm 9.5}$	23.3$_{\pm 4.3}$	24.9$_{\pm 5.1}$	0.8$_{\pm 1.4}$	3.0$_{\pm 4.7}$	0.0$_{\pm 0.0}$
Overall	**24.2**$_{\pm 1.0}$	**27.1**$_{\pm 2.0}$	**23.4**$_{\pm 1.1}$	**40.6**$_{\pm 1.0}$	**27.6**$_{\pm 1.0}$	**40.1**$_{\pm 1.8}$	**70.8**$_{\pm 4.3}$

Table A.12: Mean accuracy (%) using exact-match is shown in gray, accuracy using reformatted exact-match described in Section 5.4 is shown in black. AM-Parser's graph-based output yields identical scores for both metrics hence only a single column is reported.

Generalization cases	Long pred-arg dependency?	Vanilla Transformer	T5	LLaMa	AM parser
Sub-case: Passive indirect objects **A fish was given** to [a cat that slept]$_{\text{iobj}}$.	✗	72.0$_{\pm 6.6}$	74.2$_{\pm 2.7}$	97.1$_{\pm 1.2}$	99.5$_{\pm 0.6}$
Sub-case: Indirect object in PP datives Emma **gave a fish** to [a cat that slept]$_{\text{iobj}}$.	✗	27.0$_{\pm 9.8}$	38.9$_{\pm 5.3}$	72.7$_{\pm 7.8}$	99.3$_{\pm 1.1}$
Sub-case: Indirect object in double object datives Emma **gave** [a cat that slept]$_{\text{iobj}}$ **a fish**.	✓	7.9$_{\pm 8.5}$	0.2$_{\pm 0.2}$	0.3$_{\pm 0.3}$	28.9$_{\pm 17.2}$
Subject [**A cat** that slept]$_{\text{subj}}$ **ate** a fish.	✓	0.0$_{\pm 0}$	0.2$_{\pm 0.2}$	29.4$_{\pm 3.4}$	51.7$_{\pm 8.4}$

Table A.13: Performance of RC modification generalization broken down by construction.

exhibit a significant performance discrepancy between constructions involving unseen long predicate-argument dependencies and those that do not.

For novel positions that introduce long predicate-argument dependencies, RC modification in the indirect object appears more difficult than in the subject position, contrary to the case with PP modifiers. The primary error pattern (18) demonstrates that models struggle to detect the RC boundary when the relative clause ends with a verb. They systematically misinterpret the indirect object `a fish` of the main verb `gave` as the direct object of the adjacent embedded verb `slept`.

A.6.3 Passive subject *wh*-questions

For subject *wh*-questions, which exhibit no gap, T5 and AM-Parser perform near-perfectly on both active and passive subject questions. Vanilla Transformer and LLaMa also perform well on active subject questions, but achieve much lower performance on passive subject questions. This performance discrepancy is the most evident in sub-cases where passive subjects function as theme (e.g., (17))—the vanilla Transformer has near-zero accuracy for these sub-cases, systematically failing to map *wh*-words to '?' as in (17b):

(17) Input: What was eaten by Emma ?

a. Gold: `eat.theme (x2, ?) ∧ eat.agent (x2, Emma)`

b. Output of Vanilla Transformer and LLaMa: `eat.theme (x2, x4) ∧ eat.agent (x2, Emma)`

As discussed in Section 5.5.3, this error pattern may result from the highly imbalanced label distribution in training output space. Both LLaMa and vanilla Transformer are inclined to repeat the substantially more common subsequence `theme(`x_i, x_j`)` over `theme(`x_i`,?)`.

(18) Gold LF and model-predicted LF for *Emma gave a cat that slept a fish*:

a. Gold: `give.agent (`x_1`,Emma) ∧ give.`**`recipient`**` (`x_1, x_3`) ∧ give.theme (`x_1, x_7`)∧ cat(`x_3`) ∧ cat.nmod (x3,`x_5`) ∧ sleep.agent(`x_5, x_3`) ∧ fish(`x_7`)`

b. Out: `give.agent (`x_1`,Emma) ∧ give.theme (`x_1, x_3`) ∧ cat(`x_3`) ∧ cat.nmod (x3,`x_5`) sleep.agent(`x_5, x_3`) ∧ sleep.theme(`x_5, x_7`) ∧fish(`x_7`)`

A.6.4 *Wh*-questions with modified NPs

In *wh*-questions with PP and RC modifiers, even though the SLOG training set only contains *wh*-questions with unmodified NPs, all models generalize well (accuracy > 80%) to direct object NPs with modifiers (e.g., *Who ate a cake on the table?*). These are cases where the modification pattern is observed in training as a part of declarative sentences. In contrast, performance declines when models encounter *wh*-questions with modifiers in the indirect object position (i.e., modification structure not observed as part of declarative sentences). Similarly, for *wh*-questions with subject position modifiers, the performance is very low: both T5 and vanilla Transformers achieve near-zero accuracy, and LLaMa achieves around 5%.

This observation mirrors the patterns discussed in §5.5.2, attributed to difficulties introduced by unseen subject-verb dependencies across PPs or RCs. In contrast, the structure-

aware model exhibits significantly better performance in *wh*-question with subject modification.

A.7 SLOG: Results of variable-free LFs

Generalization cases	Vanilla Transformer	T5	LLaMa
Deeper PP recursion	$7.8_{\pm 1.8}$	$63.0_{\pm 2.9}$	$90.9_{\pm 3.3}$
Deeper tail CP recursion	$1.0_{\pm 0.5}$	$46.2_{\pm 2.6}$	$44.1_{\pm 7.9}$
Deeper center-embedding	$0.0_{\pm 0.0}$	$7.8_{\pm 1.1}$	$9.4_{\pm 2}$
Shallower PP recursion	$98.2_{\pm 1.6}$	$99.6_{\pm 0.9}$	$100.0_{\pm 0.0}$
Shallower tail CP recursion	$89.3_{\pm 3.3}$	$99.3_{\pm 1.6}$	$100.0_{\pm 0.0}$
Shallower center-embedding	$0.1_{\pm 0.2}$	$99.8_{\pm 0.3}$	$99.8_{\pm 0.4}$
PP in subject NPs	$0.2_{\pm 0.3}$	$73.2_{\pm 9.0}$	$93.4_{\pm 4.8}$
PP in indirect object NPs	$29.3_{\pm 10.7}$	$97.4_{\pm 2.1}$	$98.1_{\pm 1.9}$
RC in subject NPs	$0.1_{\pm 0.1}$	$60.8_{\pm 6.3}$	$73.9_{\pm 13.5}$
RC in indirect object NPs	$4.0_{\pm 1.9}$	$71.9_{\pm 0.8}$	$73.6_{\pm 3.9}$
Indirect object-extracted RC	$0.0_{\pm 0.0}$	$62.4_{\pm 7.5}$	$3.3_{\pm 2.8}$
Indirect object *wh*-questions	$34.1_{\pm 31.1}$	$93.4_{\pm 4.8}$	$83.8_{\pm 11.3}$
Active subject *wh*-questions	$99.0_{\pm 0.5}$	$99.8_{\pm 0.3}$	$96.2_{\pm 2.6}$
Passive subject *wh*-questions	$57.3_{\pm 23.8}$	$99.9_{\pm 0.1}$	$96.0_{\pm 3.0}$
Direct object *wh*-questions	$41.8_{\pm 3.8}$	$48.4_{\pm 0.4}$	$44.1_{\pm 4.6}$
Wh-questions with modified NPs	$18.1_{\pm 2.3}$	$68.0_{\pm 1.9}$	$69.4_{\pm 6.8}$
Wh-questions long movement	$7.4_{\pm 3.7}$	$45.6_{\pm 4.6}$	$35.7_{\pm 6.5}$
Total	$28.7_{\pm 4.1}$	$72.7_{\pm 1.1}$	$71.3_{\pm 3}$

Table A.14: Mean accuracy (%) on SLOG using the variable-free logical form of Qiu et al. (2022a).

Table A.14 reports the accuracy on SLOG using variable-free logical forms. The AM-Parser is unable to handle the variable-free format and therefore is omitted. The hyperparameters for the three tested models are the same as the experiments described in Section 5.4.

The variable-free LF, as discussed in Section 5.3 and Wu et al. (2023), exhibits certain limitations and ambiguities which render direct comparisons with variable-based LF results inappropriate. Regardless, all three models achieve higher accuracy scores on the variable-free LFs compared to the COGS LFs, with pretrained models experiencing a particularly significant boost. This aligns with the observations of Qiu et al. 2022b.

Despite the change in LF, the overall trends and challenges remain consistent. The vanilla Transformer struggles with the same generalization cases, failing to extrapolate

to deeper recursion depths and struggling with cases involving unseen long-distance dependencies. Pretrained models, while exhibiting better overall performance, continue to struggle with more structurally complex generalization cases in their respective categories. These include deeper center-embedding, indirect object-extracted RC and *wh*-questions with long movement.

A.8 Discussion and limitations

While SLOG offers a targeted and well-controlled approach to assess structural generalization, it presents some limitations.

First, SLOG is a synthetic corpus and covers only a fraction of the diverse structures in English. Furthermore, previous research has demonstrated that the design of meaning representation (MR) can have a nontrivial effect on model performance in semantic parsing tasks (Guo et al., 2019; Herzig et al., 2021; Qiu et al., 2022b). For example, as noted by Wu et al. (2023), the variable indexing scheme may introduce additional semantically irrelevant challenges when assessing structural generalization. SLOG's reformatted exact-match evaluation metric partially addresses this concern by taking into consideration several variations of MRs that are semantically equivalent including MRs that are equivalent up to constant renaming. However, a more comprehensive study of the effect of artifacts from the formalism is left to future work.

Second, there also exist challenges specific to the evaluation of pretrained models. That is, distributional shift between training and generalization sets intended by SLOG, such as withholding the constructions *PPs modifying subject NPs* from training, is difficult to strictly enforce when pretraining is involved Kim et al. (2022). This potential violation of distributional control makes the interpretation of the obtained results difficult; we cannot disentangle whether generalization success in pretrained models derives from genuine compositional capabilities or simply exposure during pretraining to the target constructions meant to be withheld from the evaluated models. Still, corpus analyses such as Karlsson (2007) suggest that deep center-embedding beyond three levels is very rare in naturally occurring data, so it is possible that very deep embedded structures are withheld as intended even from models exposed to large amounts of pretraining data. We hope the additional structural generalization cases that SLOG offers can also help with future work investigating the interaction between structures available in pretraining data and structural generalization.