Jackson Petty, 2022
Department of Linguistics, Yale University

CHARACTERIZING ALGEBRAIC GENERALIZATION IN LINGUISTIC NEURAL NETWORKS

"Thou shalt not make a machine in the likeness of a human mind."

— Frank Herbert, *Dune*

This document was typeset with LuaLaTeX on April 26, 2022. Serif text is set in Cochineal. Sans-serif text is set in Alegreya Sans. Monospace text is set in Andale Mono.

Characterizing Algebraic Generalization in Linguistic Neural Networks

Characterizing Algebraic Generalization in Linguistic Neural Networks

Jackson Petty
Advised by Robert Frank

*Submitted to the faculty of the Department of Linguistics
in partial fulfillment of the requirements for the degree of
Bachelor of Arts*

Yale University
April 26, 2022

Abstract

To learn an unbounded problem is to generalize well from a limited set of training data. In humans, robust language acquisition requires language learners to form strong generalizations on the basis of very limited evidence (Chomsky 1980). These generalizations seem to require the acquisition of functional abstractions of some sort. Various analysis of these abstractions have been put forth in the context of replicating this generalization in artificial settings (Fodor & Pylyshyn 1988, G. F. Marcus 1998a, Lake & Baroni 2018). But what connects these descriptions of generalization, and how do they characterize the difficulty of a particular linguistic phenomena in a way which informs our expectations of whether it can be learned by an artificial neural network?

This thesis takes a broad approach to answering this question. First, we explore a formalism to unite the various descriptions of linguistic generalization in a way which establishes a complexity hierarchy for tasks requiring generalization. We then use this typology to explore whether a given task is learnable by simple recurrent networks lacking explicit inductive biases for such generalization. We present constructive evidence that such simple models are in fact capable of learning stronger generalizations than previously thought, raising important questions about the mechanisms by which generalizations can be learned by neural network models of language.

Contents

Chapter 1

Introduction

This thesis began as an attempt to replicate a negative result. Working with my advisor, Professor Robert Frank, I attempted to reproduce a result described in an earlier joint paper of his exploring the ability (or lack thereof) of recurrent language models to learn a task known as *anaphora resolution*: when presented with an input like *Alice sees herself,* models were tasked with providing the correct interpretation of the reflexive anaphor *herself* in the context of the larger sentence (in this example, *herself* → ALICE). The failure of small recurrent models to effectively capture this phenomena was supported empirically by earlier work (cf. Frank, Mathis & Badecker 2013) and theoretically by doubts about the capacity of such models to learn *algebraic generalizations*—generalizations which involve abstracting over the identity of particular inputs to learn a general pattern—in the absence of specific inductive biases to do so.

We were thus surprised when we found that simple recurrent models are able to succeed at this task with perfect generality when used in a sequence-to-sequence context. This unexpectedly successful result raised further questions: How are these models able to succeed where others failed? By what mechanism can simple models learn these tasks? To what degree can their performance be extended to more complicated domains which more closely resemble natural language?

Although the experimental domain for this initial experiment is small and synthetic, capturing only a small slice of the linguistic context provided by natural language to human learners, the ability of models to succeed at this task is important. As machine learning techniques become more advanced and the capabilities of neural models more pronounced, it is becoming increasingly clear that structurally-sensible generalization remains a stumbling block in the capacity of neural models to understand and interpret natural language (Lake 2019). The effects of this are twofold: on the one hand, achieving adequate (i.e., humanlike) performance on arbitrary input data requires the collation of vast quantities of data and the expenditure of enormous quantities of energy in order to successfully train models (Dodge et al. 2021). This situation is undesirable for a number of reasons. Current models require orders of magnitude more training data than human language learners to achieve comparable levels of performance, which poses theoretical concerns for the comparability of the two systems. More practically, acquiring that much data and expending that much energy ranges from devilishly tedious in the case of well-documented languages to downright impossible for under-documented ones, posing serious implementation challenges for research scientists and industry practitioners (Walther & Sagot 2017, Bender et al. 2021). Furthermore, enormous training sets are often difficult or impossible to thoroughly vet for bias and data poisoning, leaving models vulnerable to unpredictably undesirable outcomes (Bender et al. 2021). On the other hand, even in the cases where such quantities of data can be found and such energy expended so as to produce well-performing models, the actual general performance of such models is often frustratingly brittle in hard-to-interpret ways. Models are often hypersensitive to small perturbations in input data and fail to generalize in domain-sensible ways (see, for instance, Abdou et al. (2020)).

Efforts to remedy these problems often focus on studying the generalizable capacity of models: that is, how they learn to make educated guesses on data which falls outside their training domain. Models which learn to make sensible generalizations on the basis of more limited support are better equipped to be used in academia and industry. Defining what counts as a 'sensible' generalization and producing models which display these characteristics is, of course, easier said than done. In linguistic domains, much attention has been given to the ideas of *compositional* and *structural* generalization, wherein knowledge of component pieces combines to produce knowledge of a larger, unseen whole (Fodor & Pylyshyn 1988, Lake & Baroni 2018). Success at tasks involving structural composition remains elusive in the best-performing models of language and vision (Lake & Baroni 2018, Gordon et al. 2019).

This thesis charts a narrow path through this large open problem. We seek to relate the various descriptions of generalization in a way which admits an analysis of what problems in linguistic domains are learnable by what kinds of artificial neural network models. To do this, we explore several problems requiring a degree of algebraic generalization, demonstrating empirically that simple recurrent networks lacking inductive biases for algebraic generalization are more powerful than previously thought. We then attempt to analyze these models to better understand how these networks acquire knowledge of their training domains consistent with algebraic generalization to a larger context.

1.1 Organization

The remainder of this thesis explores the capacity of linguistic neural networks to exhibit algebraic generalization in various realms. Chapter 2 introduces algebraic generalization as a formal property of learning and summarizes the previous work that has been done to explore how neural models exhibit algebraic generalization in linguistic domains. We adopt the analysis of Gordon et al. (2019) to provide a formal characterization of generalization complexity as measured by the complexity of the structural and function components associated with a particular task, and use this formalism to connect notions of algebraic, lexical, structural, and compositional (linguistic) generalization under a single formalism. We then pick out specific natural-language phenomena exhibiting patterns of algebraic knowledge and explore if and how networks manage to acquire this knowledge in in a way which generalizes. This empirical exploration is divided into two parts: Part I focuses on exploring properties of algebraic generalization in sequence-to-sequence models. Chapter 3 provides experimental evidence that even simple recurrent networks without attention are able to display algebraic generalization in domain-specific tasks, contradiction earlier assumptions about the generative capacity of recurrent models. Chapter 4 extends this work to include more modern transformer models and explores the degree to which these various architectures exhibit preferences for linear generalization strategies which are at odds with the generalizations made by children to account for patterns in natural language. Chapter 5 takes the results of chapter 3 and attempts to analyze how the models surveyed therein are able to learn and exhibit algebraically-generalized knowledge.

Part II Moves from a sequence-to-sequence context to a language modelling context. In Chapter 6 we explore the degree to which large, general-purpose, pre-trained language models have acquired a generalization for alignment between surface position and thematic roles.

Chapter 7 concludes with an extended discussion of how the results of these studies inform the work being done in adjacent sectors of cognitive science, machine learning, and computational linguistics, and sketches the outlines of several promising avenues for future research to further explore the topics presented here.

1.2 Reproducability and Code

The validity of any scientific insight is fundamentally predicated on the ability of others to independently examine and reproduce its results. In many cases, well-meaning authors and researchers must mange the competing interests of economy and completeness when reporting their findings, a natural consequence of which is ambiguity in the interpretation of data, results, and experimental methodology. Yet as this work is not bound by any firm page limit, nor need it conform to particular expectations for what information about an experiment is privileged in explanation, I have made every effort to provide clear and thorough accounts for how I arrived at my conclusions and how future researchers may attempt to replicate these results should the interest arise.

Working in a computational domain provides solutions to many of these challenges, but also introduces new problems for readers to contend with. Link rot and dependency ambiguity are often obstacles which impede researchers in attempting to reproduce a particular paper's results. To address this challenge, I have included with every chapter a post-script describing how to reproduce the discussed experiments using the code and data which created the original results. By combining a dependency-pinned virtual environment (`conda`) with a virtualization and containterization framework (`docker`), it is possible to provide a platform-independent solution to the problem of training and evaluating machine learning results. Each experiment herein described is thus explicable and reproducible by any who wish to do so.

1.3 Acknowledgements and Previously Published Material

A massive debt of gratitude is owed to many people besides myself who facilitated the work that is presented here. First and foremost, I must acknowledge that authorial credit for many of the individual chapters is shared with Robert Frank and Michael Wilson who have patiently worked with me for the past two years on exploring various topics related to the intersection of machine learning and theoretical linguistics. This work began in earnest during the summer of 2020 when I began working with Professor Fank in the CLAY Lab. The result of that work was the publication of two papers which form the core of Chapters 3 and 4. This work continued through the fall of 2021, where we wrote with Michael Wilson the core of Chapter 6. In addition to being coauthors of these publications, Bob and Michael have also been invaluable instructors as I've learned to conduct research and present it in written and spoken form, and I certainly couldn't have accomplished any of the work that follows without their help and guidance.

A broader acknowledgement is needed too for the many members of the CLAY Lab who have worked with me over the past several years and who have continued to provide useful feedback and friendly camaraderie during my time at Yale. I owe particular thanks in this regard to Noah Amsel, Shayna Sragovitcz, and Arohi Srivasatva.

Chapter 2

Formalizing Generalizing

> *"All generalization is hard. But some generalizations are harder than others."*
>
> — George Orwell, *Animal Farm* (paraphrased)

To learn a pattern in data is to learn a function which associates particular input data to particular outputs in a sensible manner. But why are some patterns harder to learn than others? Enormous effort is expended by scientists and engineers to develop models which are "better" than the previous iteration, where "better" means that the model gives more accurate results in ever larger problem spaces. As one moves beyond toy models and trivial datasets, it quickly becomes clear that fully training a model on all its possible inputs is not feasible. On a practical level, many problem domains are infinite: a model which *knows* a language to arbitrary levels of competence cannot possibly train on all possible linguistic inputs or outputs because natural language is not a finite domain. On a theoretical level, models which exhaust the set of possible training data are fundamentally uninteresting, for the true value of models lies in their predictive capacity, which of course necessitates that a model is not simply memorizing seen mappings between its input and output domain.

What we desire, then, are models which *generalize*—make sensible educated guesses on data which is qualitatively or quantitatively divergent from the training set. Creating generalizable models for arbitrary problems is then a, if not *the*, centrally important task to machine learning and computational scientists who wish to model observed phenomena. In the past decade, computational scientists have witnessed an explosion in the generative capacity of artificial neural models, which are now employed to solve increasingly difficult tasks in an ever-widening field of problem domains. Yet despite this renaissance of machine learning, some tasks remain frustratingly difficult to solve.

This disparity is made all the more confusing when artificial models are compared to human learners, who are often able to achieve better, or at least comparable, levels of performance on a vast array of different problem tasks on the basis of far less training than is provided to state-of-the-art models. This gap in performance and training raises important questions for engineers and researchers who seek to improve the capacity of their models: How are natural learners able to acquire generalizations on the basis of such comparatively limited support? What makes some problems harder to solve than others in a sufficiently general manner? How can we design models which better emulate the patterns of learning we observe in human subjects?

2.1 What makes something difficult to learn?

Some forms of generalization amount to sensible treatment of unforeseen inputs. Imaging a color classification model which is trained to classify input hues into discrete categories. Given a robust but limited sample of possible inputs for each of the various output categories 'red,' 'blue,' 'green,' 'yellow,' and 'purple,' a model with good generalizational capacity will be able to accurately classify inputs whose shade or hue is reasonably perturbed from any of the training samples despite never having seen this novel input before. This kind of generalization is fundamentally analyzable as a tolerance on input data. On the basis of a limited set of input data, generalizable models can make sensible inferences about new inputs which are 'nearby' in the learned encoding representation of these inputs. We may think of the set of inputs in colorspace not shown to the model during training as the withholding set, and so say that the model generalizes well to this withholding set because it accurately classifies members of this set on the bases of its knowledge of the training set to arbitrary accuracy—that is to say that given an arbitrary level of desired accuracy, we can provide sufficient training support to teach a model to approximate the ideal classification to within this desired accuracy.

Although many forms of generalizational capacity can be expressed in this way, other forms of generalization seem to require more computational capacity than mere tolerance. Consider a linguistic model which is tasked with copying a given but arbitrary input. While certainly a trivial task when the input domain is finite, how can a model learn to generalize this task to arbitrary known inputs? These kinds of problems seem to require more computational machinery than simple generalizational tasks. Yet the characterizations of different kinds of generalization tasks remains both vague, relying heavily on intuition to provide a sense of problem difficulty; and domain-specific, providing more rigorous treatments of specific phenomena in particular domains, but failing to make connections *across* domains or connect these characterizations to formalisms of human cognition.

An early attempt to break through this problem came in G. F. Marcus (1998a)'s work in *algebraic generalization*. The key theoretical insight of G. F. Marcus (1998a) and subsequent work, including G. F. Marcus (2001) and Berent & Marcus (2019), is to observe that generalization requires one to learn an "operational abstraction" over a set of data. Berent & Marcus (2019) define this notion in somewhat vague terms, but ultimately argue that generalization requires a model to first, learn a partition of the input domain into equivalence classes of data which share abstract properties; and second, learn a set of functions which operates on the members of these equivalence classes in a uniform manner.

This notion of uniform treatment captures a quality which seems fundamental to human cognition, and which is certainly a central part of the human capacity for language. Speakers naturally learn that lexical items pattern together on the basis of their positional distribution, forming into cogent cognitive categories like noun, verb, article, and so on. But more than simply a descriptive phenomenon, these categorical abstractions are operational in the sense that knowing a word belongs to a particular syntactic category is sufficient to let a speaker use that word in any context where members of that category are valid. This inferential leap is what allows speakers of English to learn that *Xandar* is a name and then have no trouble making sense of a complicated expression like that found below in (1).

(1) "Which one of Xandar's books did he tell you to borrow?"

Despite never having encountered the name *Xandar* in this context, knowledge that *Xandar* is a name allows one to produce and interpret such a sentence without any additional training data.

Despite this intuitive foundation, Berent & Marcus do not provide concrete examples of how to formalize this notion of uniform treatment; offer any guidance on what restrictions, if any, are placed on the kinds of equivalence classes which are requisite for generalization to happen[1]; or make clear the connection between the equivalence classes themselves and the functions which act on them. The status of algebraic generalization, though promising, is thus in limbo awaiting a connection to the practical problems faced by computational scientists and researchers attempting to build models which exhibit such generalizational capacity.

One promising avenue towards formalizing the notion of algebraic generalization and connecting it to the kinds of practical problems facing modelers lies with a different treatment of generalization: *compositional generalization*. Compositional generalization views generalization capacity as the capability of a model to learn knowledge of a

[1] For instance, we may consider examples at two extremes: one one hand, a task may require that each input is 'unique' in that it forms an equivalence class with itself; conversely, a task may be described as requiring the every input to map to a single equivalence class. While such trivial examples are perhaps easily dismissed on a case-by-case basis, the larger question of what constitutes a valid equivalence class for a problem requiring algebraic generalization still requires an answer.

larger structure conditional on knowledge of smaller, modular components. Language provides a natural sandbox for studying the compositional generalizational capacity of models since it is fundamentally built on computational knowledge in which smaller structures combine in predictable ways to produce arbitrarily complex constructions Fodor & Pylyshyn (1988), Fitch, Hauser & Chomsky (2005). This property of natural language has prompted a recent flurry of research towards creating datasets which model compositional generalization and testing whether artificial neural models are able to learn compositionally general tasks (see, for instance, Lake & Baroni 2018, Lake 2019, Kim & Linzen 2020, *inter alia*).

In a linguistic setting, compositional generalization is sometimes broken down into *lexical generalization*, wherein a model learns to interpret, operate upon, and produce novel words subject to learned distributional restrictions; and *structural generalization*, where a model extends its knowledge of a structural object to larger and more complicated inputs (for instance, generalizing from short sequences to long sequences, or learning patterns of language which exhibit syntactic sensitivity and then testing in more syntactically complex environments).

Outside of the domain of natural language, compositionality and the challenges of achieving it surface in a wide variety of phenomena. Consider generative image models like DALL·E, which have been trained to generate images based on written prompts. Such models have proven remarkably adept at generating complex and detailed scenes. Yet these models still show signs of brittleness when faced with tasks which require composition. Consider as an example the collection of DALL·E mini images shown below in Figure 2.1. Here, the model displays excellent performance at generating images which match component prompts like 'a red ball,' 'a blue cube,' and 'a yellow pyramid,' but this knowledge does not translate into success at composing these components into a larger scene containing 'a yellow pyramid on top of a blue cube, next to a red ball.'

While compositional generalization likewise provides a useful framework for thinking about how models can learn to generalize given knowledge into new constructions, there remains work to be done in characterizing how compositional tasks are measured. For example, while it is intuitively obvious to speakers of natural language that tasks on the COGS dataset requiring structural generalization are harder than those involving mere lexical generalization (Weißenhorn et al. 2022), what formal reason can be given for why this is the case? What makes structural generalization a harder task than lexical generalization? Where does the boundary between structural and lexical tasks lie? More generally, what are the extra-domain analogues to lexical and structural generalization? How do these concepts translate into the domain of multimodal language-vision models like DALL·E? Finally, how does the structural knowledge outlined in theories of compositional knowledge relate to the kind of algebraic knowledge described by Berent & Marcus (2019), which focused on identity relations as the key component to generalized learning?

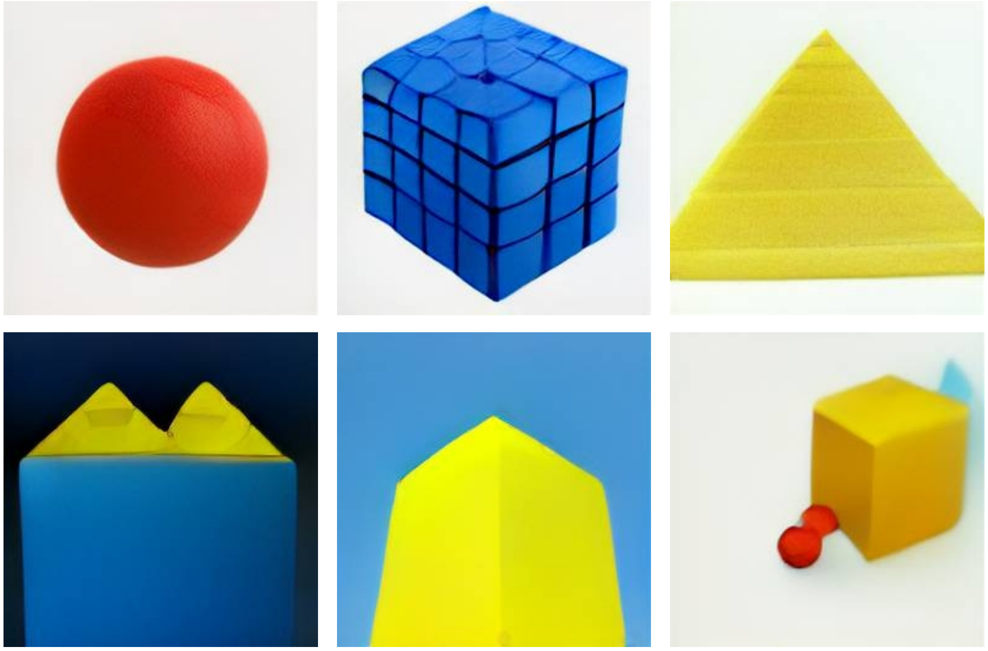

Figure 2.1: Top Row: DALL·E mini shows excellent performance at generate scenes matching individual component inputs ('a red ball,' 'a blue cube,' and 'a yellow pyramid'). **Bottom Row:** However, success at interpreting these inputs independently does not mean that it knows what to do with them when they are composed in the input ('a yellow pyramid on top of a blue cube, next to a red ball').

2.2 A Proposed Typology of Generalization

One large step taken towards reconciling these disparate understandings of generalization comes from Gordon et al. (2019), who begin to formalize notions of compositionality in relation to the SCAN task of Lake & Baroni (2018). Here, the authors attempt to characterize the kind of compositional knowledge which plagues current sequence-to-sequence models Lake & Baroni (2018), Gordon et al. (2019). Their main contribution is to show that some forms of compositionality can be described as a form of *group equivariance*. To recount their characterization, consider first the definition of an equivariant map.

(2) **Equivariant function:** Let $\mathcal{S}, \mathcal{T}$ be sets, and let G be a group with left-action on $\mathcal{S}$ and $\mathcal{T}$. A function $f : \mathcal{S} \longrightarrow \mathcal{T}$ is equivariant if it composes over the action of the group; that is, $f(g \cdot x) = g \cdot f(x)$ for all $x \in \mathcal{S}, g \in G$.

Any $g \in G$ acts as a permutation on the sets $\mathcal{S}, \mathcal{T}$. Equivariant functions are those which respect those permutations. We can further refine this definition by considering cases when the group action on a sequence is equivalent to an individual permutation applied to each term.

(3) a. **Local action:** A group G acts on a set $\mathcal{S} = S \times \cdots \times S$ locally if for all $g \in G$ there exits some action $g_S \cdot$ such that $g \cdot (s_1, \cdots, s_n) = (g_S \cdot s_1, \cdots g_S \cdot s_n)$.

b. **Locally equivariant function:** An equivariant function is locally equivariant if its associated action is local.

That is, a local action is one in which permutation factors into the individual tokens of the input domain. In the context of sequence-to-sequence tasks (Sutskever, Vinyals & Le 2014) where our input and output domains are sequences of text built from vocabularies, local actions are those were each permutation of sequences is decomposable into a fixed permutation on the indivual vocabulary words.

Gordon et al. (2019) propose that compositionality in linguistic tasks requires the learning of equivariant functions. To see why, consider a small subset of Lake & Baroni (2018)'s SCAN task, which requires a model to translate a series of input predicates into commands for a robot, as shown below in (4).

(4) a. *jump* ⟶ JUMP

b. *run* ⟶ RUN

c. *walk left* ⟶ L_WALK

d. *jump twice* ⟶ JUMP JUMP

Importantly, some predicates in the input domain, like *twice*, are special in that they tell the robot to repeat whatever predicate precedes them. To learn to interpret all possible SCAN commands, it is not enough to learn that *jump twice* ⟶ JUMP JUMP, *walk left twice* ⟶ L_WALK L_WALK, and *run twice* ⟶ RUN RUN. Rather, one must acquire the more general rule that *twice* is a reduplication command which doubles any predicate x which precedes it. A function f which accurately maps between the input and output domains then needs to act uniformly on all possible *twice*-expecting predicates. Formally, we might say that if $\mathcal{S}$ is the SCAN input domain and $\mathcal{D} \subset \mathcal{S}$ is the set of predicates which can be reduplicated by *twice*, then for any permutation g of $\mathcal{S}$ we require that $f(g \cdot d) = g \cdot f(d)$ for all $d \in \mathcal{D}$; that is, if we permute *jump* to *run*, then f should likewise permute JUMP JUMP to RUN RUN. This further stipulation shows that the subset of the SCAN task which includes predicates like *twice* requires the learned functions to be locally equivariant, so as to preserve the connection between permutations of entire sequences and permutations of the tokens within each sequence.

Though Gordon et al. (2019) do not explicitly draw this connection, notice that this notion of equivariance implicitly encodes a set of equivalence classes in the input and output domains. These classes are defined exactly by the orbits of elements under action by G:

(5) **Induced equivalence classes:** For all $s, t \in \mathcal{S}$, $s \sim t$ if $G \cdot s = G \cdot t$.

These equivalence classes are the sets of elements in the input and output domains which are permuted with one another.

This notion of group action inducing equivalence classes is important because it precisely captures the notions presented in Berent & Marcus (2019) about operational equivalence classes: learners who make (algebraic) generalizations learn to partition the input domain into equivalence classes (i.e., learn the appropriate structure of a group G such that the orbits of the inputs under action by G form the notionally correct classes) and then learn uniform functions on those classes (i.e., learn an function on the input domain which is equivariant with respect to G).

That the intuition expressed in Berent & Marcus (2019) is so cleanly described by the formalism presented in Gordon et al. (2019) strongly suggests that the algebraic generalization described by Berent & Marcus is captured in the subset of compositional generalization tasks described by Gordon et al. (2019).

Taking a bird's-eye view of the landscape of generalization tasks, we turn to the first major contribution of this work, which is to extend the formalism of Gordon et al. (2019) into a typology of forms of generalization which can characterize the relative difficulty of different kinds of generalization on the basis of their formal properties. This proposed classification is shown beside in Figure 2.2. Each level on this hierarchy corresponds to requirements of either the learned group representation for the symmetries present in the underlying data of the phenomenon being modelled, or of the learned functions which map from the equivalence classes induced by these symmetries into the output domain. As the levels increase, the requirements on the group structure or the functions acting on the induced equivalence classes become stronger. Thus, a problem requiring *distributional* generalization (ℓ_1 on this typology) requires learning a simpler set of functions than are required to learn a problem equivalent to an *identity* generalization problem.

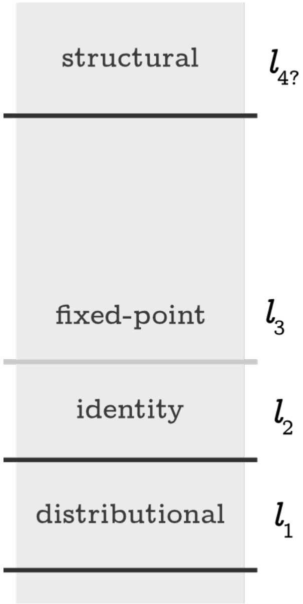

Figure 2.2: Proposed typology of generalization forms

To be absolutely explicit about the properties which distinguish one kind of generalization from another, we will introduce the following commutative diagram.

ℓ_1 **Simple** or **Distributional Generalization**. This is the kind of generalization found in the hue-classification task proposed earlier, where a model generalizes knowledge of a particular class to novel inputs on the basis of their distribution. Formally, it requires a learner to:

(a) Learn a set of equivalence classes on the input space $\mathcal{S}$ by learning the structure of the appropriate permutation group on the elements; and

(b) Learn a function f which is *invariant* on each equivalence class.

Invariance is a weakened form of equivariance, where we only require f to act invariantly on all members of a particular equivalence class; that is, $f(g \cdot a) = f(a)$ for all $a \in \alpha$ and all $g \in G$. These are functions which are constant over each equivalence class. Such generalization naturally extends to classifiers, for whom knowledge of an input's equivalence class is sufficient to determine its output, but some forms of lexical generalization in non-classifier contexts also fall into this category. Consider, for instance, the work of Kim & Smolensky (2021) who teach BERT (Devlin et al. 2019) to model a novel noun *thax* and verb *dax* in a limited distributional context and then show that the model prefers to produce the correct form in more complicated noun- and verb-accepting contexts.

Importantly, distributional generalization does not make use of the identity of elements of the input domain once they are assigned to an equivalence class; this information is collapsed and the learned mapping acts only on the identities of the equivalence classes themselves.

ℓ_2 **Identity Generalization** requires a learner to not only learn an assignment of inputs to equivalence classes but further requires that the learned mapping on those classes preserves the identity of each element in the class. Formally, it requires one to learn:

(a) A set of equivalence classes on $\mathcal{S}$ induced by learning the structure of the permutation group G; and

(b) An function g which is *equivariant* with respect to G on each equivalence class.

The identity of the elements of each equivalence class are preserved under f because each group action factors over f. This is the simplest formulation of compositional generalization in Gordon et al. (2019), and corresponds directly to the identity problem posed by G. F. Marcus (1998a), where a network is tasked with learning to apply the 'an x is an x' predicate to a novel value of x; here, a model must learn not only that a novel value of x is valid in this construction, but must also carry over knowledge of x's identity when given the prompt 'an x is a ...' and then asked to complete the expression with 'x.'

ℓ_3 **Fixed-Point Generalization** extends identity generalization to impose additional structure on the learned permutation group acting on the input domain. Specifically, a problem requires fixed-point generalization if there exists a set of inputs $\{s_1, \ldots, s_k\}$ which are fixed points of every action $g \in G$, where $g \cdot s_i = s_i$ for all $g \in G$.

The subset of the SCAN tasks which involves the reduplicative predicates *twice*, *thrice*, and so on fall into this category. These predicates are fixed points under action by G since they cannot be mapped to another predicate in the input domain and still compose with the learned interpretation function. This further restriction encompasses the anaphora resolution task posed by Frank, Mathis & Badecker (2013) as a test of algebraic generalization, where a model must learn to interpret a n expression containing a reflexive anaphora, like 'John sees himself in the mirror,' for a contextually-novel antecedent *John*. Notably, the orbit of the reflexive anaphor *himself* is trivial in the input domain, since *himself* cannot map to any other valid object of *sees* while still generalizing in the output domain (since if *himself* ⟶ *Bill*, for example, then all reflexive sentences will now have the object *Bill*), as the permutation induced by the group structure must carry-through equivariantly to the output domain.

I also tentatively propose a further class ℓ_4 which characterizes structural generalization. This class is not (yet) defined in as formal manner as the other generalization classes, but it is clear from the failures of neural models to solve structural tasks in generality (Lake & Baroni 2018) that they are qualitatively distinct from the lower kinds of generalization. Theoretically, linguistic problems requiring what we might call "full structural generalization" seem to require a more complicated relationship between input tokens and learned equivalence classes: while the classes required to characterize the identity problem put forth by G. F. Marcus (1998a) or the anaphora resolution task of Frank, Mathis & Badecker (2013) are formed seemingly on the basis of individual tokens in the input representation, knowledge of person agreement in arbitrary clauses requires knowledge equivalent to the imposition of syntactic hierarchy and recognition of a mechanism reminiscent of c-command; here, entire phrases (at least) and elements of structural positions in a tree hierarchy form equivalence classes.

In their work on modeling compositional generalization as the learning of group-equivariant functions, Gordon et al. (2019) similarly note that within the tasks broadly considered to be 'compositional generalization' there are differences in the requirements for the necessary group actions: some tasks require the learning of functions which are not local. They additionally note that even models explicitly designed with inductive biases for local group-equivariant functions are incapable of exhibit length generalization, where withheld data is qualitatively similar to the training domain but simply longer, as measured by token length. These theoretical and empirical results suggest that further rungs on this typological ladder exist between the fixed-point generalization identified as ℓ_3 and the full structural generalization identified as ℓ_4.

This typological classification of forms of generalization has the virtue of connecting various notions of generalizational capacity and providing a shared formal characterization of each. This lets us directly compare the representational complexity of different generalization tasks to better understand what makes some generalization tasks harder than others. Ultimately, this typology further serves as a predictive instrument, where each generalization class represents a particular level of computational difficulty to acquire the relevant generalization. Thus if a problem can be described using an analytical formalism presented here, we make the claim that the learnability of that task is equivalent to the learnability of other members of that generalization class.

Orthogonal Difficulties in Generalization

The typological hierarchy of generalization proposed here classifies generalization problems on the basis of the complexity of their representation in terms of symmetries and equivariant functions between the input and output domain respecting these symmetries. But this measure of difficulty is, in some sense, incomplete, since it does not take into account the measure of support that these patterns of symmetry and functional transformation have in the training data. In effect, this hierarchy ignores the size and scope of the withholding set, which limits the interpretability of a proposed task's classification in terms of its learnability by a given network.

We can see plainly that this this other measure of difficulty is orthogonal to the one captured in the hierarchy; a more difficult problem may be presented to a model with broad training support, while a comparatively simple problem may be presented with a relative paucity of training support. Chapter 3 tests this orthogonal axis of difficulty explicitly by constructing an ℓ_3-difficult task (equivalent to the anaphora resolution task of Frank, Mathis & Badecker (2013), but in a sequence-to-sequence context) and then progressively decreasing the training support shown to the model for this generalization, to the effect of degrading the performance of some models on the withheld data.

Of course, this general observation is nothing new: the relationship between training data and model performance is central to our understanding of deep learning and generalization. In a cognitive science context, child acquisition of language on the basis of extremely limited support has long been presented as the classic 'Poverty of the Stimulus' argument in favor of the presence of some 'language acquisition device' present in the human capacity for language which facilitates the formulation of syntactically-valid generalizations for how language works (Chomsky (1980), though see Pullum & Scholz (2002) for critique of whether or not this argument holds in human language acquisition).

However, it is crucial to observe that this measure of difficulty is separate from the inherent complexity of the task. We leave unexamined in the rest of this work the daunting task of providing an explicit connection between the size of training support and learnability of a given task by a model which task into consideration both the model architecture (and hence any implicit biases present in the network design) and the proposed classification of this task on the kind of hierarchy presented here.

2.3 Algebraic Generalization in Natural Language

To ground this discussion of formal generalization categories and their learnability by neural models to phenomena relevant to theoretical linguistics, it is worth exploring various observed patterns in natural language which, when translated into the formalism of the hierarchy presented here, require degrees of algebraic generalization. The presence of these phenomena within the scope of 'natural language learnable by humans' thus demonstrates the centrality of algebraic generalization to the questions of human language acquisition and language learnability by artificial neural networks.

Reduplication: Many languages exhibit morphological reduplication, where words or segments of words are productively repeated to modify the meaning of the original word in a predictable way (Urbanczyk 2017). English makes use of a type of this pattern known as *contrastive reduplication* which involves the reduplication of words or phrases to achieve a contrastive meaning with the simple, unreduplicated form (Ghomeshi et al. 2004). Consider the following (marginally adapted) examples:

(6) a. I'm up, I'm just not UP-up.
b. Yeah, but is he a doctor or is he a DOCTOR-doctor?
c. I'm eating a tuna salad, not a SALAD-salad.
d. Oh, we're not LIVING-TOGETHER living-together. [Ghomeshi et al. 2004]

In each case, a certain predicate is reduplicated with initial stress to achieve a contrast where the reduplicated form is understood to be an intensive or truer form of the predicate than the simple, unreduplicated form. This pattern is robustly productive in North-American English, and can apply to a wide class of predicates ranging from the concrete nouns of (6b) to the verbal constructions of (6d). Since this reduplication is productive, speakers of English cannot learn it merely on the basis that a *SALAD-salad* is a purer form of *salad*, or that a *DOCTOR-doctor* is a truer form of *doctor*; rather, they must acquire a more general understanding that given a predicate x with an associated interpretation, an X-x is a predictably-intensive or truer form of whatever x's interpretation is.

Root-Template Morphology: Semitic languages employ noncontatenative morphology where tuples of consonontal roots are placed into morphological templates to productively derive words (Bat-El 2003). Consider the following examples from Hebrew, wherein various triconsonontal roots are combined with two templates to produce simple and passive verbal forms.

(7)		**□a□á□**	**ni□□á□**
	K-T-B	*katáv*, ‘he read’	*niḫtáv*, ‘it was read’
	L-B-Ś	*lavâś*, ‘he wore’	*nilbáś*, ‘it was worn’
	ʾ-K-L	*ʾaḫál*, ‘he ate’	*niʾkál*, ‘it was eaten’
	P-T-Ḥ	*patáḥ*, ‘he opened’	*niftáḥ*, ‘it was opened’

Subject to some phonological modification of how each root consonant is pronounced (in certain positions, stops may be realized as fricatives), each verb form represents a kind of template into which the root consonants may be substituted to form a verb. A speaker who has acquired Hebrew well enough to internalize this pattern of production need not re-learn these output forms for when learning new roots, as knowledge of the realized forms can be abstracted into knowledge of the template pattern.[2]

Reflexive Anaphora Interpretation: Natural languages frequently employ reflexive anaphors to represent reflexive meaning in otherwise non-reflexive constructions. In learning to interpret sentences which contain such reflexive anaphora like *himself* and *herself*, learners may be exposed to various inputs like those in (8) below.

(8) a. “Alice sees herself” ⟶ SEE(ALICE, ALICE)
b. “Claire sees herself” ⟶ SEE(CLAIRE, CLAIRE)

For a limited number of possible antecedents, learners could come to the conclusion that *herself* has a number of possible interpretations based on the context of the surrounding sentence: that is, *herself* means ALICE in the context of *Alice* and CLAIRE in the context of *Claire*. However, this pattern is not general enough to extend to new names; rather, speakers acquire the more general rule that *herself* takes on the interpretation of whatever antecedent occupies the nearest position c-commanding it in the binding domain. This pattern cannot be learned as a mere extension of particular form-context pairings, but must rather be learned as an algebraically general rule.

The bulk of the remainder of this thesis will explore models which attempt to learn exactly this problem: interpreting reflexive anaphora in the context of antecedents which have not previously been seen in a reflexive context.

[2] Of course, native speakers of Hebrew must also contend with the fact that these verbal templates are not completely productive. Some roots do not have simple *pa‘al* forms, while others do not have the passive *nif‘al* forms, and so learners are likely to overgeneralize this root-template knowledge to produce ungrammatical constructions.

2.4 Algebraic Generalization in Neural Models

Given the importance of algebraic generalization in models of cognition and its demonstrated prevalence in natural language, it is not surprising that computational models of cognitive processes like language which display algebraic generalization would prove useful, both practically and theoretically (G. F. Marcus 2001). Despite this desire, initial work examining artificial neural models' ability to solve algebraically general tasks has been treated with skepticism. Early treatment of this dilemma comes in the work of G. F. Marcus, who argues that the kinds of artificial neural networks commonly used (here, taken to be the recurrent model proposed by Elman (1990)) are fundamentally incapable of learning to solve algebraically general problems (G. F. Marcus 1998a,b). This prediction derives from the argument that models trained via backpropogation are incapable of learning abstracted relationships to features of data which do not appear in the training set. This fact, according to G. F. Marcus, follows from the mathematical properties of the general backpropogation algorithm, wherein weight updates are strictly local based only on the signal available to a particular connection between neurons rather than any global information.

In this view, neural models are then fundamentally incapable of solving generalization tasks which require abstraction over an input set because the learned properties of the training inputs will never apply to novel inputs with a different distribution in the testing data. Such abstraction is required to allow for the variable scoping of (9), where a model learns to approximate a function f on the basis of limited support and then apply this knowledge to a full set of inputs.

(9) $\forall x.f(x)$ where f is some learned function

This kind of variable application is directly required for algebraic knowledge, as previously defined, since algebraic functions must implicitly apply equivariantly over equivalence classes learned on the basis of incomplete support. Hence, G. F. Marcus argues, existing neural models are incapable of algebraic generalization.

To make concrete this abstract claim about the limitations imposed via backpropagation, G. F. Marcus (1998a) offers the following experimental paradigm designed to test a neural model's ability to learn the identity function in full generality.[3] A recurrent

[3] Actually, G. F. Marcus (1998a) describes two paradigms; the first involves training models only on sentences of the form shown in (11b), and then testing the model on a wholly-novel token *lilac* $\notin$ $\{rose, tulip, lily\}$. This experiment is bound to fail simply on the premise that the recurrent networks surveyed require a fixed embedding width to function, and so are incapable of generalizing to a novel token merely because there is no analogy to 'learning a word on the fly.' Even relaxing this experiment to permit a model knowledge that such a token exists but providing it no signal whatsoever about its distribution in training data (akin to leaving the random initialization of *lilac*'s embedding unchanged over the course of training), or providing a default `<UNK>` *unknown* token to which all novel tokens are mapped is destined to fail because the model has been provided with no distributional knowledge of *lilac* whatsoever. This formal

language model is presented with training data of the forms shown in (11) below,

(11) a. The bee lands on the x.
b. A y is a y.

where $x \in \{lilac, rose, tulip, lily\}$ and $y \in \{rose, tulip, lily\}$. The model is then tested on an input of the form shown below in (12), where *lilac* appears in a distributional context not seen during training.

(12) A *lilac* is a ...

For natural language learners, the appearance of *lilac* in the training frame defined by (11a) is enough to signal that *lilac* is a valid member of a noun class, and can therefore can have the same distribution as any member of y. Felicitously completing the prompt of (12) as 'A *lilac* is a *lilac*' is therefore trivial. Yet for the neural models surveyed by G. F. Marcus (1998a), success proved elusive. Networks were unable to learn the abstraction necessary to include *lilac* in an equivalence class with other members of y, and hence were unable to learn an equivariant, algebraic function to produce the output token *lilac* conditional on the 'identity prompt' input of 'A *lilac* is a ...'

This 'identity' problem typifies the kind of task which requires algebraic generalization. It is a strictly harder task than learning to merely *accept* the sequence 'A lilac is a lilac' as valid since the model must draw an explicit connection between the novel prompt 'A lilac is a ...' and a contextually-novel output 'lilac'. To learn a generalized acceptance function on such prompts, it suffices for a model to merely learn to associate contextually novel inputs with known inputs as members of a single equivalence class, and then learn a binary function on compositions of input tokens involving members of this equivalence class. This weaker challenge is exactly the task of lexical generalization, which has been robustly shown to be within the capabilities of neural models (see, e.g., Kim & Smolensky 2021 for a recent example). Rather, to learn an identity mapping, a model must manage to further learn an equivariant map on this equivalence class, preserving the identity of the relevant token in the novel prompt to produce not just *a* valid response, but *the* valid response.

Though the domain is trivial, and hence unrepresentative of the kind of problem domain or training support presented to human language learners, the simplicity of

problem is perhaps analogous to the fact that while humans have no trouble understanding a sentence like (11b) for novel words, we do require these words to be known to us as nouns; competent speakers of English would likewise be puzzled by a prompt like (10) below, even though the 'novel' predicate *without* is infact know to us!

(10) *A* without *is a ...*

From here on, we limit our discussion of the identity problem to the more relaxed one described in (11), since this success or failure in this paradigm is more illustrative of an interpretable success or failure to acquire generalization.

the task perhaps strengthens G. F. Marcus (1998a)'s argument: the resolute failure of recurrent language models to solve this task shed doubt on their ability to acquire algebraically general knowledge and supported Berent & Marcus's theoretical argument that neural networks did not have the computational capacity to solve such problems to full generality.

Although this thesis will ultimately take a stance in opposition to Berent & Marcus's results and argue empirically and analytically that such capacity is *not* beyond reach of simple recurrent models, it is worth taking a moment to attempt to reconcile Berent & Marcus's views with the overwhelming success of contemporary neural networks at solving increasingly difficult problems across a wide variety of domains. One may be tempted to cite the incredible capacity of modern[4] generative models such as GPT-3 (Brown et al. 2020) as clear evidence that the claims made in G. F. Marcus (1998a) are incorrect. While the conclusion is, in my view, true, the argument[5] itself is not sound. The claim made by Berent & Marcus is one about neural models acting, in essence, as perfectly general machines with an arbitrary capacity for the kind of generalization demanded by the identity problem. It is not enough for such models to solve the identity problem for any arbitrarily large set of predicates on the basis of massive training support; they must be able to learn identity as applied to *any* valid predicate. This is no small task, as evidenced by the failure of our best-performing general models to solve problems requiring exactly this kind of abstraction. As a concrete counterfactual, consider the plight of language models tasked with performing arithmetic, a set of operations which makes foundational use of the notion of equality. The largest GPT-3 model, with 175 billion trainable parameters, shows categorical disparities in accuracy between two-digit and three-digit arithmetic inputs (see section 3.9.1 of Brown et al. 2020). While the middling performance of such an otherwise large and capable model on three-digit arithmetic is disappointing in the context of creating well-performing models, the disparate outcomes between two- and three-digit arithmetic prompts belies a much larger issue: the model has clearly not learned an abstract notion of 'equality' in an algebraically general way; rather, it has used massive training support to fit a devilishly large problem space. When the size of the problem space outpaces the training support, the models accuracy suffers greatly. This behavior stands in stark contrast to a hypothetical model which has acquired an algebraic knowledge of an identity/equality relation between inputs; such a model does not need arbitrarily large training support

[4] Alas, such a description is already made anachronous by the mere passage of time. What was 'modern' at the time of writing is surely archaic at the time of reading.

[5] I do not make any specific claim or citation for who may make such an argument; rather, I wish to present it along with a rebuttal simply because I find it to be a natural claim to make at first glance. One can be forgiven for seeing the amazing capacity of neural networks in stark comparison to the unassuming problem domain set forth by G. F. Marcus (1998a) and concluding that the latter's arguments must surely be refuted.

to solve arithmetic relations of arbitrary length.

Berent & Marcus's skepticism may then stand a moment longer in the face of current machine learning achievements, though the following sections will show that it may not do so indefinitely.

Inductive Biases for Algebraic Knowledge

The critique offered in G. F. Marcus (1998a) and subsequent work has been focused on 'purely neural' models, a class of model ranging from simple recurrent neural networks surveyed here to the massively large and capable transformer models which dominate the playing field today. In distinction to such models, which provide no explicit way to model the kind of symbolic relationships which he sees as necessary to achieve algebraic generalization, G. Marcus (2020) offers symbolic and hybrid neural-symbolic models as a better candidate for modelling problems which require algebraic knowledge. These models make use of explicit symbol-manipulation computational machinery to sidestep the limitations of pure neural models, and have been shown to be successful at the kinds of tasks which haunt traditional neural networks (Smolensky et al. 2016, Schlag et al. 2020).

Such designs are representative of the larger approach of building *inductive biases* into models which aid models at solving a general task by providing a model with a set of *a priori* assumptions about how the task may be solved. Inductive biases are often realized as explicit computational machinery present in a neural model to facilitate the learning of an otherwise-difficult task. Though symbolic machinery is perhaps a natural tool to reach for when faced with the prospect of learning algebraically-general rules, it is by no means the only such inductive bias which is used to improve the performance of neural models. One of the most prevalent inductive biases used today is *attention*, mechanisms which provide a model selective focus over a particular part of the input conditioned on the current output position (Schmidhuber & Huber 1990). This innovation has been adapted to linguistic domains with great success; attention mechanisms have been used widely in many of the most successful linguistic neural networks to date (Bahdanau, Cho & Bengio 2016, Vaswani et al. 2017). Beyond providing general computational capabilities, it is not hard to see how the ability to attend to specific positions in an input sequence when producing an output sequence is of great use to solving the identity problem posed in G. F. Marcus (1998a). Indeed, recent work has demonstrated that attention mechanisms provide an inductive bias for learning equivariant functions over permutations, and thus provide a clear path towards solving at least part of the challenge posed by algebraically-sensitive tasks (Goyal & Bengio 2021).

While the use of architectural design as an inductive biases towards solving the problem of acquiring algebraic knowledge may ultimately be fruitful, I am additionally interested in exploring if and how models lacking this explicit machinery manage to nevertheless solve problems requiring algebraic generalization. As will be shown chapter 3, we manage to train simple recurrent networks to solve a problem analogous to the identity task of G. F. Marcus (1998a). That models lacking any inductive bias for algebraic generalization are capable of solving this task demands elucidation, especially in light of Berent & Marcus's firm view that purely neural models lacking explicit biases are incapable of such feats.

2.5 Anaphora Resolution as a Test of Algebraic Knowledge

As mentioned in Section 2.3, the interpretation of reflexive anaphora is a phenomenon in natural language which requires algebraic generalization. Speakers understand that reflexive pronouns like *himself* and *herself* take on the meaning of that of the local c-commanding antecedent (Safir 2013). Learners come to this understanding on the basis of limited training support: children manage to learn to interpret reflexive pronouns early, even in the absence of strong attestation of these forms in a wide variety of contexts (Clackson, Felser & Clahsen 2011, O'Grady 2013). Indeed, the problem domain for reflexive anaphora is unbounded since names form an open syntactic class of nominal antecedents for such forms, meaning that even robust attestation of reflexive sentences with clear interpretations does not alleviate the problem that learning to interpret reflexive pronouns in full generality requires the acquisition of an algebraic rule which can abstract over the class of possible antecedents while preserving the identity of the particular name which binds the anaphor.

Formally, a reflexive anaphor's interpretation is governed by a c-command relation with its antecedent:

(13) **Reflexive Anaphora Resolution:** A reflexive anaphor is interpreted as identical to the nearest c-commanding noun phrase.

Thus in (14) below we see how the interpretation of the reflexive anaphor *herself* is determined by the identity of the c-commanding noun phrase.

(14) a. Mary sees herself: $[\![$ herself $]\!]$ = MARY
b. Mary's mother sees herself: $[\![$ herself $]\!]$ = MOTHER-OF-MARY

Since the resolution of reflexive anaphora is a task which both requires algebraic generalization and is learned by children in the face of limited support in child-directed speech (O'Grady 2013), it serves as a natural subject of inquiry in the computational modelling of language learning. By virtue of its generalized difficulty, anaphora resolution serves as a benchmark to test a neural model's ability to acquire the algebraic knowledge necessary for human-like performance on linguistic tasks. By virtue of its relatively uncommon support in child-directed speech, the task also serves as a good point of comparison between the capability of human language learners to acquire linguistic capacity and that of artificial neural models.

There are a number of ways which this task can be used to measure the generalization capacity of neural models depending on the context in which data is presented. The first is in a *language modelling* context, where a model is tasked with learning the conditional probability of words in an output distribution based on the surrounding words present in the input (Sutskever, Martens & Hinton 2011). Language models are used to produce time-dependent predictions based on a given input sequence: given an enumeration of an input sequence, as shown below in (15), a model produces a prediction for each step in the sequence. In this context, a model may be trained to produce an output interpretation for each input token at that token's respective time-step.

(15)

$$\begin{array}{ccc} \text{ALICE} & \text{SEES} & \text{ALICE} \\ \uparrow & \uparrow & \uparrow \\ \textit{Alice} & \textit{sees} & \textit{herself} \\ t_0 & t_1 & t_2 \end{array}$$

For reflexive anaphora resolution, language model which successfully learns this task will learn to assign each word in the input with its semantic output. For nouns and verbs, this task is trivial, as it involves merely learning to associate particular input tokens with their corresponding output tokens. Thus, *Alice* ⟶ ALICE, *Mary* ⟶ MARY, *sees* ⟶ SEES, and so on. For reflexive anaphora, however, the task is more difficult. Given a context of a sentence beginning with *Alice ...*, the model must learn to associate *herself* ⟶ ALICE, while learning to associate *herself* ⟶ MARY in the context of *Mary*, and so on for each name which can appear as an antecedent to the reflexive pronoun. For a model to generalize to arbitrary antecedents, it is not enough to learn particular context-form mappings: rather, a language model must be able to extract from an arbitrary conditional context the identity of the relevant antecedent in a single time-step. Generalization can then be tested by withholding a subset of possible names in a reflexive context during training, and then testing the model on reflexive sentences involving these names. Models which can successfully resolve reflexive anaphors to withheld antecedents have acquired the algebraically-generalized rule for resolution.

This is the experimental setup of Frank, Mathis & Badecker (2013), which seeks to explore the challenge posed by G. F. Marcus (1998a) in a linguistically-interesting context. Here, reflexive sentences with the antecedent *John* were withheld from the training support of a simple recurrent language model tasked with providing interpretations to inputs. Although such models face little difficulty at learn to interpret in-domain inputs, Frank, Mathis & Badecker find that they fail categorically when given sentences like *John sees himself* from outside the training domain. This uniform failure mirrors the findings of G. F. Marcus (1998a)'s *lilac*-identity experiment and provides further experimental evidence in support of the claim that simple recurrent models are unable to learn algebraically general rules.

Interestingly, though all networks surveyed resoundingly failed to generalize, Frank, Mathis & Badecker (2013) do make an important observation about the latent representations of the models' penultimate layer of hidden units: while unable to generate the correct probability distribution over the set of possible output tokens, the models did manage to learn to represent the input vectors in such a way as to render the correct output's representation as linearly separable from the incorrect outputs' representations in all cases (that is, for each input context of *x verbs self*, the vector corresponding to the correct interpretation is linearly separable from all others for every antecedent, including those withheld during training). This suggests that the model has learned *something* about the task in a general way, although it is unable to capitalize on this knowledge to solve the full task in generality.

A second way to test for algebraic generalizational capacity via an anaphora resolution task comes in a sequence-to-sequence paradigm. Unlike language modelling, where a model produces a single output for each input in an ordered fashion, a sequence-to-sequence task separates computations on the input and output sequences from one another (Sutskever, Vinyals & Le 2014). By removing the time-step and length dependence of the output on the input, a sequence-to-sequence model is able to map arbitrary input sequences to arbitrary output sequences via a compressed representation of the input. These models are bicameral: the first part of the model, known as the encoder, processes an input sequence and produces a hidden-vector representation of the input. This input is then passed to the decoder, which uses this representation to produce a series of output tokens until it generates a [STOP] token.

Treated as a sequence-to-sequence task, reflexive anaphora resolution is then seen as a mapping between reflexive input sentences and formal representations of their interpretation, as shown below in (16).

(16) Mary sees herself ⟶ SEE(MARY, MARY)

Here, a model fully encodes the input *Mary sees herself* and is tasked with decoding the representation of this sequence into the ordered sequence of output tokens 'SEE', '(', 'MARY', ',', 'MARY', ')'. Just as in the language-modelling paradigm, this sequence-to-sequence task can be structured to test not only whether a model can learn this task to arbitrary accuracy but also whether it can learn to generalize it's knowledge of this context-form-token mapping to novel reflexive antecedents. By withholding some subset of possible antecedents in a reflexive context from a model and then testing a model on these inputs we can determine whether or not a model is able to learn the resolution rule in generality. This is the experiment of the first empirical result of this thesis, Chapter 3, which tests sequence-to-sequence models by withholding all sentences involving *Alice* as the antecedent of a reflexive anaphor.

As will be elaborated upon further in the coming sections, treating the anaphora resolution task as a sequence-to-sequence problem produces markedly different results from those of the language-modelling problem. In a sequence-to-sequence context, even simple recurrent networks are able to learn an algebraically-generalized pattern to arbitrary accuracy, exhibiting perfect generalization with the withheld set of *Alice*-reflexive pronouns. The success of such simple models at solving a task requiring algebraic generalization is thus notably for two main reasons: First, it provides a constructive counterexample to G. F. Marcus (1998a)'s view that such algebraic knowledge is beyond the generalizational capacity of pure neural networks without algebraic inductive biases; second, it raises a number of important questions about how these models are able to solve this task: By what mechanism are inattentive recurrent models able to learn algebraically-general rules? What benefit does the sequence-to-sequence paradigm confer over a language modelling paradigm? Where in the model does anaphora resolution actually take place? Can we provide a formal interpretation for how such models make use of their encoding space to represent inputs? The remaining chapters of Part I will explore these questions in more detail.

Part I

Sequence-to-sequence Models

Chapter 3

Recurrent Resolution of Reflexive Referents

3.1 Abstract:

Reflexive anaphora present a challenge for semantic interpretation: their meaning varies depending on context in a way that appears to require abstract variables. Past work has raised doubts about the ability of recurrent networks to meet this challenge. In this paper, we explore this question in the context of a fragment of English that incorporates the relevant sort of contextual variability. We consider sequence-to-sequence architectures with recurrent units and show that such networks are capable of learning semantic interpretations for reflexive anaphora which generalize to novel antecedents. We explore the effect of attention mechanisms and different recurrent unit types on the type of training data that is needed for success as measured in two ways: how much lexical support is needed to induce an abstract reflexive meaning (i.e., how many distinct reflexive antecedents must occur during training) and what contexts must a noun phrase occur in to support generalization of reflexive interpretation to this noun phrase?

Recurrent neural network architectures have demonstrated remarkable success in natural language processing, achieving state of the art performance across an impressive range of tasks ranging from machine translation to semantic parsing to question answering (Sutskever, Vinyals & Le 2014, Cho et al. 2014, Bahdanau, Cho & Bengio 2016). These tasks demand the use of a wide variety of computational processes and information sources (from grammatical to lexical to world knowledge), and are evaluated in coarse-grained quantitative ways. As a result, it is not an easy matter to identify the specific strengths and weaknesses in a network's solution of a task.

In this paper, we take a different tack, exploring the degree to which neural networks successfully master one very specific aspect of linguistic knowledge: the interpretation of sentences containing reflexive anaphora. We address this problem in the context of the task of semantic parsing, which we instantiate as mapping a sequence of words into a predicate calculus logical form representation of the sentence's meaning.

(17) a. Mary runs → RUN(MARY)
b. John sees Bob → SEE(JOHN, BOB)

Even for simple sentences like those in (17), which represent the smallest representations of object reflexives in English, the network must learn lexical semantic correspondences (e.g., the input symbol *Mary* is mapped to the output MARY and *runs* is mapped to RUN) and a mode of composition (e.g., for an intransitive sentence, the meaning of the subject is surrounded by parentheses and appended to the meaning of the verb). Of course, not all of natural language adheres to such simple formulas. Reflexives, words like *herself* and *himself*, do not have an interpretation that can be assigned independently of the meaning of the surrounding context.

(18) a. Mary sees herself → SEE(MARY, MARY)
b. Alice sees herself → SEE(ALICE, ALICE)

In these sentences, the interpretation of the reflexive is not a constant that can be combined with the meaning of the surrounding elements. Rather, a reflexive object must be interpreted as identical to the meaning of verb's subject. Of course, a network could learn a context-sensitive interpretation of a reflexive, so that for any sentence with *Mary* as its subject, the reflexive is interpreted as MARY, and with *Alice* as its subject it is interpreted as ALICE. However, such piecemeal learning of reflexive meaning will not support generalization to sentences involving a subject that has not been encountered as the antecedent of a reflexive during training, even if the interpretation of the subject has occurred elsewhere. What is needed instead is an interpretation of the reflexive that is characterized not as a specific (sequence of) output token(s), but rather as an abstract instruction to duplicate the interpretation of the subject. Such an abstraction requires more than the "jigsaw puzzle" approach to meaning that simpler sentences afford.

G. F. Marcus (1998a) argues that this kind of abstraction, which he takes to require the use of algebraic variables to assert identity, is beyond the capacity of recurrent neural networks. G. F. Marcus's demonstration involves a simple recurrent network (SRN, Elman 1990) language model that is trained to predict the next word over a corpus of sentences of the following form:

(19) a. A rose is a rose.
b. A mountain is a mountain.

All sentences in this training set have identical subject and object nouns. G. F. Marcus shows, however, that the resulting trained network does not correctly predict the subject noun when tested with a novel preamble '*A book is a* …'. Though intriguing, this demonstration is not entirely convincing: since the noun occurring in the novel preamble, *book* in our example, did not occur in the training data, there is no way that the network could possibly have known which (one-hot represented) output should correspond to the reflexive for a sentence containing the novel (one-hot represented) subject noun, even if the network did successfully encode an identity relation between subject and object.

Frank, Mathis & Badecker (2013) explore a related task in the context of SRN interpretation of reflexives. In their experiments, SRNs were trained to map input words to corresponding semantic symbols that are output on the same time step in which a word is presented. For most words in the vocabulary, this is a simple task: the desired output is a constant function of the input (*Mary* corresponds to MARY, *sees* to SEE, etc.). For reflexives however, the target output depends on the subject that occurs earlier in the sentence. Frank, Mathis & Badecker tested the network's ability to interpret a reflexive in sentences containing a subject that had not occurred as a reflexive's antecedent during training. However, unlike Marcus' task, this subject and its corresponding semantic symbol did occur in other (non-reflexive) contexts in the training data, and therefore was in the realm of possible inputs and outputs for the network. Nonetheless, none of the SRNs that they trained succeeded at this task for even a single test example.

Since those experiments were conducted, substantial advances have been made on recurrent neural network architectures, some of which have been crucial in the success of practical NLP systems.

- **Recurrent units**: More sophisticated recurrent units like LSTMs (Graves & Schmidhuber 2005) and GRUs (Cho et al. 2014) have been shown to better encode preceding context than SRNs.
- **Sequence-to-Sequence architectures**: The performance of network models that transduce one string to another, used in machine translation and semantic parsing, has been greatly improved by the use of independent encoder and decoder networks (Sutskever, Vinyals & Le 2014).
- **Attention mechanism**: The ability of a network to produce contextually appropriate outputs even in the context of novel vocabulary items has been facilitated by content-sensitive attention mechanisms (Bahdanau, Cho & Bengio 2016, Luong, Pham & Manning 2015).

These innovations open up the possibility that modern network architectures may well be able to solve the variable identity problem necessary for mapping reflexive sentences to their logical form. In the experiments we describe below, we explore whether this is the case.

3.2 Experimental Setup

Our experiments take the form of a semantic parsing task, where sequences of words are mapped into logical form representations of meaning. Following Dong & Lapata (2016), we do this by means of a sequence-to-sequence architecture Sutskever, Vinyals & Le 2014 in which the input sentence is fully processed by an encoder network before it is decoded into a sequence of symbols in the target domain (cf. Botvinick & Plaut 2006, Frank & Mathis 2007 for antecedents). This approach removes the need to synchronize the production of output symbols with the input words, as in Frank, Mathis & Badecker (2013), allowing greater flexibility in the nature of semantic representations.

The sequence-to-sequence architecture is agnostic as to the types of recurrent units for the encoding and decoding phases of the computation, and whether the decoder makes use of an attention mechanism. Here, we explore the effects of using different types of recurrent units and including attention or not. Specifically, we examine the performance and training characteristics of sequence-to-sequence models based on SRNs, GRUs, and LSTMs with and without multiplicative attention (Luong, Pham & Manning 2015).

In all experiments, we perform 5 runs with different random seeds for each combination of recurrent unit type (one layer of SRN, LSTM or GRU units for both the encoder and decoder) and attention (with or without multiplicative attention). All models used hidden and embedding of size of 256. Training was done using Stochastic Gradient Descent with learning rate of 0.01. Models were trained for a maximum of 100 epochs with early stopping when validation loss fails to decrease by 0.005 over three successive epochs.

We conduct all of our experiments with synthetic datasets from a small fragment of English sentences generated using a simple context-free grammar. This fragment includes simple sentences with transitive and intransitive verbs. Subjects are always proper names and objects are either proper names or a reflexive whose gender matches that of the subject. Our vocabulary includes 8 intransitive verbs, 7 transitive verbs, 15 female names, and 11 male names. The grammar thus generates 5,122 distinct sentences. All sentences are generated with equal probability, subject to the restrictions imposed by each experiment. We use a unification extension to CFG to associate each sentence with a predicate calculus interpretation. The symbols corresponding to the predicates and the entities in our logical language are identical with the verbs and names used by our grammar, yielding representations like those shown in (17) and (18). The output sequences corresponding to the target semantic interpretations include parentheses and commas as separate symbols. Quite clearly, this dataset does not reproduce the richness of English sentence structure or the distribution of reflexive anaphora, and we leave the exploration of syntactically richer domains for future work. However, even this simple fragment instantiate the kind of contextual variable interpretation found in all cases of reflexive interpretation and therefore it allows us to probe the ability of networks to induce a representation of such meanings.

As discussed in the previous section, we are interested in whether sequence-to-sequence models can successfully *generalize* their knowledge of the interpretation of sentences containing reflexives to ones having novel antecedents. To do this, we employ a *poverty of the stimulus* paradigm that tests for systematic generalization beyond a finite (and ambiguous) set of training data Chomsky 1980. In our experiments, we remove certain classes of examples from the training data set and test the effect on the network's success in interpreting reflexive-containing sentences. Each of our experiments thus defines a set of sentences that are withheld during training. The non-withheld sentences are randomly split 80%–10%–10% between training, validation, and testing sets. Accuracy for each set is computed on a sentence-level basis, i.e., an accurate output requires that all symbols generated by the model be identical to the target. Our experiments focus on two sorts of manipulations of the training data: (1) varying the number of lexical items that do and do not occur as the antecedents of reflexives in the training set, and (2) varying the syntactic positions in which the non-antecedent names occur. As we will see, both of these manipulations substantially impact the success of reflexive generalization in ways that vary across network types.

3.3 Experiment 1: Can Alice know herself?

In the first experiment, we directly test whether or not networks can generalize knowledge of how to interpret *herself* to a new antecedent. We withhold all examples whose input sequence includes the reflexive *herself* bound by the single antecedent *Alice*, of the form shown in (20).

(20) Alice *verbs* herself → *verb*(ALICE, ALICE)

Sentences of any other form are included in the training-validation-test splits, including those where *Alice* appears without binding a reflexive.

Results

All network architectures were successful in this task, generalizing the interpretation of *herself* to the novel antecedent *Alice*. Even the simplest networks, namely SRN models without attention, achieve 100% accuracy on the generalization set (sentences of the form shown in (20)). This is in sharp contrast the negative results obtained by Frank, Mathis & Badecker (2013), suggesting an advantage for training with a language with more names as well as for instantiating the semantic parsing task in a sequence-to-sequence architecture as opposed to a language model.

3.4 Experiment 2: Doesn't Alice know Alice?

While the networks in Experiment 1 are not trained on sentences of the form shown in (20), they are trained on sentences that have the same target semantic form, namely sentences in which *Alice* occur as both subject and object of a transitive verb.

(21) Alice *verbs* Alice ⟶ $verb(\text{ALICE}, \text{ALICE})$

In Experiment 2 we consider whether the presence of such semantically reflexive forms in the training data is helpful to networks in generalizing to syntactically reflexive sentences. We do this by further excluding sentences of the form in (21) from the training data.

Results

All architectures except SRNs without attention generalize perfectly to the held out items. Inattentive SRNs also generalize quite well, though only at a mean accuracy of 86%. While success at Experiment 1 demonstrates the networks' abilities to generalize to novel input contexts, success at Experiment 2 highlights how models can likewise generalize to produce entirely new outputs.

3.5 Experiment 3: Who's Alice and who's Claire?

So far, we have considered generalization of reflexive interpretation to a single new name. One possible explanation of the networks' success is that they are simply defaulting to the (held-out) ALICE interpretation when confronted with a new antecedent, as an elsewhere interpretation (but see Gandhi & Lake (2020) for reasons for skepticism). Alternatively, even if the network has acquired a generalized interpretation for reflexives, it may be possible that this happens only when the training data includes overwhelming lexical support (in Experiments 1 and 2, 25 out of the 26 names in our domain appeared in the training data as the antecedent of a reflexive). To explore the contexts under which networks can truly generalize to a range of new antecedents, we construct training datasets in which we progressively withhold more and more names in sentences of the forms shown in (22), i.e., those that were removed in Experiment 2.[1]

(22) a. P *verbs* herself ⟶ $verb(P, P)$
b. P *verbs* P ⟶ $verb(P, P)$

[1] Since *himself* and *herself* are different lexical items, it is unclear if the network will learn their interpretations together, and whether sentences containing *himself* will provide support for the interpretation of sentences containing *herself*. We therefore withhold only sentences of this form with names of a single gender. We have also experimented with witholding masculine reflexive antecedents from the training data, but the main effect remains the number of female antecedents that is withheld.

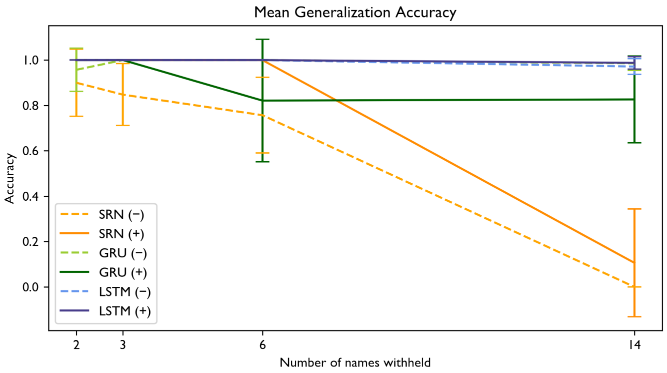

Figure 3.1: Mean generalization accuracy by number of names withheld in Experiment 3. The (+) or (−) next to the type of recurrent unit indicates the presence or absence of attention. Error bars display the standard deviation of accuracies.

Our domain contains 15 distinct feminine antecedents; we perform several iterations of this experiment, withholding progressively more feminine names from appearing in the contexts in (22), until only a single feminine name is included in the training data as the antecedent of a reflexive.

Results

As shown in Figure 3.1, reducing the set of names that serve as antecedents to reflexives in the training data resulted in lower accuracy on the generalization set. SRNs, especially without attention, show significantly degraded performance when high numbers of names are withheld from reflexive contexts during training. With attention, SRN performance degrades only when reflexives are trained with a single feminine antecedent (i.e., 14 names are held out). In contrast, LSTMs both with and without attention maintain near-perfect accuracy on the generalization set even when the training data allows only a single antecedent for the feminine reflexive *herself*. The performance of GRUs varies with the presence of an attention mechanism: without attention, GRUs achieve near perfect generalization accuracy even for the most demanding case (training with a single feminine antecedent), while the performance of GRUs with attention has mean accuracy near 80%.

We also explored how recurrent unit type and attention affect *how* models learn to generalize. One way to gauge this is by examining how quickly networks go from learning reflexive interpretation for a single name to learning it for every name. Table 3.1 shows the mean number of epochs it takes from when a network attains 95% accuracy

Architecture	*# contexts withheld*			
	2	3	6	14
SRN (−)	7.5	5.0	—	—
SRN (+)	0.6	0.6	0.6	—
GRU (−)	1.8	2.2	3.4	9.4
GRU (+)	2.2	3.6	5.3	1.5
LSTM (−)	1.2	2.2	4.4	12.2
LSTM (+)	0.6	0.8	1.4	3.4

Table 3.1: Average number of epochs between having learned one context and having learned all contexts, calculated as the mean difference among runs which succeeded in eventually learning all contexts once. A '—' in a row indicates that no models were able to achieve this degree of generalization.

on a single antecedent contexts[2] to when it has attained more than 95% accuracy on *all* held out antecedent contexts.[3]

This 'time to learn' highlights the disparate impact of attention depending on the type of recurrent unit; SRNs with attention and LSTMs with attention acquire the generalization much faster than their attentionless counterparts, while attention increases the length of time it takes for GRUs to learn for all but the condition in which 14 antecedents were withheld. Figure 3.2 illustrates another important aspect of reflexive generalization: it proceeds in a piecemeal fashion, where networks first learn to interpret reflexives for the trained names and then generalize to the held out antecedents one by one. In Figure 3.2 we show an SRN without attention, but the same pattern is representative of the other networks tested.

3.6 Experiment 4: What if Alice doesn't know anyone?

The experiments we have described thus far removed from the training data input sentences and logical forms that were exactly identical to those associated with reflexive sentences. The next pair of experiments increases the difficulty of the generalization task still further, by withholding from the Experiment 2 training data all sentences containing the withheld reflexive antecedent, *Alice*, in a wider range of grammatical contexts, and testing the effect that this has on the network's ability to interpret *Alice*-reflexive sentences.

[2] An 'antecedent context' is the set of all reflexive sentences with a particular antecedent.

[3] Note that this doesn't mean that models retained more than 95% accuracy on all contexts — some models learned a context, only to forget it later in training; this measurement does not reflect any such unlearning by models.

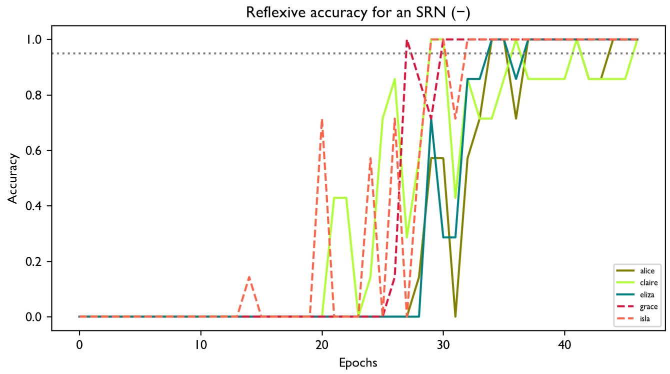

Figure 3.2: Reflexive accuracy with different antecedents during training of an SRN without attention. *Alice*, *Claire* and *Eliza* were withheld during training while *Grace* and *Isla* present in the training data.

Experiment 4a starts by withholding sentences where *Alice* appears as the subject of a transitive verb (including those with reflexive objects, which we already removed in earlier experiments). This manipulation tests the degree to which the presence of *Alice* as a subject more generally is crucial to the network's generalization of reflexive sentences to a novel name. We also run a variation of this experiment (Experiment 4b) in which sentences containing *Alice* as the subject of intransitives are also removed, i.e., sentences of the following form:

(23) Alice *verbs* ⟶ *verb*(ALICE)

If subjecthood is represented in a uniform manner across transitive and intransitive sentences, the absence of such sentences from the training data might further impair the network's ability to generalize to reflexive sentences.

Results

Experiment 4a The left plot in Figure 3.3 shows the reflexive generalization accuracy for the runs of the different architectures in the first variant of this experiment. Models without attention uniformly perform poorly across all recurrent unit types. With attention, performance is more variable: LSTMs perform at ceiling and SRNs do well for most random seeds, while GRUs perform poorly for most initializations with a single seed performing at ceiling. The top portion of Table 3.2 contrasts the means of these results with the generalization performance on transitives with *Alice* subjects. Here again LSTMs without attention performed poorly while those with attention did much worse on *Alice*-transitives than on *Alice*-reflexive sentences.

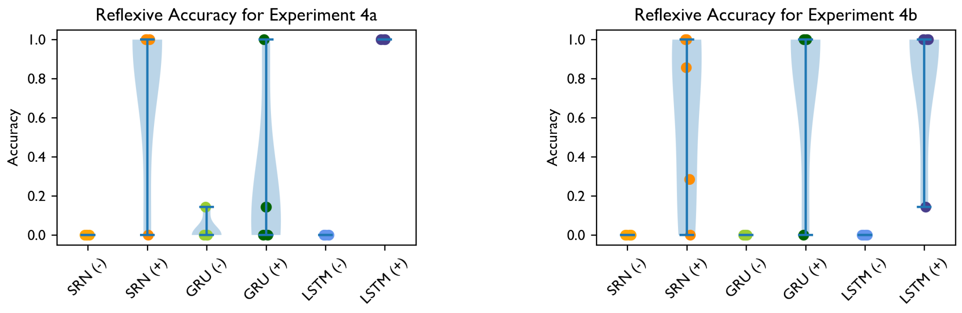

Figure 3.3: Mean accuracy on *Alice*-reflexive sentences in Experiments 4a (left) and 4b (right).

Experiment 4a	SRN (−)	SRN (+)	GRU (−)	GRU (+)	LSTM (−)	LSTM (+)
Alice-reflexive	0.00	0.80	0.03	0.26	0.00	**1.00**
Alice-subject (trans)	0.02	**0.83**	0.04	0.29	0.03	0.28

Experiment 4b	SRN (−)	SRN (+)	GRU (−)	GRU (+)	LSTM (−)	LSTM (+)
Alice-reflexive	0.00	0.63	0.00	0.80	0.00	**0.83**
Alice-subject (trans)	0.00	0.25	0.01	**0.78**	0.03	0.23
Alice-subject (intrans)	0.00	0.80	0.58	0.95	0.98	**1.00**

Table 3.2: Mean accuracy on generalization sets for Experiments 4a and 4b.

This result at once highlights the role that attention plays in learning this type of systematic generalization; attention appears to be necessary for recurrent architectures to generalize in this context. The pattern of results also demonstrates a substantial effect of model architecture: attentive SRNs substantially outperform the more complex LSTM and GRU architectures on generalization to *Alice*-transitives, though this was not the case for reflexive sentences, where LSTMs showed a substantial advantage.

Experiment 4b The right plot in Figure 3.3 shows the impact of withholding *Alice*-intransitive sentences from training. As before, models without attention fail on interpreting *Alice*-reflexive sentences. LSTMs and SRNs with attention perform nearly as well as in Experiment 4a, with some seeds performing at ceiling and a somewhat larger number than before failing to doing so. In contrast, the performance of attentive GRUs is improved in this context. The bottom of Table 3.2 shows the mean generalization accuracy for transitive and intransitive sentences with *Alice* subjects. In some cases the transitive subject performance is as in Experiment 4a or worse, but in one case, namely attentive GRUs, it improves in this more difficult context, paralleling what we saw for reflexive generalization.

The reversal of GRU (+) and SRN (+) accuracies better lines up with what we might expect given the complexity of the network architectures, with the more complex GRUs now outperforming the simpler SRNs. These results also reinforce the connection observed in those from Experiment 4b on the effects of attention in generalization.

While withholding more information during training as we move from Experiment 4a to 4b might be expected to impair generalization for attentive GRUs, as it did for all other architectures, we in fact see an increase in performance on *Alice*-reflexive sentences. One possible explanation of this surprising result is that the attentive GRU networks in experiment 4a have learned from the training data a context-sensitive regularity concerning the distribution of the withheld name *Alice*, namely that it occurs only as the subject of intransitive verbs. In Experiment 4b, however, the absence of evidence concerning the types of predicates with which *Alice* may occur allows the network to fall back to a context-free generalization about *Alice*, namely that it has the same distribution as the other names in the domain. Note that this explanation is possible only if the network treats intransitive and transitive subjects in a similar way.

3.7 Experiment 5: What if nobody knows Alice?

In the final experiment, we restrict the grammatical context in which *Alice* appears by removing from the training data of Experiment 2 all instances of transitive sentences with *Alice* in object position (but it is retained in subject position, apart from reflexive sentences). In a second variant (Experiment 5b), we further restrict the training data to exclude all intransitive sentences with *Alice* subjects. Although English, as a language with nominative-accusative alignment, treats subjects of intransitives in a grammatically parallel fashion to subjects of transitives, other languages (with ergative-absolutive alignment) treat intransitive subjects like transitive objects. Though the word order of our synthetic language suggests nominative-accusative alignment, intransitive subjects have in common with transitive objects being the final argument in the logical form, which might lead to them being treated in similar fashion.

Results

Experiment 5a The left plot in Figure 3.4 shows reflexive generalization accuracy when the missing antecedent *Alice* is withheld from transitive objects. In contrast to

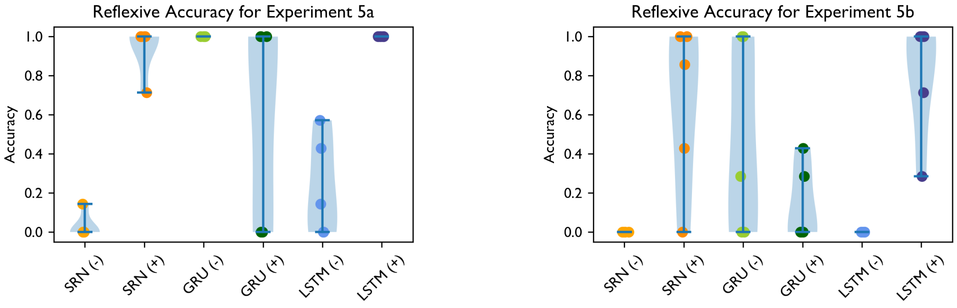

Figure 3.4: Mean accuracy on *Alice*-reflexive sentences in Experiments 5a (left) and 5b (right).

Experiment 5a	SRN (−)	SRN (+)	GRU (−)	GRU (+)	LSTM (−)	LSTM (+)
Alice-reflexive	0.03	0.94	0.98	0.60	0.23	**1.00**
Alice-object	0.00	**0.97**	0.04	0.25	0.04	0.37
Experiment 5b	SRN (−)	SRN (+)	GRU (−)	GRU (+)	LSTM (−)	LSTM (+)
Alice-reflexive	0.00	0.65	0.45	0.14	0.00	**0.80**
Alice-object	0.00	**0.94**	0.03	0.09	0.03	0.17
Alice-subject (intrans)	0.00	0.13	0.00	0.00	0.00	**0.40**

Table 3.3: Mean accuracy on generalization sets for Experiments 5a and 5b.

the results in Experiment 4, the effect of attention is more varied here. While SRNs and LSTMs without attention perform poorly, GRUs without attention perform well (for some seeds). As the top panel in Table 3.3 shows, no models without attention performed well on sentences with *Alice* in object position. For the models with attention, SRNs and LSTMs performmed uniformly well while the performance of GRUs was more mixed. On *Alice*-object sentences attentive SRNs again showed excellent performance, whereas the GRUs and LSTMs fared less well. At the same time, while GRUs with attention outperformed GRUs without attention on *Alice*-object sentences (25% to 4%), they greatly underperformed them on the reflexive sentences (60% to 98%).

Experiment 5b The right plots in Figure 3.4 shows the effects of further withholding *Alice*-intransitive sentences for *Alice*-reflexive sentences. This manipulation has devastating effects on the performance of all models without attention. For models with attention, there is also a negative impact on reflexive generalization, but not as severe. As shown in the bottom portion of Table 3.3, this manipulation has little impact on the network's performance on *Alice*-object sentences, with SRNs with attention continuing to perform strongly and the other models performing less well. GRUs continue to interact with attention in unusual ways. While they perform poorly on *Alice*-object and *Alice*-intransitive sentences with and without attention, inattentive GRUs continue to outperform attentive ones on reflexive sentences.

Overall, as in Experiment 4, LSTMs with attention show the highest accuracy on the *Alice*-reflexive sentences by a wide margin, while SRNs with attention attain the best performance on *Alice*-object sentences. Unlike in Experiment 4, withholding the *Alice*-intransitive sentences from training does not yield any benefit for GRUs with attention in performance on the reflexive set, in fact the opposite is true. This may be interpreted once again as evidence that GRUs are treating transitive and intransitive subjects as belonging to the same category. In Experiment 5a, *Alice* occurs in both positions, leading the network to treat it as a subject like any other, and therefore potentially capable of serving as a subject of a reflexive. *Alice*'s absence from object position does not impact the formation of this generalization. In Experiment 5b, on the other hand, where *Alice* occurs only as a transitive subject, it leads the attentive GRU to treat it as name with a distinctive distribution, which impairs generalization to reflexive sentences.

3.8 Conclusions

Because of their abstract meaning, reflexive anaphora present a distinctive challenge for semantic parsing that had been thought to be beyond the capabilities of recurrent networks. The experiments described here demonstrate that this was incorrect. Sequence-to-sequence networks with a range of recurrent unit types are in fact capable of learning an interpretation of reflexive pronouns that generalizes to novel antecedents. Our results also show that such generalization is nonetheless contingent on the appearance of the held-out antecedent in a variety of syntactic positions as well as the diversity of antecedents providing support for the reflexive generalization. Additionally successful generalization depends on the network architecture in ways that we do not fully understand. It is at present unknown whether the demands that any of these architecture impose on the learning environment for successful learning of reflexives are consistent with what children experience, but this could be explored with both corpus and experimental work. Future work will also be necessary to elucidate the nature of the networks' representations of reflexive interpretation and to understand how they support lexical generalization (or not).

The question we have explored here is related to, but distinct from, the issue of systematicity (Fodor & Pylyshyn 1988, Hadley 1994), according to which pieces of representations learned in distinct contexts can freely recombine. This issue has been addressed using sequence-to-sequence architectures in recent work with the synthetic SCAN robot command interpretation dataset (Lake & Baroni 2018) and on language modeling (Kim & Linzen 2020), in both cases with limited success. One aspect of the SCAN domain that is particularly relevant to reflexive interpretation is commands involving adverbial modifiers such as *twice*. Commands like *jump twice* must be interpreted by duplicating the meaning of the verb, i.e., as JUMP JUMP, which is similar to what we require for the interpretation of the reflexive object, though in a way that does not require sensitivity to syntactic structure that we have not explored here. Recently, Lake (2019), Li et al. (2019) and Gordon et al. (2019) have proposed novel architectures that increase systematic behavior, and we look forward to exploring the degree to which these impact performance on reflexive interpretation.

Our current work has focused exclusively on recurrent networks, ranging from SRNs to GRUs and LSTMs. Recent work by Vaswani et al. (2017) shows that Transformer networks attain superior performance on a variety of sequence-to-sequence tasks while dispensing with recurrent units altogether. Examining both the performance and training characteristics of Transformers will allow us to compare the effects of attention and recurrence on the anaphora interpretation task. This is especially interesting given the impact that attention had on performance in our experiments.

Finally, while our current experiments are revealing about the capacity of recurrent networks to learn generalizations about context-sensitive interpretation, there are nonetheless limited in a number of respects because of simplifications in the English fragment we use to create our synthetic data. Reflexives famously impose a structural requirement on their antecedents (c-command). In the following example, the reflexive's antecedent must be STUDENT and cannot be TEACHER.

(24) The student near the teacher sees herself → SEE(STUDENT, STUDENT)

We do not know whether the architectures that have succeed on our experiments would do similarly well if the relevant generalization required reference to (implicit) structure. Past work has explored the sensitivity of recurrent networks to hierarchical structure, with mixed results (Linzen, Dupoux & Goldberg 2016, McCoy, Frank & Linzen 2020). In ongoing work, we are exploring this question by studying more complex synthetic domains both with the kind of recurrent sequence-to-sequence network used here as well networks that explicitly encode or decode sentences in a hierarchical manner. A second simplification concerns the distribution of reflexives themselves. English reflexives can appear in a broader range of syntactic environments apart from transitive objects (Storoshenko 2008). It would be of considerable interest to explore the reflexive interpretation in a naturalistic setting that incorporate this broader set of distributions.

Chapter 4

Architectural Effects on Generalization Strategies

4.1 Abstract

Natural language exhibits patterns of hierarchically governed dependencies, in which relations between words are sensitive to syntactic structure rather than linear ordering. While recurrent network models often fail to generalize in a hierarchically sensitive way McCoy, Frank & Linzen 2020 when trained on ambiguous data, the improvement in performance of newer Transformer language models Vaswani et al. 2017 on a range of syntactic benchmarks trained on large data sets Goldberg 2019, Warstadt et al. 2019 opens the question of whether these models might exhibit hierarchical generalization in the face of impoverished data. In this paper we examine patterns of structural generalization for Transformer sequence-to-sequence models and find that not only do Transformers fail to generalize hierarchically across a wide variety of grammatical mapping tasks, but they exhibit an even stronger preference for linear generalization than comparable recurrent networks.

4.2 Introduction

One of the fundamental properties of human languages is their sensitivity to relations among elements that are not easily characterized in linear terms. In phenomena like subject-verb agreement or reflexive anaphora, the relationship between the agreeing verb and its agreement target or the reflexive pronoun and its antecedent is not governed by linear properties like adjacency or recency, but instead by the hierarchical organization of the sentence. Similarly, the relationship between related sentences, which are represented in some grammatical theories as transformational operations or as lexical rules in others, is also governed by hierarchical organization. English polar questions, for instance, involve the fronting of an auxiliary verb in the corresponding declarative to a sentence-initial position. Questions with complex subjects like (25a) demonstrate that the verb that is fronted in such cases is the determined by hierarchical prominence (i.e., move-main yielding (25b)) and not linear considerations (move-first yielding (25c) or move-last yielding (25d)).

(25) a. [The president who **can** smile] **will** lead [those who **would** sing].
b. **Will** the president who **can** smile __ lead those who **would** sing?
c. * **Can** the president who __ smile **will** lead those who **would** sing?
d. * **Would** the president who **can** smile **will** lead those who __ sing?

Chomsky (1971) argues that, in spite of receiving little input of the form in (25b), which would unambiguously demonstrate the necessity for a hierarchically governed dependency, children uniformly generalize the process of question formation in a hierarchical fashion. Such consistent behavior suggests that humans possess an inherent bias of some sort towards hierarchical generalization (though see Ambridge, Rowland & Pine (2008) and Perfors, Tenenbaum & Regier (2011) for arguments against this view). Replicating such a bias in generalization would indicate the ability to mimic patterns of human cognition and learning.

Previous investigations of recurrent neural architectures have yielded some evidence for hierarchically-governed linguistic knowledge Gulordava et al. 2018, Marvin & Linzen 2018, Hu et al. 2020. Even greater success has been achieved with neural networks the incorporate explicit representation of syntactic structure Kuncoro et al. 2018. Architecturally-constrained models when trained without explicit information about syntactic structure show only modest benefits Shen et al. 2018, Kim et al. 2019, Merrill et al. 2019. However, all of these studies involve models that are trained on large quantities of text which may not be impoverished in domains that these benchmarks assess. As a result, it is unclear whether any apparent hierarchical behavior reported in these works is the effect of a bias for hierarchical generalization or the accumulation of patterns explicitly guided by the training data. McCoy, Frank & Linzen (2020) take a different tack: the training data is carefully controlled so that hierarchical behavior can emerge only if a model itself is biased to extract hierarchical generalizations. Their experiments demonstrate that recurrent neural network seq2seq models show a clear preference for linear generalization.

The recently developed Transformer architecture has led to revolutionary advances across many areas of natural language processing, including machine translation and question answering Vaswani et al. 2017, Devlin et al. 2019. Transformer-based models have also shown considerable success on benchmarks that appear to require the representation of hierarchical abstractions Rogers, Kovaleva & Rumshisky 2020, Goldberg 2019, Warstadt et al. 2019. Further, investigations of Transformers' representations of sentences Hewitt & Manning 2019, Lin, Tan & Frank 2019 point to encodings of hierarchical syntactic structure. Yet, for the reasons noted above, it is difficult to conclude much about the inductive bias in the Transformer: they are trained on vast datasets, leaving open the question of the impact of inductive bias as opposed to training data (Warstadt & Bowman (2020), but see van Schijndel, Mueller & Linzen (2019) for arguments that even massive data may not be sufficient). This paper contributes to our understanding by examining the degree to which the Transformer architecture is biased toward hierarchical generalization when the data underdetermine such generalization. Specifically, we study whether Transformers learning sequence-to-sequence mappings generalize in a structure sensitive way, and compare their performance with recurrent models.

4.3 Experiments

Our experiments involve a variety of English-language transduction tasks that highlight hierarchically-governed patterns. For each task, the training data is ambiguous between a linear and hierarchical generalization. This allows us to evaluate performance on both a TEST set, drawn from the same distribution as the training set, and a GEN set of data, that contains out-of-distribution data consistent only with hierarchical patterns of generalization.

We compare transformer models with a number of recurrent architectures (LSTMs and GRUs with no attention, with additive attention Bahdanau, Cho & Bengio 2016, and with multiplicative attention Luong, Pham & Manning 2015). Transformer models follow their usual implementation with self- and multi-headed attention. For each model type, we perform 10 runs, initialized with different random initial seeds, and report median accuracy metrics. Recurrent units are single-layer models, with hidden and embedding dimensions of 256. Transformers are 4-headed, 3-layer models with hidden and embedding dimensions of 128. All models are trained at a learning rate of 0.01 using SGD optimization for 100 epochs with early stopping.

Polar Question Formation

Our first task involves the process of question formation discussed earlier. We borrow the formulation of this task from McCoy, Frank & Linzen (2020): the training dataset consists of an input sentence (a simple declarative with relative clauses optionally modifying the subject and object), a transformation token, `decl` or `quest`, and an output sentence. The transformation token specifies what the form of the target output should be. Following the logic surrounding example (25), examples with subject-modifying relative clauses are never paired in the training data with the `quest` transformation token. As a result, the network is not trained on sentences in which an auxiliary verb must be fronted past an intervening relative clause, and the target generalization is therefore ambiguous between something akin to MOVE-MAIN and MOVE-FIRST. While a network that acquires the MOVE-FIRST generalization will succeed on the in-distribution TEST set consisting of examples of the same structure as in the training data, it will fail on the GEN set consisting of input sentences with subject-relative clauses and the `quest` transformation.

All trained network types performed well on the in-distribution TEST set, attaining mean full-sentence accuracies of at least 95%. In contrast, none of the models succeeded on the GEN set in full sentence accuracy. Following McCoy, Frank & Linzen (2020), we instead assess GEN set performance using the more lenient metric of first-word accuracy. Since the GEN set includes only sentences with distinct auxiliary verbs in the main and relative clauses, the identity of the first output word reveals whether the network has acquired a linear (MOVE-FIRST) or hierarchical (MOVE-MAIN) generalization. Results are shown in Figure 4.1. As noted in McCoy, Frank & Linzen (2020), there is

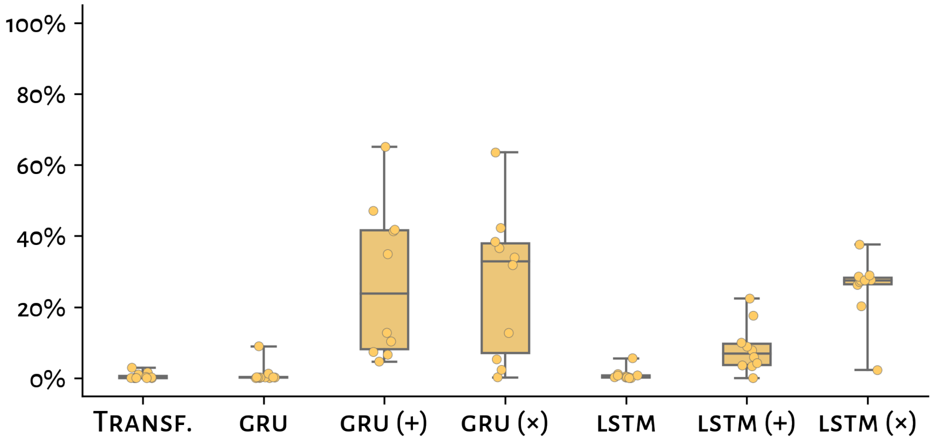

Figure 4.1: Proportion of first-word predictions consistent with hierarchical generalization in the question GEN set. A (+) denotes additive attention, (×), multiplicative. Horizontal bars denote max, median, and min values.

variation in performance among the different types of recurrent networks: GRUs with multiplicative attention achieved median accuracy of 32.9%. Transformers exhibit the worst median performance among all architectures surveyed, with a median first-word accuracy of just 0.03% and virtually no variability across different random initializations. Instead, Transformer models overwhelmingly predicted sequences consistent with a linear MOVE-FIRST rule on the GEN set. These results are robust across changes in learning rate.

Tense Reinflection

Our second mapping task, again borrowed from McCoy, Frank & Linzen (2020) involves the reinflection of a sentence with a past tense verb into one with either a past or present tense verb. Significantly, the English present tense involves structurally-conditioned agreement with the verb's subject. In complex expressions like (26a), distractor nouns with different number within the subject linearly separate the verb from the subject, but the grammatical agreement is nonetheless governed by a hierarchical AGREE-SUBJECT relation (predicting (26b)). as opposed to an AGREE-RECENT relation (predicting (26c)).

(26) a. My newt near the elephants ran.
b. My **newt** near the elephants **runs**.
c. * My newt near the **elephants run**.

Our datasets consist of past-tense English sentences as inputs, optionally with prepositional phrases or relative clauses modifying the subject or object, along with PRES and PAST transformation tokens that indicate the form of the target output. For training and in-distribution test data, examples with the PRES token do not have modified subjects, so that the reinflection mapping is ambiguous between AGREE-SUBJECT and AGREE-RECENT. In contrast, the GEN set includes sentences where the two rules make different predictions (modified subjects with distractor having distinct number). Results are shown in Figure 4.2. Like the recurrent architectures, Transformers systematically fail to exhibit hierarchical in favor of linear generalization.

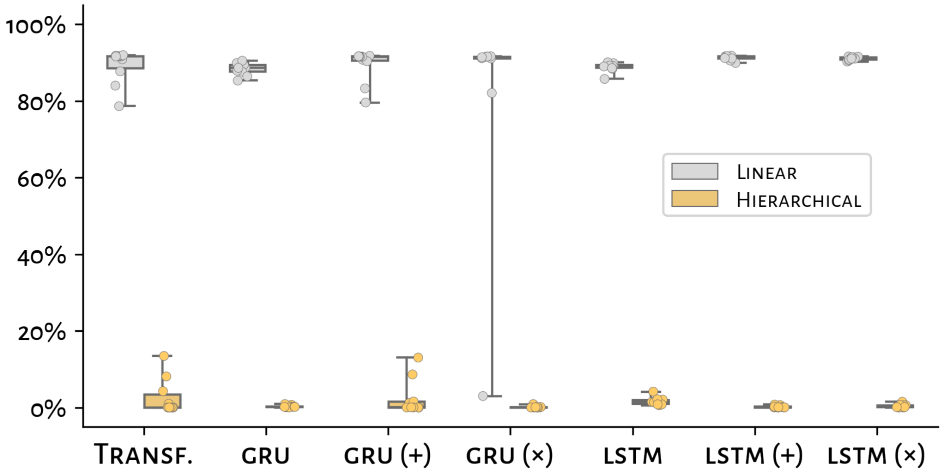

Figure 4.2: Proportion of linear and hierarchical predictions on the reinflection GEN set.

Negation

Our third task involves the conversion of an affirmative sentence into a negative one. Negation requires the insertion of the negative marker "not" immediately prior to the main verb.

(27) a. The bird will sing.
b. The bird will **not** sing.

When an adverbial clause is placed before or after the main clause (28), the main verb is no longer consistently the linearly first or last verb in the sentence.

(28) a. The bird will sing because the cat will swim.
b. The bird will **not** sing because the cat will swim.
c. Because the cat will swim the bird will **not** sing.

Our dataset consists of affirmative sentences, with adverbial clauses optionally preceding or following the main clause. These are transformed either into (identical) affirmatives or corresponding negatives. The training and in-distribution test set excludes sentences with initial adverbial clauses that must be mapped to negatives. As a result, this data set is ambiguous between a linear NEG-FIRST generalization and a hierarchical NEG-MAIN. This ambiguity is resolved in the GEN set, which contains sentences with preceding adverbials that must be converted into negative sentences, following the NEG-MAIN generalization.

All models, including the Transformer, perform exceedingly well on in-distribution data, attaining near-ceiling full-sentence accuracy on the TEST set. By contrast, all models, again including the Transformer, fail uniformly on the GEN set, attaining near-zero performance even using a more forgiving metric looking only at correct placement of the negative marker. Closer examination of the model outputs on the GEN set reveals that networks of all sorts overwhelmingly produce predictions consistent with the linear generalization (NEG-FIRST).

Reflexive Anaphoric Interpretation

Our final task, similar to that of Kim & Linzen (2020) and Frank & Petty (2020), involves the semantic parsing of a sequence into a predicate calculus representation, as in (29).

(29) Alice sees Bob → SEE(ALICE, BOB)

For entities whose meaning is context-independent, like nouns or verbs, this task involves learning a combination of token correspondence and form composition. As Frank & Petty (2020) note, reflexive anaphora like "herself" present a challenge since their meaning is not context-independent but rather conditioned on a linguistically-determined antecedent. In sentences with complex subjects, like that in (30) with a prepositional phrase modifier, the identification of the correct antecedent for the anaphor is conditioned not by the linear distance between a potential antecedent and the reflexive but rather by the hierarchical relation between the antecedent and reflexive.

(30) The boy by the king sees himself → SEE(BOY, BOY) ∧ BY(BOY, KING)

Our in-distribution data consists of sentences, transitive and intransitive, paired with predicate calculus representations of their meanings. Input sentences in this set may have complex subjects or the reflexive objects ("himself" or "herself"), but not both. As a result, the training and TEST data does not disambiguate whether the reflexive is co-referent with the grammatical subject or the noun phrase immediately preceding the verb. The GEN set contains only sentences reflexive objects and complex subjects containing prepositional phrases, and therefore serves to distinguish between the linear and hierarchical generalizations.

All models examined perform well on the TEST set, attaining median full sequence accuracy of 100%. Results on the GEN set, as shown in Figure 4.3, are more varied.

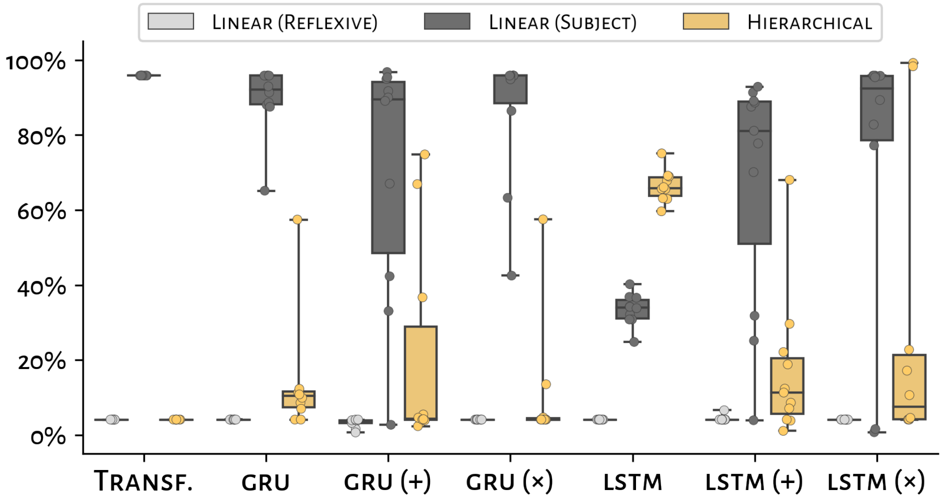

Figure 4.3: Proportion of reflexive-linear, subject-linear, and hierarchical predictions in the anaphora GEN set.

We categorize the predictions made by the network into three distinct classes: subject-verb linear, where the model interprets the subject of the verb as being the linearly most recent noun (incompatible with the training data); reflexive linear, where the model interprets the antecedent of the reflexive as being the linearly most recent noun (compatible with the training set); and hierarchical, where the model correctly interprets both the subject and antecedent in a manner consistent with the hierarchical structure of the sentence (also compatible with training). Transformers and GRU models overwhelming make predictions consistent with reflexive linearity. LSTMs are more varied, with inattentive LSTMs attaining the highest hierarchical scores of all network types with a median performance of 65.8%.

4.4 Conclusion

Transformers have shown great success on syntactic benchmarks. Is this because the architecture has useful syntactic biases, or is it because cues to hierarchical structure are present in their training data? Our results find no evidence for the former, suggesting that their syntactic successes can mainly be attributed to their ability to leverage massive training sets rather than linguistically-relevant architectural biases. Though the Transformer models studied here were the best performers on in-distribution data across all tasks, their strong preference for linear over hierarchical generalization suggests an explanation for their poor performance on tasks requiring structural generalization (Kim & Linzen 2020) despite their promise in other syntactically sensitive tasks. Finally, we note that the preference we have observed for linear generalization is consistent with previous theoretical work on the (limited) expressive power of Transformers Hahn 2020, Merrill 2019.

Chapter 5

Alice's Adventures in Reflexiveland

5.1 Introduction

Chapter 3 demonstrates that inattentive recurrent neural networks are capable of solving the problem of reflexive anaphora resolution to a high degree of generality on limited training support. In particular, experiments **Alice-1** and **Alice-2** show, in part, that inattentive GRUs and SRNs are capable of learning to generalize knowledge of a reflexive anaphor like *herself* to an antecedent *Alice* which has been withheld from syntactically reflexive (**Alice-1**) and semantically reflexive (**Alice-2**) contexts during training when the problem is presented in a sequence-to-sequence context. As shown in Chapter 2, reflexive anaphora resolution is equivalent to an ℓ_3 level of algebraic generalization on the proposed typology of generalization classes, and is hence a strictly stronger problem than something akin to G. F. Marcus (1998a)'s original identity generalization task (ℓ_2), or the lexical generalization results obtained in Kim & Smolensky (2021) (ℓ_2, and see Chapter 6 for additional such results). Our constructive results here stand in contrast to those of Frank, Mathis & Badecker (2013), where treatment of the same anaphora resolution task in a language modelling context found that such inattentive recurrent models were unable to achieve lexical generalization to withheld antecedents.

These positive results then demonstrate that the theoretical incapability of recurrent networks to solve such ℓ_2- and ℓ_2-class problems is not borne out. Yet the discrepancy between the results presented in the previous chapter and the earlier negative results of G. F. Marcus (1998a) and Frank, Mathis & Badecker (2013) raise important questions: how are models able to solve such generalization problems in the absence of explicit inductive biases, like attention mechanism, to do so? To what extent is the sequence-to-sequence design context an inductive bias for such generalization tasks?

To move towards an answer to these broad questions, in this chapter we consider two methods for attempting to elucidate the mechanisms inattentive SRN and GRU models employ to learn a generalized rule for reflexive anaphora resolution. We begin by trying to determine where in these models the resolution of anaphora actually occurs, drawing inspiration from well-known observations of word-embedding models to test if our models learn analogical representations of input sequences and using this to test how models treat *herself* in various conditions. We then relax this experiment to try to characterize the embedding space of the models to understand how the decoder interprets vectors in the hidden embedding space. Finally, we draw connections to recent work done in modelling embedding spaces using explicit hypothesis for input-embedding representations and theorize about ways this can be used to provide a more explicit characterization of how models learn to represent contextually-ambiguous inputs in a generalizable way.

5.2 Analogical Arithmetic

Our first attempt at understanding how and when these networks solve the task of reflexive anaphora resolution seeks to clarify where within the model this resolution takes place. Since a trained models is able to interpret an input sequence with an arbitrary antecedent correctly, it must at someone resolve the representation of an input sequence like *Alice sees herself* such that the decoder will ultimately produce the correct output sequence 'SEE(ALICE, ALICE)'. There are two possibilities for where the model achieves this resolution: either the model resolves *herself* to the correct antecedent in the encoder, or it does so in the decoder.

To elucidate where inside these models the resolution takes place, we take inspiration from the observed properties of word embedding models, like Word2Vec (Mikolov et al. 2013). These models compute continuous vector representations of words drawn from large corpora, optimizing for distance in encoding space as a measure of semantic similarity under the intuition that the encodings of words which are semantically similar to one another ought to be represented as vectors which are relatively near compared to the encodings of words which are semantically distant. In addition to achieving good performance on tasks measuring syntactic and semantic word similarity (Mikolov et al. 2013), word embeddings display properties of linearity which correspond nicely to naïve intuitions about the semantic distance between interpretable sentences (Allen & Hospedales 2019).

Concretely, word embedding models appear to encode information akin to analogical semantics even though they are not explicitly trained to do so. For instance, consider the following word-level analogy:

(31) *man* is to *king* as *woman* is to ______

Providing an answer to this analogy (say, *queen*) involves making a semantic judgement about the relationship between the first pair of words and then applying that same relationship to the different third word to produce a fourth. Word embeddings optimized for semantic similarity appear to latently represent this semantic relationship as vector displacement; thus, given an analogical relationship like (32a), it is often the case that a word embedding model will display the property shown in (32b), where $\boldsymbol{w}_i$ is the embedding of w_i.

(32) a. w_a is to $w_{a'}$ as w_b is to $w_{b'}$
b. $\boldsymbol{w}_b + (\boldsymbol{w}_{a'} - \boldsymbol{w}_a) \approx \boldsymbol{w}_{b'}$

In the context of word embedding models, this larger, structural pattern is surprising given that metrics for word similarity are only computed using local distributional data from training corpora (Allen & Hospedales 2019). However, given that the embedding model's task is to build functional representations of lexical data in a vector space, it is natural to view the encoder component of a sequence-to-sequence model as an extension of these simpler skip-gram models. Indeed, encoders contain an learned word-embedding layer prior to the recurrent layers which operate on the vectors produced by this learned embedding (Sutskever, Vinyals & Le 2014). If we think of the encoder as an embedding model which takes in a sequence of input tokens and produces a high-dimensional representation of this sequence on the joint basis of input and output distribution, we might wonder if the encoding vectors which are passed to the decoder represent the semantic content of the input sequences in an analogously interpretable fashion, where displacement between two encodings corresponds to the semantic difference between the two respective input sequences.

If this were the case, how could we tell whether or not the encoding of an input like *Alice sees herself* had already resolved the anaphor *herself* → ALICE? A naïve first guess might be to simply compare the encoded representations of a true reflexive sentence like *Alice sees herself* with that of its corresponding pseudoreflexive sentence *Alice sees Alice*; since both of these inputs will ultimately be decoded as the same sequence, it would be reasonable to hope that a model which resolves reflexive anaphora in the encoder might simply map both sequences to vectors so functionally close by to one another in encoding space that the difference between the two was marginal. That is, it could be as in (33) that the displacement between the two encodings is approximately the zero vector in the encoding space, where "approximate" means that the magnitude is small enough that it is treated as irrelevant by the decoder (i.e., we could perturb any valid encoding of an input sequence by this displacement vector and it would not affect the output decoding).

(33) $\mathrm{Enc}(\textit{Alice sees Alice}) - \mathrm{Enc}(\textit{Alice sees herself}) \approx 0$

While an invariantly-null[1] displacement vector would imply that a model resolves anaphora in the encoder, it is not the case that encoder resolution entails an invariantly-null displacement vector; it may just as well be the case that the encoder simply redundantly encodes the identity of reflexive anaphora separately from the interpretations of pseudoreflexive sentences, where $\mathrm{Enc}(\textit{Alice sees herself})$ and $\mathrm{Enc}(\textit{Alice sees Alice})$ both separately decode to 'SEE(ALICE, ALICE)' but they are not encoded as each other. In this scenario, the displacement vector between the encodings would act as a kind of semantic displacement capturing the distinction between the 'reflexive' and 'pseudoreflexive' meanings of each input sequence.

On the other hand, it might be the case that the encoder merely encodes *herself* as its own token and defers responsibility for resolving the input on to the decoder. To distinguish between these two possibilities, we can turn back to the analogical displacement model to provide a possible test for encoder resolution. If we assume that the encoder does represent analogical semantics in the same way that word embeddings are observed to do, consider the following analogical setup:

(34) *Alice sees Mary* is to *John sees Mary* as *Alice sees Bob* is to *John sees Bob*

Here the analogical relationship represented between the two pairs of sentences is a change-of-subject; the corresponding displacement vector

(35) $\Delta_{\textit{John,Alice}} = \mathrm{Enc}(\textit{John sees Mary}) - \mathrm{Enc}(\textit{Alice sees Mary})$

would encode the difference between *John* and *Alice* as the sentential subject. Performing this kind of analogical arithmetic is then a kind of encoding-space subject replacement, where we use this displacement vector to swap out the subject of a different sentence, like in (36) below.

(36) $\mathrm{Enc}(\textit{John sees Bob}) + \Delta_{\textit{John,Alice}} \approx \mathrm{Enc}(\textit{Alice sees Bob})$

Consider now the effect of performing this subject replacement on reflexive sentences: let $\Delta_{A \to M}$ be the displacement vector between the encodings of *Alice knows Bob* and *Mary knows Bob*, and consider the sum below.

(37) $\mathrm{Enc}(\textit{Alice knows herself}) + \Delta_{\textit{Alice,Mary}}$

If the pattern of (36) holds, (37) should have the encoding representation of *Alice knows herself* with the subject *Alice* swapped for *Mary*. But, how should the model interpret *herself* now that the subject has been replaced?

If the result of this arithmetic is interpretable, there are two sensible outcomes: if the encoder is responsible for resolving anaphoric identity, then the subject replacement

[1] *Invariantly-null* meaning that the displacement vector is approximately zero for all reflexive-pseudoreflexive sentence pairs in the input domain.

will have no effect on the identity of the object, as in (38a); or else, if the encoder does not resolve anaphoric identity then this sum passed to the decoder will be interpreted as *herself* in the context of *Mary*, as in (38b).

(38) a. Enc(*Alice knows herself*) + $\Delta_{Alice,Mary}$ → KNOW(MARY, ALICE) [Encoder resolution]
b. Enc(*Alice knows herself*) + $\Delta_{Alice,Mary}$ → KNOW(MARY, MARY) [Decoder resolution]

This distinction recalls a property of the semantic interpretation of reflexive anaphora observed in natural language, known as the 'strict-sloppy' reading distinction. Consider a case of simple VP ellipses like that of (39) below.

(39) a. John goes to see Mary, and Bill does too
b. → Bill goes to see Mary

Here, the elision of the second VP does not impact the meaning of the conjoined phrase, since the content of the first predicate is copied over to the second. For simple predicates such as that in (39), this interpretation is trivial. But consider a case where the predicate contains a reflexive anaphor, as in (40) below.

(40) John likes himself, and Bill does too

Here, the elision of the second VP creates an ambiguity in the interpretation of the second phrase: does *himself* retain the interpretation it has in its original position (i.e., JOHN) or does it get re-resolved in its semantically-copied position (i.e., BILL)? These two possibilities, shown in (41a) below, are known as the 'strict' and 'sloppy' readings of anaphora, respectively.

(41) a. John likes himself, and Bill does too [like himself]
b. → Bill likes John [strict identity]
c. → Bill likes Bill [sloppy identity]

Note that reflexive anaphora are usually taken to only have a sloppy interpretation; that is, an expression like (41a) will only be interpreted by human speakers as (41c).

In our contrived example of subject-replacement on reflexive sentences, we can draw a parallel between encoder resolution and strict identity, where the reflexive anaphor retains the interpretation it originally had; and between decoder resolution and sloppy identity, where the interpretation changes as the anaphor is reinterpreted in its new context.

A Note on Sensibility and Interpretability

While the arithmetic trick described above presents a clever way to distinguish between encoder- and decoder-resolution of reflexive anaphora, its validity only holds under some very strong and potentially unfounded assumptions. The linearity properties of word embedding models are surprising, given that the models do not optimize for representing this kind of higher-order structure. We have no *a priori* reason to assume that the encoders of our trained models should behave in this way, and it is entirely possible that the results of this arithmetic surgery are uninterpretable: the decoder may produce expressions which are malformed, or which do not have sensible connections to the initial input sequences involved. Furthermore, the decoder may not treat such sums uniformly, instead producing different kinds of results depending on the inputs.

With this note of caution heeded, it is still worth exploring how our models perform when given the task of decoding a hidden representation which has been selectively modified in such a way as to attempt to force the encoder into revealing its tricks. If the models do show a categorical preference for one method of resolution over another, as tested by this analogical arithmetic, then we will know not only where resolution happens in the model but also that the encoding representations display the same kind of unexpected linearity as is found in word embedding models. If models exhibit probabilistic, but not categorical preference for one method of resolution over another, we likewise learn that models exhibit variability in their methods of resolution. And finally, if this arithmetic task is founded on untenable assumptions about the properties of the encoding space, we learn that the encoding representations do not display the same properties as word embedding models. Furthermore, by analyzing the ways in which the attempted decoding of arithmetically-modified encoding representations fails, we may learn about how the decoder treats systematically malformed inputs.

Experimental Setup

We test inattentive SRN and GRU models trained on the **Alice-2** experiment described in Section 3.4, where both reflexive (42a) and pseudoreflexive (42b) sentences involving *Alice* as an antecedent are withheld from models during training. Models are then trained on a training set containing reflexive, non-reflexive, and pseudoreflexive sentences-interpretation pairs. As noted in Section 3.4, both SRN and GRU models perform near-ceiling on the test and generalization sets, indicating that both model variants have acquired robust algebraic generalization over the input space.

(42) a. *Alice sees herself* → SEE(ALICE, ALICE)
b. *Alice sees Alice* → SEE(ALICE, ALICE) [withheld from **Alice-2**]

We choose to analyze these models for several reasons: first, **Alice-2** is a strictly harder task than **Alice-1**, since models do not receive any training support to suggest that an input sequence will ever map to an output containing ALICE in both subject and object position. Despite this additional difficulty, both inattentive SRNs and GRUs are able to solve this task to near-perfect accuracy on the generalization set. Since **Alice-2** is more difficult, it sands to reason that the generalization learned by models in **Alice-2** is more robust than that learned in **Alice-1**, and therefore represents a better subject of inquiry if we wish to understand how simple models are able to learn algebraically general rules.

Second, the performance of inattentive SRN models begins to degrade in the subsequent, harder experiments (**Alice-3**, **-4**, and **-5**), meaning that analysis conducted on models from those experiments will not yield any insights into how SRN models successfully exhibit algebraic generalization.

We choose to focus on SRNs and GRUs, to the exclusion of the inattentive LSTMs and the attentive SRNs, GRUs, LSTMs, and transformer models additionally surveyed in Chapter 3 and Chapter 4 for two reasons. First, the positive results obtained in these chapters are most surprising for the computationally simpler models. Inattentive models lack any of the benefit conferred by the inductive bias of attention, which as previously noted in Chapter 2 models the exact phenomenon of permutation equivariance needed to solve algebraically general problems; SRNs and GRUs are also simpler than LSTMs, meaning they lack as much computational expressiveness (Merrill et al. 2020). Since we care most about how these unexpectedly-performant models manage to solve algebraically general tasks, it makes sense to focus our analysis on these simpler models.

Second, the limited computational capacity of inattentive SRNs and GRUs is reflected in their relatively simple implementation when compared to LSTMs (which make use of an extra computational cell passed between encoding and decoding steps), attentive versions of the three recurrent architectures used, and transformers. Since mucking about with the internal representations of the encoder is not a task which is provided out of the box for PyTorch models, limiting our current analysis to just inattentive SRNs and GRUs results in a more manageable and verifiable codebase.

We delineate three different ways of using the analogical arithmetic described in the previous subsection to probe the behavior of the models' encoders: *subject replacement*, *verb replacement*, and *object replacement*. In each case, we attempt to permute tokens of the relevant grammatical category using displacement vectors as described previously.

For clarity of reference, we will introduce the following terminology to refer to components of an analogical arithmetic expression.

(43) a. **α, β-Displacement Vector:** Let α and β be elements of a particular syntactic category (here, either SUBJECT, VERB, or OBJECT). The displacement vector $\Delta_{\alpha,\beta}$ is the vector formed by subtracting expression σ_2 from σ_1 where $\alpha \in \sigma_1$ and $\beta \in \sigma_2$ and σ_1 is identical to σ_2 for all tokens excepting α, β. Note that for any given α, β there are actually many different displacement vectors $\{\Delta_{\alpha,\beta}\}$ corresponding to different choices of σ_1 and σ_2. When referring $\Delta_{\alpha,\beta}$, know that it refers to an arbitrary member of $\{\Delta_{\alpha,\beta}\}$.

b. **Operand:** Let $\Delta_{\alpha,\beta}$ be an α, β-displacement vector and let σ be an input sequence containing α. In the expression

$$\mathrm{Enc}(\sigma) + \Delta_{\alpha,\beta},$$

we refer to $\mathrm{Enc}(\sigma)$ as the operand—the encoding or expression being operated on by $\Delta_{\alpha,\beta}$.

In addition do distinguishing between different kinds of analogical arithmetic expressions based on the category of the token being replaced, we can further refine a classification of these expressions based on the properties of the operand in the expression. Since reflexive, pseudoreflexive, and non-reflexive input sequences have thus far proven useful analytical classes of input sentences, we believe it is worthwhile to distinguish CATEGORY-replacement with reflexive, pseudoreflexive, and non-reflexive operands. To provide a concrete example consider the typology of subject replacement shown below in (44), where $\Delta_{A,n}$ is a subject displacement vector between ALICE and an arbitrary name *n*:

(44) **Subject Replacement Typology**

a. $\mathrm{Enc}($*Alice sees herself*$) + \Delta_{A,n}$ [Reflexive (Refl.) operand]
b. $\mathrm{Enc}($*Alice sees Alice*$) + \Delta_{A,n}$ [Pseudoreflexive (PR) operand]
c. $\mathrm{Enc}($*Alice sees Bob*$) + \Delta_{A,n}$ [Non-reflexive (NR) operand]

For each combination of MODEL ARCHITECTURE, CATEGORY, and OPERAND we take a model of the requisite type and generate $10,000$ random arithmetic expressions of the appropriate CATEGORY-OPERAND type, perform the arithmetic surgery on the operand, and decode the resulting sum as if it were a normal input's encoding. We evaluate the decoded sequence on a variety of accuracy metrics, where each metric is reported as an average over the $10,000$ predictions. The following four metrics are reported for all CATEGORY-replacements:

(45) a. **Sentence Accuracy:** Measures full sequence accuracy of the output, where model scores 1 on an input if the output matches the target, or else 0 if there is any discrepancy. Note that for reflexive operands undergoing subject replacement, we count outputs as correct if they are consistent with encoder resolution of anaphora (where the predicted object is identical to the operand's original subject) or with decoder resolution (where the subject and object are both equal to the new subject introduced by the subject-replacement displacement vector).

b. **Subject Accuracy:** A model scores 1 if the tokens in the subject position of the output and target agree, or else 0.

c. **Verb Accuracy:** A model scores 1 if the tokens in the subject position of the output and target agree, or else 0.

d. **Object Accuracy:** A model scores 1 if the tokens in the object position of the output and target agree, or else 0.

Additionally, we report three metrics which measure the degree to which models predict that a given results of arithmetic surgery is reflexive.

(46) a. **Predicted Reflexive:** The proportion of the outputs which are reflexive relative to the total number of predictions. Note that for NR and PR operands, a prediction is only counted as 'reflexive' if the predicted subject and object are identical, while for reflexive operands in subject and verb replacement a prediction is also counted as 'reflexive' if the predicted object is identical to the original operand's subject. This corresponds to the case where the resolution of *herself* happens in the encoder, and so in such cases the output is still a valid interpretation of a reflexive sentence. For reflexive operands undergoing object replacement, we again only count a prediction as 'reflexive' if the subject and object are identical since object replacement of a reflexive operand should necessarily result in a distinct subject and object, and so there are no felicitous results which could have a reflexive interpretation but decode to an output with distinct subject and object.

b. **(dec):** Reports the percentage of the reflexive outputs which show resolution consistent with decoder resolution, where the subject and object are identical. Note that this metric is only interpretable if the result of the arithmetic surgery should still have a representation of a reflexive anaphor (i.e., subject replacement on reflexive operands), so we omit this metric when in all other cases.

c. **(enc):** Reports the percentage of the reflexive outputs which show resolution consistent with encoder resolution, where the object of the output is identical to the original subject of the operand. Note that this metric is only interpretable if the result of the arithmetic surgery should still have a representation of a reflexive anaphor (i.e., subject replacement on reflexive operands), so we omit this metric when in all other cases.

Finally, note that in each table of results for CATEGORY-replacement, the CATEGORY Accuracy measure has been highlighted in grey. This is to distinguish between the accuracy of the model on the category which should have undergone replacement and the accuracy of the model on categories which may have undergone spurious modification as a result of the analogical arithmetic.

Subject Replacement Results

We conduct subject replacement surgery on NR, PR, and reflexive operands, as typified below in (47).

(47) **Subject Replacement**

a. Enc(*Alice sees Bob*) + $\Delta_{Alice,Mary}$ ≈ Enc(*Mary sees Bob*) [NR]

b. Enc(*Alice sees Alice*) + $\Delta_{Alice,Mary}$ ≈ Enc(*Mary sees Alice*) [PR]

c. Enc(*Alice sees herself*) + $\Delta_{Alice,Mary}$ ≈ Enc(*Mary sees herself*) [Refl.]

Table 5.1 shows the results of each model type decoding the results of this analogical arithmetic. *Prima facie*, the most notable figure reported here is the fact that GRU models display a categorical failure to permit subject replacement when the operand is not lexically reflexive (e.g., *herself*); on NR and PR operands, GRU models uniformly decode the results of the subject replacement incorrectly as reflexive outputs. On NR operands, models are strongly biases towards overgeneralizing the object, replacing the true subject with the original object by a two-to-one margin. On PR operands, this trend is reversed, where models categorically overgeneralize the subject, replacing the actual object with the new subject. For reflexive operands, GRU models score quite well, performing at ceiling and displaying a clear categorical preference for interpreting the results of subject replacement in a manner consistent with decoder-resolution of the reflexive pronouns.

	SRN			GRU		
	NR	PR	Refl	NR	PR	Refl
Sent Acc	0.622	0.646	0.317	0.001	0.000	0.993
Subj Acc	0.787	0.806	0.716	0.333	0.843	0.933
Verb Acc	1.000	0.998	0.994	1.000	1.000	1.000
Obj Acc	0.771	0.779	0.339	0.662	0.152	0.003
Pred Refl	0.016	0.002	0.317	0.997	0.999	1.000
(dec)	—	—	0.155	—	—	1.000
(enc)	—	—	0.845	—	—	0.000

Table 5.1: Subject-replacement accuracies by MODEL and OPERAND

In comparison to this all-or-nothing reflexive performance of the GRUs, SRN models display more middling, less categorical performance. While SRNs do not spuriously produce reflexive outputs from NR or PR operands, they permit subject replacement only about 63% of the time. Accuracy on the individual objects or replace subjects is higher (around 78% in each case), suggesting that the instances of failure are uncorrelated and the lower performance on the full sentence accuracy is simply a result of these errors not overlapping. On reflexive predicates, though, SRN performance is quite a bit worse: models only generate reflexive outputs in 32% of cases, with the majority of the error arising from incorrect predictions for the object (while subject accuracy dips only slightly to 72%, object accuracy drops strongly to only 34%). This suggests that SRNs' encoding of reflexive objects is somehow less separable than that of non-reflexive objects, permitting subject replacement at a far lower rate in reflexive operands. Of the reflexive operands which are decoded in a manner consistent with the expected results of subject replacement, SRNs display a clear preference for decodings consistent with encoder resolutions of reflexive anaphora, though this preference isn't quite as categorical as that of GRUs.

Both SRN and GRU models demonstrate near-perfect verb accuracy, suggesting that for both models the encoded representation of the verb is separable enough not to be impacted by the subject replacement operation.

Verb Replacement Results

We conduct verb-replacement surgery on NR, PR, and reflexive operands, as typified below in (48).

(48) **Verb Replacement**

a. $\mathrm{Enc}(\textit{Alice sees Bob}) + \Delta_{\textit{sees,knows}} \approx \mathrm{Enc}(\textit{Alice knows Bob})$ [NR]

b. $\mathrm{Enc}(\textit{Alice sees Alice}) + \Delta_{\textit{sees,knows}} \approx \mathrm{Enc}(\textit{Alice knows Alice})$ [PR]

c. $\mathrm{Enc}(\textit{Alice sees herself}) + \Delta_{\textit{sees,knows}} \approx \mathrm{Enc}(\textit{Alice knows herself})$ [Refl.]

Table 5.2 reports how the various models decoded the results of this arithmetic. As is the case in subject replacement, both SRN and GRU models display high levels of verb accuracy, indicating that for both architectures the representation of the verb in the embedding space is targetable by replacement in a robust manner.

GRU models again display an overall categorical failure on non-reflexive operands, again owing to errors in the subject or object decoding due to models spuriously producing reflexive outputs. By a roughly two-to-one margin, GRU models on NR operands favor overgeneralizing the object to the subject. On PR operands, however (in contrast to their performance on subject replacement), and on reflexive operands the GRU models exhibit at-ceiling performance. This suggests that GRU models are representing the identity of the subject and object of an input in a linked way, and that this representation is mildly sensitive to displacement by the verb representation. Thus in verb replacement, when the subject and object are distinct, verb replacement causes the model to interpret the result as reflexive, with a bias towards preserving the identity of the object. For PR and reflexive predicates, since the result is still reflexive, this displacement has little effect. In cases of subject replacement, however, the model preserves the identity of the new subject at the expense of the original object, resulting in total failure on both NR and PR operands but success on reflexive ones.

Compared to GRUs, SRNs exhibit more stable, although worse, performance across operand type. Models achieve around 75% full sentence accuracy on NR operands,

	SRN			GRU		
	NR	PR	Refl	NR	PR	Refl
Sent Acc	0.748	0.829	0.644	0.001	0.996	1.000
Subj Acc	0.871	0.905	0.861	0.315	1.000	1.000
Verb Acc	0.999	0.999	0.996	0.999	0.998	1.000
Obj Acc	0.854	0.891	0.707	0.637	0.999	1.000
Pred Refl	0.005	0.859	0.646	0.997	0.999	1.000

Table 5.2: Verb replacement accuracies by MODEL and OPERAND

raising to 83% accuracy on PR operands and falling to 64% on reflexive operands. Notably, as with GRUs, SRNs permit felicitous verb replacement on all types of operands, scoring at-ceiling on Verb Accuracy for NR, PR, and reflexive operands. Failure, then, comes from spurious changes to the subject and object tokens as a result from verb replacement. For all operand types, we observe that models are better at preserving subject identity than object identity, although the difference is not strongly pronounced. While GRUs display complementary failures at preserving subject and object identity, SRNs exhibit no such correlation between subject and object accuracies. Rather, changes in overall sentence accuracy are attributable to independent changes in the underlying subject and object accuracies. Notably, SRN models to not display a tendency to over-generate reflexive forms when none are required

Object Replacement Accuracy

We conduct object replacement on NR, PR, and reflexive operands as typified below in (49).

(49) **Object Replacement**

a. $\text{Enc}(\textit{Alice sees Bob}) + \Delta_{Bob,Mary} \approx \text{Enc}(\textit{Alice sees Mary})$ [NR]

b. $\text{Enc}(\textit{Alice sees Alice}) + \Delta_{Alice,Mary} \approx \text{Enc}(\textit{Alice sees Mary})$ [PR]

c. $\text{Enc}(\textit{Alice sees herself}) + \Delta_{Alice,Mary} \approx \text{Enc}(\textit{Alice sees Mary})$ [Refl.]

Note that object replacement of a reflexive operand requires a bit more interpretation than that of an NR or PR operand, since in a reflexive case we are replacing an object whose lexical (input) identity is different from its decoded identity. To generate displacement vectors for these sentences, we take as the original object the subject of the operand, so in (49c) the reflexive *herself* is given the displacement vector $\Delta_{Alice,Mary}$ since *herself* would have been decoded as MARY in the original expression.

Table 5.3 shows the results of these replacements passed to the various models' decoders. GRU models display an across-the-board failure causes entirely by their predilection to generate reflexive outputs. The results of the object replacement arithmetic presented to the model are never interpretable as reflexive, since we explicitly replace the objects of reflexive and pseudo-reflexive operands with distinct objects, but GRU models still generate reflexive outputs with a strong bias towards copying the new object into the subject position.

	SRN			GRU		
	NR	PR	Refl	NR	PR	Refl
Sent Acc	0.630	0.670	0.356	0.001	0.000	0.000
Subj Acc	0.847	0.888	0.719	0.335	0.046	0.001
Verb Acc	0.999	0.996	0.991	1.000	1.000	1.000
Obj Acc	0.738	0.737	0.474	0.620	0.950	0.997
Pred Refl	0.019	0.001	0.362	0.996	0.999	1.000

Table 5.3: Object replacement accuracies by MODEL and OPERAND

SRN behaviour is very reminiscent of that on subject replacement, with okay performance on NR and PR operands caused by non-correlated but small errors in the subject and object accuracies, and with poor performance on reflexive operands caused mostly by errors in object accuracy. Both SRN and GRU models again display high levels of verb accuracy, suggesting that the representation of the input's verb is sufficiently separable from that of the object so as to not be affected by object replacement.

Interpreting Analogical Arithmetic Results

The patterns observed in subject, verb, and object replacement suggest some broad trends in how these inattentive models are representing their inputs.

1. **Models (only) encode verbs analogically.** The robust ability of verbs to both (a) be targeted for replacement and (b) remain unaffected when other categories are targeted for replacement under arithmetic surgery suggests that the representation of the inputs verb in both SRN and GRU models encodes the same kind of linear separable, behaviour which allows for the kinds of analogical arithmetic observed in word embedding models. However, this performance doesn't extend quite as well to other categories. SRNs display decent subject and object performance on subject and object replacement tasks (accuracies between the high seventies and low nineties), although the overlap of errors between these two causes overall sentence performance to become 'just okay,' indicating that while the encodings

of each constituent might be decently analogical, the overall encodings of input sentences are less so. GRU models exhibit a clear failure to represent subjects and objects analogically, preferring instead to spuriously over-generate reflexive forms in a way inconsistent with targeted category replacement.

2. **SRNs prefer encoder resolution.** On subject replacement tasks, SRN models display a clear preference for decoding arithmetic inputs in a manner consistent with encoder resolution of anaphora: when subjects are replaced, if an SRN does decode the result as reflexive it will do so with the identity of the original object preserved. However, the poor performance of SRN models on reflexive predicates, as compared to their performance on non-reflexive and pseudo-reflexive ones, indicates that the encoder's representation of *herself* is not actually identical to that of its antecedent, and is instead more tightly bound (and less separable) to the subject's identity than an ordinary object's is.
3. **GRUs prefer decoder resolution.** On reflexive predicates undergoing subject replacement, GRUs display a clear preference for decoding the result in a manner consistent with decoder resolution of anaphora, where the object of the output is identical to the new subject. However, in light of the abysmal performance of GRU models on subject replacement involving NR and PR operands, this preference is likely a byproduct of how GRU models over-generate reflexive outputs.
4. **GRUs over-generate reflexive outputs.** GRU models have an incredible strong preference for decoding any element of their encoding space as a reflexive output unless given strong evidence otherwise. This behaviour allows it to accidentally attain high performance on the tasks involving decoding arithmetic results which have a reflexive interpretation, but yields exceptionally poor performance on when the result is not interpretable as a reflexive.

Overall, these results present strong evidence that the representations of reflexive anaphora in SRN and GRU encoding spaces are very different. GRUs seem to have learned to interpret reflexive inputs as an 'elsewhere' condition, where non-reflexive inputs are learned properly by any points outside of some tolerance from these known quantities results in an encoding vector that is interpreted as reflexive by the decoder. In contrast, SRN models display no such pattern, exhibiting representations of inputs which are not quite linear in the way that word embeddings are, but which nevertheless do capture some degree of analogical representation.

5.3 Characterizing Encoding Space

While the exploration of how models decode inputs resulting from the kind of analogical arithmetic which has been used to great effect in word embedding models suggests some interesting interpretations for how our models are making use of their hidden representations and computational design to solve the problem of anaphora resolution, this experimental setup suffers from some methodological weaknesses. First, it relies upon the potentially unfounded assumption that the encoding space of these recurrent models displays the same kind of linearity properties which permit word embeddings to display this kind of analogical representation. As we saw, our models were not uniformly good at decoding the results of our CATEGORY-replacement surgery in the way expected if the encoding representations were truly linear, limiting the degree to which we can interpret the success or informative failures as truly representative of how models are operating. Second, even in the cases of success, this experimental design privileges the decoding of certain vectors in the encoding space (those resulting from this arithmetic) has having a particular degree of interpretability while ignoring the rest of the encoding space.

To ameliorate both of these worries, we next consider a weakened version of this task which simply seeks to characterize how our models make use of their encoding space. We know from Chapter 3 that both SRNs and GRUs are able to solve the anaphora resolution task quite well. This involves, in part, a decoder learning to correctly interpret a vanishingly small subset of a model's encoding space. The decoder only learns this on the basis of signals of a finite set of points in the encoding space representing the encoded inputs found in the training set, and then successfully generalizing this knowledge to the novel encodings of *Alice*-reflexive sentences. This success raises the question of how decoders have learned to interpret the encoded representation of arbitrary inputs.

To answer this, consider the result of a decoder being given a random, arbitrary encoding vector which appears to be drawn from the model's known distribution of input encodings. This encoding does not have a *per se* interpretation as an input, since it is not the encoding of any actual sequence of input tokens, and so any decoding given by the model cannot really be regarded as "correct" or "incorrect." Nevertheless, examining patterns in how models decode such random inputs may reveal trends in how they make use of their input encodings.

To formalize this notion for a given model with parameters θ we consider the set of all possible input encodings, and construct a normal distribution $\mathcal{N}(\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta^2)$ defined by mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ of these encoding vectors taken independently over each dimension. We then draw a random vector v from this distribution and give it to the model's decoder. If over a large sample of such vectors we observe a pattern in the decoded output sequences, this may paint a picture of how a model's decoder sees the learned encoding space and hence give insight into how the model has managed to use the encoding space to solve the anaphora resolution problem. This is particularly of interest in the case of the GRU models which, as shown in the previous section, appear to exhibit a strong tendency towards producing reflexive forms when the vectors passed to the decoder do not have a reason to be reflexive, and indeed are analogically interpretable as being non-reflexive.

To test this, for each model we draw $10,000$ vectors uniformly at random from a normal distribution defined by the encodings of all possible input sequences. We then decode these vectors and characterize their outputs on the basis of reflexivity: that is, what proportion of such arbitrary vectors are decoded to reflexive outputs? Table 5.4 below reports the results for this experiment by model type. SRNs decode as reflexive outputs roughly 17% of all vectors drawn uniformly at random from the normal distribution defined by the known encodings. GRUs, by contrast, do so 99% of the time. For comparison, only about 7% of all input sentences (in both the training and generalization domains) are reflexive or pseudo-reflexive. This means that both SRN and GRU models have, to some degree, overgeneralized knowledge of reflexive constructions in the input data by interpreting an uncharacteristically high portion of the 'known encoding space' (as defined by the distribution of known input encodings) as corresponding to semantically reflexive outputs. However, the degree to which the models do this is qualitatively very different. While SRNs exhibit a relatively slight predilection for decoding arbitrary inputs as reflexive, GRUs do so categorically; nearly all arbitrary inputs to the GRU decoder are treated as reflexive. This lends credence to the hypothesis developed in the previous section that GRU models have learned to interpret reflexivity as an 'elsewhere condition' by treating inputs as reflexive by default and carving out only a small portion of the encoding space for non-reflexive inputs despite the fact that the vast majority of inputs seen during training are not reflexive.

	SRN	GRU	Input data
Proportion Reflexive	0.165	0.987	0.069

Table 5.4: Reflexivity of decoded outputs of vectors drawn uniformly-at-random from a normal distribution defined by the encodings of valid inputs

5.4 Conclusion

This chapter attempts to analyize the models trained in Chapter 3 to better understand how such models are able to solve the problem of anaphora resolution. As theorized by G. F. Marcus (1998a) and supported experimentally by Frank, Mathis & Badecker (2013), the task of anaphora resolution demonstrates a degree of algebraic generalization thought to be beyond the computational capabilities of recurrent neural networks lacking any inductive bias for this problem. That we are indeed able to solve this problem with such models, including very simple SRNs, demands explanation.

As noted in Chapter 2, the task of anaphora resolution requires learning a permutation group structure which treats reflexives as fixed points; we believe on this basis that the task of anaphora resolution is harder, in a theoretical sense, than the identity problem posed by G. F. Marcus (1998a), which merely requires learning equivariant functions over the correct equivalence classes as defined by the learned permutation group structure. Our results then demonstrate that simple recurrent networks lacking inductive biases are at least capable of solving problems requiring an ℓ_3 level of generalization. Yet the mechanisms by which these models manage to do so is not yet clear.

By examining the encoding space (sections 5.2 and 5.3) we tried to elucidate these mechanisms. The results we obtained are unclear. The results of the encoding space analysis do clearly show that GRU models have a strong inclination to decode encoding vectors as reflexive unless given enough of a reason to not do so; this is suggestive of a decoder which views the encoding space as a sea of reflexivity, where inputs have by default only a separable verb and subject-object interpretation. Inputs required to have a different subject and object (i.e., excluding reflexive and pseudo-reflexive inputs) are then placed in the encoding space in islands of interpretable non-reflexivity. GRUs, it seems, have learned reflexivity as an elsewhere condition; a clever solution to the problem, although one which poses challenges for its interpretability in the context of how human language learners view the challenge of sentence interpretation. By analogy, one can imaging an experimental setup where native speakers are played a recording of various input sentences with white noise masking out the object, and are then asked to provide their guess as to what the object of the sentence was. It seems highly unlikely that such a test would find speakers likely to interpret such inputs as reflexive on a majority of occasions, suggesting a wholely different mechanism of interpretation than what we observe in GRU models.[2]

That GRU models learn this as an elsewhere condition is likewise notable for the relative paucity of reflexive data in the training domain: only XX% of all training inputs were reflexive, indicating that GRU models managed to generalize (and indeed, perhaps overgeneralize) this rule for reflexive interpretation on the basis of very little support.

Despite this categorical treatment, our attempts at providing a constructive model for how the encoder is representing input sequences in a manner consistent with the observed generalization only yielded further questions. The various hypothesis tested for tensor-product representations of the encoding space all seemed to further give credence to the notion that GRUs have a reflexive-by-default encoding structure, but none seem to adequately characterize the representation of the encoding space.

[2] Of course, the models studied here are clearly not proxies for general human cognition, and so this failure is of course expected. Nevertheless, the patterns observed do clearly indicate that the mechanism by which these models manage to solve the problem of anaphora resolution is qualitatively distinct from that employed in human understanding of this phenomena.

Pre-`<EOS>` Arithmetic

Prior to embedding and subsequent encoding, all input sequences to the models discussed here undergo a process called tokenization. In larger models like BERT (Devlin et al. 2019), tokenization is used in part to break up morphologically complex words into smaller chunks. Our models developed here do not make use of this, but the process of tokenization does involve the appending of a start-of-sequence `<SOS>` token to the beginning, and an end-of-sequence `<EOS>` token to the end, of each sequence, as shown below in (50).

(50) *Alice sees herself* ⟶ `<SOS> alice sees herself <EOS>`

While these extra tokens do not have their own interpretations in the output domain[3], their presence in the input sequence does pose an additional way to examine how models handle CATEGORY-replacement surgery. Since each input sequence ends with the same `<EOS>` token, we can optionally perform the arithmetic immediately after the encoder has processed the object token in the operand, before the it has had a chance to encoder the `<EOS>` token, as below in (51), where Δ^* denotes a displacement vector which has likewise been computer prior to the `<EOS>` token in each constituent sequence.

(51) $h_{k-1} = \mathrm{Enc}(\texttt{<SOS> alice sees herself}) + \Delta^*_{Alice,Mary}$

This encoding vector is then taken as the hidden input to the encoder when presented with the `<EOS>` token, as in (52) below, and this encoding is what is passed to the decoder.

(52) $h_k = \mathrm{Enc}(\texttt{<EOS>}, h_{k-1})$

The computation of CATEGORY-replacement surgery prior to the `<EOS>` token may change how a model decodes these analogical expressions.

Sequence-to-sequence design as an inductive bias for generalization

The results of Chapter 3 demonstrate conclusively that simple recurrent networks are able to learn problems requiring an ℓ_3 degree of algebraic generalization. One notable distinction between the results shown here and previous negative results obtained in G. F. Marcus (1998a) and subsequently in Frank & Petty (2020) is that we use recurrent networks in a sequence-to-sequence context (Sutskever, Vinyals & Le 2014) rather than a language modelling context. A strong interpretation of this disparity (i.e., where we draw the conclusion that problems of ℓ_2- and ℓ_3-generalization are unlearnable by simple recurrent models in a language-modeling context, whereas ℓ_1-generalization tasks like those of Kim & Smolensky (2021) and Chapter 6 are certainly learnable by language

[3] We do make use of the `<EOS>` token during decoding to determine when a decoder has finished decoding an input.

models such as BERT) would suggest that the sequence-to-sequence design of the models developed in Chapter 3 constitutes an inductive bias for this kind of generalization, one which enables recurrent networks to solve problems which require persisting knowledge of a novel input's identity.

Why this should be the case is unclear. Attention mechanisms provide a clear example of an inductive bias geared to solve such problems at both an intuitive (the presence of a look-back mechanism allows a model to examine the lexical content/identity of a previous token to condition output on this identity) and formal (see the work of Goyal & Bengio (2021) connecting attention mechanisms to the learning of group equivariant functions) level. Contrastingly, there is little intuitive connection between the persistence of a token's identity between the input and output domains and the problem design of separating the input encoding from the output decoding by collapsing the input representation to a single state vector passed from encoder to decoder. While such a design clearly affords a benefit to models seeking to separate the length of the input and output representations (Sutskever, Vinyals & Le 2014), what connection this has to the representation of generalization is unclear. If it is the case that sequence-to-sequence design constitutes such an inductive bias, it would be of great interest to understand what connection this model design has to problems of requiring equivariance.

Testing Explicit Hypothesis for Input Representation

While the analysis done here is illustrative of broader patterns in how SRN and GRU models make use of their embedding space and sequence-to-sequence design to solve anaphora resolution, these methods are limited in their explanatory capabilities since they provide no testable model for how these models are actually representing their inputs in the embedding space. Rather, we merely are observing how the decoder interprets parts of the encoding space and using this to draw conclusions about how the encoder is representing information. If we could provide a constructive analysis of how our models are representing inputs in the encoding space, we can draw better conclusions about the mechanisms by which models learn this representation and understand the limitations to models' ability to solve similar problems in generality.

One method for providing such explicit and testable hypothesis about the representations of input data in a model's encoding space follows from the work of Smolensky (1990), which argues that the representation of inputs to connectionist models can be analyzed in terms of a tensor product between *roles* which are bound to semantically-contentful *fillers*. This representation provides a link between the symbolic structures which characterize linguistic data, including the data presented herein, and the learned continuous spaces of connectionist systems. As an example, consider the following filler-role paradigm for our anaphora resolution task, where ROLE ⊗ *filler* is taken to mean the tensor product of a role and filler pair which have each been embedded into some vector space.

(53) *Alice sees herself* ⟶ SUBJ ⊗ *alice* + VERB ⊗ *see* + OBJ ⊗ *self*

For a network to represent it's inputs in such a fashion means that it has implicitly learned to separate portions of sequences in its input domain and model the binding of the fillers to their respective roles and combining these representations to produce a representation for the broader input.

The connection between this tensor-product representation and the requirements for generalizational structure discussed in Gordon et al. (2019) and elaborated upon in Chapter 2 are clear when we consider each ROLE as the identity of a learned equivalence class and the various *filler*s as the members of the underlying dataset whose orbits (under action by the learned symmetry structure) correspond to these classes. One important clarification, however, is that the fillers of these tensor-product representations need not correspond uniquely to particular tokens in the input domain; rather, they need only represent the interpretational semantics of the input in some well-defined way. Thus, for instance, it is perfectly reasonably to imagine a tensor-product representation of our anaphora resolution task has having a *see-herself* filler even though this is not a unique token present in the input of *Alice sees herself* which would correspond to this filler.

The ability for recurrent neural networks in a sequence-to-sequence context to construct such tensor-product representations is not merely theoretical. McCoy et al. (2020) demonstrates that such recurrent models in fact implicitly capture tensor-product representations by constructing tensor-product decomposition networks (TPDNs) which are trained to model a recurrent model's encoder on the basis of stipulated filler-role hypothesis and then using these TPDNs as substitutes for the model's original encoder. Successfully training a TPDN to model an encoder's input representations to the degree that the TPDN can be substituted for the original encoder in the sequence-to-sequence task then provides a constructive characterization for how such a model is representing inputs in its embedding space.

We believe a similar technique could be employed here to test hypothesis about how our SRN and GRU models are representing reflexive anaphora in a manner consistent with the generalization behavior observed by considering candidate representations whose decomposition could extend to an unseen subject-reflexive combination and then using a TPDN to model the encoding spaces of the models under this candidate representation.[4]

[4] As an example, the filler-role paradigm illustrated in (53) is compatible with the generalizations observed because the proposed representation for a withheld sentence like *Alice sees herself* (⟶ SUBJ ⊗ *alice* + VERB ⊗ *see* + OBJ ⊗ *self*) is separable into constituent filler-role bindings which are each learned separately in training; but consider an "all-in-one" representation like that of (54) below.

Extending Analysis to More Complicated Models

We limited our analysis to only inattentive SRN and GRU models, since it is these networks whose successful performance on the anaphora resolution task is most surprising. Nevertheless, it is likewise of interest to understand the mechanisms employed by more complicated models like inattentive LSTMs, the attentive variants of all recurrent architectures used, and transformers, to solve this problem.

Likewise, extending this analysis to the additional experiments mentioned in Chapter 3 would likewise yield insight into how varying the task difficulty by reducing training support affects model performance. We know empirically that the simplest networks surveyed, inattentive SRNs, exhibit a strong degradation of generalization when the training support of reflexive sentences is progressively decreased, yet the GRU models used showed a remarkable resilience to this increased difficulty, with some models attaining near-ceiling performance when the training data contained very little evidence of feminine-reflexive sentences. This disparity, and the fact that inattentive GRUs are able to solve this increasingly-difficult problem at all, demands an explanation. By characterizing how all models represent their inputs in encoding space, we can provide a more thorough comparison of how such models manage to solve problems requiring algebraic generalization and perhaps provide a qualitative comparison of how model architecture relates to the learnability of problems of different classes of generalization complexity.

(54) *Alice sees Bob* ⟶ SENT ⊗ *alice-see-bob*

While assigning a unique filler to each sentence would successfully model the inputs of the training domain, it is not possible for this representation to be learned in such a way so as to extend to the withheld data, since these bindings (SENT ⊗ *alice-see-herself* for *Alice sees herself*) would never be attested in training.

Part II

Pretrained Language Models

Chapter 6

Do Language Models Learn Position-Role Mappings?

6.1 Introduction

During language learning, children come to know what thematic relations hold between verbs and their syntactic arguments. In a prepositional dative (PD) construction like (55a), the first object is assigned the THEME role, while the second (prepositional) object is assigned RECIPIENT. In contrast, in double object (DO) constructions like (55b), it is the first object that is assigned RECIPIENT, while the second is assigned THEME.

(55) a. I gave [the ball] to [the dog].
b. I gave [the dog] [the ball].

When such sentences are passivized, the position-role mapping changes yet again: for PDs, the subject of the sentence now takes the role of THEME, while for DOs, the subject takes the role of RECIPIENT.

(56) a. [The ball] was given to [the dog].
b. [The dog] was given [the ball].

Such patterns raise a learning problem: how do learners come to know which thematic role to assign to a given syntactic argument? We might, for instance, expect that a learner who has acquired the position-role mapping for a DO sentence would generalize her knowledge of the considerably more frequent passives of transitive verbs to passives of DO sentences. In passives of sentences with transitive verbs, it is the THEME role, as opposed to the RECIPIENT, that is assigned to the passive subject.

(57) a. I threw [the ball].
b. [The ball] was thrown.

Since THEMES are the subjects of passives in these simpler structures, a learner might be tempted to (erroneously) accept examples like the following:

(58) *[The ball] was given [the dog].

Strikingly, this pattern of raising the THEME but not the RECIPIENT in DO sentences is cross-linguistically unattested.[1] Such a gap calls out for explanation in terms of the process of language learning.

One way a learner could avoid such a faulty generalization would be if the primary linguistic data included evidence that directed a learner away from it. Indeed, given sufficient evidence about the thematic properties of the arguments of verbs in both active and passive DO structures, a learner might eschew any generalization between active and passive entirely, favoring instead a structurally specific mapping for each sentence type. Such an approach would however fail to capture the systematicity of the relationship across argument structure alternations like dative shift in (55) and different syntactic variants like the voice alternations between active and passive. Further, if generalization is eschewed entirely, we might expect the properties of individual verbs to be learned separately (e.g., Tomasello 1992). While verbs exhibit well-known variability in their participation in argument structure alternations (e.g., *give* participates in dative shift but *donate* does not), the relationship between the active and passive forms is entirely regular: if a verb can appear in an active DO sentence, it can also appear in a passivized DO sentence, with thematic properties that are entirely predictable. Verb- and structure-specific learning would not provide an account of such systematicity and would not support generalization to forms that are sparsely represented in the learning data.

An alternative approach, widely adopted in work in generative grammar, posits the presence of an innate language-specific learning bias that constrains position-role mappings. For example, in work on the acquisition of argument structure, whether rooted in semantic bootstrapping (Pinker 1989) or syntactic bootstrapping (Gleitman 1990), the child is assumed to know the relationship between the thematic roles of events of transfer on the one hand, and syntactic positions in a double object or prepositional dative sentence on the other. Similarly, syntactic theories derive the fact that passivization of a DO structure necessarily allows the promotion of the the argument occupying the highest (indirect) object to subject position from properties of the syntactic representation of such constructions (cf. Alsina (1996), McGinnis (2002), Holmberg, Sheehan & van der Wal (2019) *inter alia*).

A final possible account of this learning problem, which constitutes a middle ground

[1] Bresnan & Moshi (1990) (and much subsequent work) explore languages like Kichega which allow "symmetric" passives, where either argument in a DO construction can be raised to subject position. Certain English dialects also permit symmetric passives (Woolford 1993). Such languages and dialects still raise the learning problem we discuss here, though in a modified form. We leave the exploration of this variation for the future.

between these two approaches, would attempt to derive constraints on the acquisition of position-role mappings from the combination of domain-general learning biases and the evidence present in the learning data. Such an approach to language learning is now widespread in work in NLP, where contemporary language models have little in the way of hard-wired linguistic structure, but the linguistic generalizations they learn are indeed guided by the properties of their architectures and the data to which they are exposed (McCoy, Frank & Linzen 2020, Min et al. 2020, Mulligan, Frank & Linzen 2021). Most often these models are evaluated on the basis of their performance on some extrinsic task like question answering or natural language inference (NLI), and such results do not shed light directly on the nature of linguistic generalizations that they encode. Here, we consider the linguistic generalization question more directly by studying the degree to which a widely-used language model, the Bidirectional Encoder Representations from Transformers model (BERT, Devlin et al. (2019)), exhibits knowledge of position-role mappings across variations in argument structure, syntactic structure, and lexical identity. BERT is a general purpose neural network architecture that composes multiple Transformer layers (Vaswani et al. 2017) each with a bidirectional attention mechanism. It is trained to perform a masked language modeling task (i.e., to predict the identity of masked tokens within a sentence) using a data set consisting of the 800M words of the BooksCorpus and the 2,500M words of English Wikipedia.[2] Quite clearly, BERT lacks explicit linguistic bias on what can constitute possible position-role mappings and how these mappings can vary across structures. As a result, any knowledge in this domain that it demonstrates must derive from the combination of its training data and domain-general biases that stem from the transformer architecture.

6.2 Experiment 1: Probing Position-Role Mappings Through Distributional Similarity

Our first experiment tests BERT's knowledge of the position-role mappings for the THEME and RECIPIENT arguments of ditransitive predicates. To do this, we make use of constraints imposed by the selectional restrictions of verbs: the limitations that a verb imposes on the content of its arguments. Though such restrictions are verb specific (e.g., a verb like *drink* will take different direct objects than a verb like *read*), there are nonetheless general distributional patterns that can be associated with more coarse-grained thematic roles. If a verb assigns the role like AGENT or RECIPIENT to an argument, we would expect the distribution of that argument's head nouns to favor animate nouns.

[2] BERT's training regimen also includes a next sentence prediction task, in which it must be determined whether or not two sentences were originally adjacent to one another in the source text. Subsequent work with a BERT-variant called RoBERTa (Liu et al. 2019) has found this component of training to be unnecessary to its success.

In contrast, for arguments assigned the THEME role, we might expect a higher proportion of inanimates (or at least the absence of a strong animacy preference).

Because BERT is trained to perform masked language modeling, it can be used to extract distributional predictions directly. For this experiment, then, we presented BERT with sentences containing ditransitive predicates with the head nouns of the THEME and RECIPIENT arguments masked out:

(59) Alice sent the [MASK] a [MASK] .

If the predicted distributions of nouns in multiple argument positions of a single sentence, say an active double object example, are distinct, this provides a first bit of evidence of BERT's knowledge of the distinctive properties of these arguments. We use this approach to systematically assess BERTs knowledge of selectional restrictions by performing two calculations. First, for each positions predicted distribution, we compare the total probability assigned to a small set of (frequent) animate nouns A with the probability assigned to a small set of (frequent) inanimate nouns I.[3] We call the result the animacy confidence (aconf) of position w_i:

$$\mathrm{aconf}(w_i) = \log \frac{\sum_{v \in A} p(w_i = v)}{\sum_{v' \in I} p(w_i = v')}$$

By computing mean aconf across comparable positions in a set of sentences of the same type, we can get a measure of the model's overall preference for animate nouns in a given position. Reliable differences between means in different positions will point to a representation of the different roles. A more interesting question that aconf scores allow us to ask is the degree to which they are consistent across different syntactic realizations of the same argument: do THEMES and RECIPIENTS in double object structures have the same profile as THEMES and RECIPIENTS in prepositional datives? And does the passivized version of each structure show the right pattern of aconf scores for the corresponding argument positions?

One limitation of aconf scores is their dependence on the specific sets of nouns A and I we use to evaluate the preference. To assess the distribution in a more neutral fashion, we also compute the entropy of position w_i.

$$H(w_i) = -\sum_{v} p(w_i = v) \log p(w_i = v)$$

Higher entropy is associated with a more diffuse set of predictions, i.e., cases where the language model is less certain about the identity of the words that can fill a posi-

[3] We use the following sets of nouns. Animate: person, man, woman, student, teacher, king, queen, prince, princess, writer, author, builder, driver, human, dog, bird, dancer, player, angel, actor, actress, singer, director, bee, friend, wolf, lion, scholar, pirate, spirit, fox. Inanimate: apple, book, chair, table, phone, shoe, water, earth, land, light, sun, moon, plate, eye, ear, branch, tree, time, energy, bottle, can, mask, leaf, tile, couch, button, box, cap, wire, paper.

tion. Different thematic roles impose varying degrees of selectivity on their associated arguments, and consequently, entropy measures can provide us with a diagnostic of such selectivity that we can compare across arguments of the same role in different syntactic constructions and voices.[4] Using the `tregex` tool (Levy & Andrew 2006), we extracted sentences from the Wall Street Journal portion of the Penn Treebank (PTB, Marcus, Santorini & Marcinkiewicz (1993)) containing ditransitive predicates, using both double object and prepositional dative structures, in both active and passive voice. For each structure-voice pairing, we selected 50 sentences, and masked the head noun of the THEME and RECIPIENT arguments. We evaluated these data on the BERT model. In order to examine what effect, if any, variations in model architecture and training regimen had on performance, we also examined the behavior of two recently developed variants, RoBERTa (Liu et al. 2019) and DistilBERT (Sanh et al. 2020). RoBERTa utilizes the same architecture as BERT while modifying the pretraining regimen. DistilBERT, by contrast, uses a different, smaller architecture with roughly 40% fewer parameters, while retaining high levels of performance. For space considerations, we only report results from the BERT model, but results were consistent between all three model architectures.

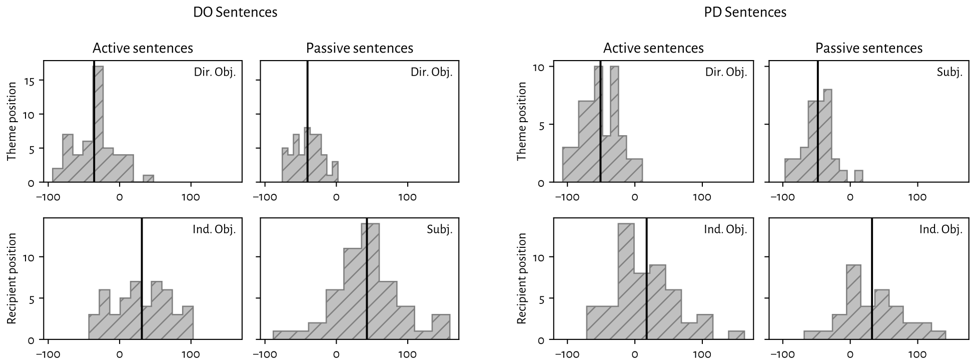

Figure 6.1: Animacy confidence of THEME- and RECIPIENT-expecting positions in active and passive double object sentences (left) and prepositional dative sentences (right) from the Penn Treebank. Vertical lines indicate mean values.

Both animacy confidence (Figure 6.1) and entropy (Figure 6.2) show consistent differences between THEME-expecting and RECIPIENT-expecting positions across double object and prepositional dative constructions, as well as across active and passive variants.

[4] One potential pitfall with this approach stems from the variability in the selectivity associated with individual roles (Resnik 1996). For example, while some transitive verbs like 'drink' restrict their THEME arguments to words denoting liquids, others like 'see' are much less limiting on their themes. Nonetheless, our goal here is exploring the possibility that entropy measures support systematic differences at the thematic role level.

In each case, the mean animacy confidence is negative for THEMES (meaning a preference for inanimate nouns) and positive for RECIPIENTS (meaning a preference for animates), and the mean entropy value is higher for RECIPIENTS than it is for THEMES. The difference in means between THEME- and RECIPIENT-positions is statistically significant under a two-sided Welch's unequal variances t-test with $p < .001$ for animacy confidence and $p < 0.05$ for entropy.

This consistently distinct treatment of THEME and RECIPIENT arguments across different argument structures and across active and passive constructions is suggestive of the fact that pretrained language models have knowledge of how thematic relations are realized across different syntactic structures in ditransitive constructions. This is notable not only for the sensitivity it requires to grammatical context but also because the alternation between active and passive voice in double object constructions exhibits the unusual property discussed above, where the RECIPIENT which is promoted to subject position under passivization, rather than the THEME. It appears then that language models trained on massive corpora are not only capable of learning role restrictions across syntactic contexts but that they are able to put aside widely supported generalizations in specific cases, in the absence of an explicit bias to do so.

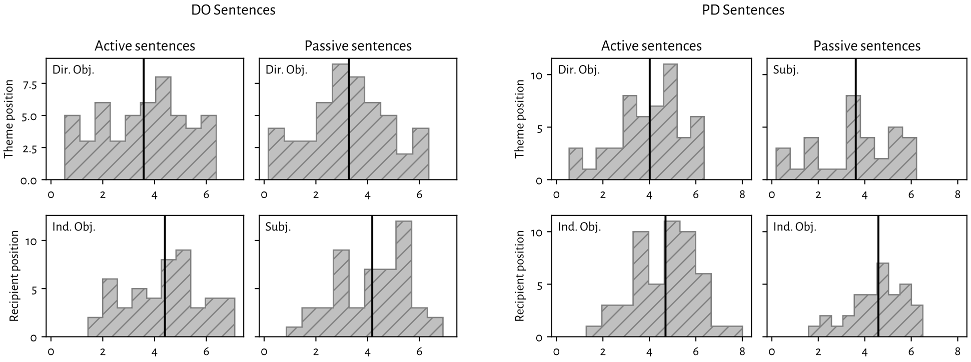

Figure 6.2: Entropy of THEME- and RECIPIENT-expecting positions in active and passive double object sentences (left) and prepositional dative sentences (right) from the Penn Treebank. Vertical lines indicate mean values.

6.3 Experiment 2: Syntactic and Structural Generalization

The results of Experiment 1 show that BERT (and its variants) exhibits latent knowledge of the connection between argument structure and thematic role across voice (that is, in both active and passive constructions) in ditransitive sentences. Yet there is no guarantee that the model has any *shared* knowledge connecting the alternative structures (DO or

PD) or the active and passive constructions. For example, does the model understand that the RECIPIENT position in an active DO sentence (the indirect object) corresponds to the RECIPIENT position in the passive one (the subject), or has it simply learned the thematic role/argument structure correspondences in these two sentence types independently?

To test whether the knowledge is shared or independent, we adapt the method proposed by Kim & Smolensky (2021) to diagnose linguistic generalization in language models: fine-tuning an already trained language model on sentences that include novel words that are associated with some linguistic property. During fine tuning, these words are only presented to the model in a single syntactic context. We then test the model's ability to generalize its knowledge of these novel words to structures in which they had not been seen during fine-tuning.

Our adaptation of this methodology involves the use of novel nouns that occur uniquely in positions associated with specific thematic roles: *thax* as a theme and *ricket* as a goal. We take our three BERT variants and fine-tune separate models using one or two paradigms: DO 'give' and PD 'give'. The DO paradigm contains hand-constructed sentences containing only DO examples, and likewise for the PD paradigm.[5] Example (60) below gives the full set of tuning data for the DO 'give' paradigm.

(60) a. I gave the *ricket* a box.
b. I gave a *ricket* the camera.
c. I gave the teacher a *thax*.
d. I gave a student the *thax*.

The intuition behind this set-up is similar to what we explored in Experiment 1: the semantic classes of nouns appropriate for the different thematic roles differ in systematic ways, and such selectivity will vary systematically across the different position-role mappings. Our expectation is that fine-tuning will lead the language model to identify the relevant properties of these nouns. If the language model represents position-role mappings in a way that generalizes across argument structure alternations and variations in syntactic structure, we should see generalization of its predictions of the novel items to other syntactic structures.

Following Kim & Smolensky (2021), we freeze all of the model weights prior to fine-tuning except for the word embeddings of two unused items in the model's vocabulary.[6] We then fine-tune the model on a minimal synthetic dataset such as the one in (60) until the predicted log probability of either of the novel tokens in a masked position begins to

[5] Importantly, each training sentence contains only a single novel token, either *ricket* or *thax*. These novel tokens never appear together in the same training example to prevent the model from learning any association between them. Thus the model will never learn that if it sees *ricket* in one position, it should expect *thax* in another.

[6] BERT-variant models utilize shared input and output embeddings, so the model is able to learn to predict novel words even though all weights except for two input embeddings have been frozen.

sharply decrease. We use this early-stopping criterion in an attempt to avoid the model becoming overly confident in the prediction of the novel tokens at the expense of the rest of its vocabulary. We use unused tokens in BERT's vocabulary to represent the nonce words.

We evaluated the tuned models' performance on a number of synthetic test sets containing masked THEME-expecting and RECIPIENT-expecting positions, as in (61) below. Examples in these sets varied with respect to the choice of determiners, non-masked nouns, syntactic frame (DO vs. PD), voice (active vs. passive), and verb. In each masked position, we compute the log of the probability ratio (so-called log odds) of the two novel words; if the log-odds of the novel *thax* tokens are higher than those of the *ricket* tokens in THEME-expecting positions, and vice-versa for the RECIPIENT-expecting positions, we infer that the model has learned to distinguish the distributions of these nonce words.

(61) a. The teacher gave a [MASK] the [MASK].
     b. A [MASK] was given the [MASK].

(62) a. A teacher gave the [MASK] to the [MASK].
     b. The [MASK] was given to the [MASK].

(63) a. The teacher sent a [MASK] a [MASK].
     b. A [MASK] was sent a [MASK].

(64) a. The teacher sent the [MASK] to a [MASK].
     b. The [MASK] was sent to a [MASK].

This regimen of training and evaluation allows us to measure a model's proclivity for generalization in syntactic (voice) and structural (frame) contexts. Table 6.1 summarizes the performance of our 3 BERT-variants models by reporting the percentage of novel tokens which are correctly predicted in evaluation sentences across voice and frame. DO and PD rows in the table correspond to different training regimens.

Voice Generalization

Voice generalization measures a model's ability to infer the placement of the novel *thax* and *ricket* tokens in a passive sentence for a model trained only on active sentences, or vice-versa. We know from Experiment 1 that BERT performs analogously (as measured by entropy and animacy confidence) across corresponding positions in active and passive constructions. By testing BERT on novel token prediction, we can determine whether knowing how to place tokens in one construction suffices to know where to place tokens in another (in essence, whether BERT can make use of this knowledge of distributional similarity). Our results, reported in Table 6.1, show that BERT models do indeed exhibit

	Double Object				Prepositional Dative			
	Active		Passive		Active		Passive	
	TH.	RE.	TH.	RE.	TH.	RE.	TH.	RE.
BERT								
DO	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
PD	100.0	95.0	100.0	100.0	100.0	100.0	100.0	100.0
RoBERTa								
DO	85.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
PD	78.8	82.5	100.0	100.0	100.0	100.0	100.0	100.0
DistilBERT								
DO	90.0	98.8	97.5	100.0	97.5	100.0	80.0	100.0
PD	100.0	62.5	100.0	100.0	100.0	93.8	100.0	100.0

Table 6.1: Performance of various models on placing novel tokens (*thax* or *ricket*) in the correct position (THEME- or RECIPIENT-expecting) within active and passive sentences in DO and PD frames. Columns represent evaluation data while rows represent training contexts. Shaded cells indicate in-domain evaluation results; unshaded cells report generalization results. All training and evaluation sets reported here used a single verb, 'give'. Each reported value is an average over 10 different model runs.

robust voice generalization and are able to accurately predict token placement in passive sentences when trained on corresponding active sentences.[7]

All models evaluated showed equal or improved performance on the passive variants of their active training data. Indeed, all models achieved nearly perfect performance on the passive complements to their active training data. This generalization is supported across frames as well, where models trained on active DO sentences perform well on passive PD sentences.

Frame Generalization

Frame generalization measures a model's ability to infer the distribution of novel tokens on sentences whose frame (DO versus PD) differs from those in the model's training data. We trained models on DO and PD data separately, allowing us to test generalization from DO to PD frames and from PD to DO frames. Just as with voice generalization, we find that all models exhibit good generalization between frames, although there is some variance in the directionality of this success. Models trained on DO data exhibited

[7] Throughout, 'corresponding' means that we've held other relevant parameters constant, so we might train on *active* DO 'give' sentences and test on *passive* DO 'give' sentences. Furthermore, the active and passive sentences correspond to each other as they do, for example, in (61).

excellent generalization to PD data, attaining at- or near-ceiling performance. Models trained on PD data perform slightly less well, though still substantially above chance, on DO data.

Distributional Restrictions on Roles

In all but three cases in Table 6.1, we find that models accurately predict *ricket* in RECIPIENT positions more often than they accurately predict *thax* in THEME positions. This pattern holds across models, frames, voice contexts, and training regimes. This is consistent with our results from Experiment 1, where we found that BERT, RoBERTa, and DistilBERT models had higher animacy confidence for RECIPIENT positions than for THEME positions. Thus, the higher animacy confidence associated with RECIPIENT positions travels together with higher accuracy. Under our hypothesis, this is no accident: the more restricted distribution of words that can appear in RECIPIENT positions (namely, words in RECIPIENT positions are more likely to be animate than words in THEME positions) supports the models' ability to predict the correct token in these positions.

Lexical Generalization

Lexical generalization measures a model's ability to predict novel token placement in sentences whose ditransitive verb differs from the verb in the model's tuning data. Here, we fine-tune models on 'give' sentences and evaluate them sentences with other ditransitives, namely 'teach', 'send', and 'tell'. We have carried out this analysis on the best performing of our models, namely RoBERTa. Our results, shown in Tables 6.2 to 6.4, show that RoBERTa's performance on lexical generalization tasks is lexically conditioned, with high performance on some ditransitive verbs, but quite poor performance on others. Furthermore, the frames on which models fail to generalize are not consistent between target verbs. Test data involving the ditransitive verb 'teach', for instance, is associated with reasonably good performance on both DO and PD constructions in both active and passive forms, regardless of training context. In contrast, models evaluated on data involving 'send' as the ditransitive verb show reasonably high performance on prepositional dative constructions (in both active and passive forms), but show much worse performance on double object constructions (of both voices), regardless of training context. Finally, the opposite pattern holds for test data involving 'tell', where the prepositional dative constructions yield worse results than the double object ones.

In all such cases, it is notable that the same pattern of performance holds vis-à-vis the accuracy of the models placing *ricket* in a RECIPIENT context relative to their placing *thax* in a THEME context. Indeed, this effect is pronounced in cases where the models exhibit a stark failure of generalization, as in the 'tell' frames of Table 6.4, where the models' performance in RECIPIENT-expecting positions was near ceiling while their performance on THEME-expecting positions was far lower. In all cases observed, a failure of generalization for the whole frame is due almost entirely to a failure to place *thax* tokens in THEME-expecting positions.

Compared to in-domain test cases, evaluation on sentences with novel verbs shows a greater distinction in accuracy between RECIPIENT- and THEME-expecting positions, with mean accuracy for THEME-expecting positions substantially lessened while that of RECIPIENT-expecting positions remained roughly at ceiling. This further highlights the impact of the distributional restrictions placed on the RECIPIENT position as observed in Experiment 1 by measuring relative entropy and animacy confidence.

One natural place to look as the source of these lexical distinctions is in the training data. If the model's experience with different verbs during training reveals divergent distributional patterns, we might expect the network to generalize less well. Because of the infeasibility of assessing the distributions in the BERT or RoBERTa training data, we instead explored the relative abundance of the different structures in parsed PTB data.[8] Though the syntactic annotation provided in the PTB does not allow us to perfectly identify double object structures (argument and adverbial NPs are parsed identically), the resulting patterns are robust enough to allow us to identify interesting patterns for the verbs 'send' and 'tell': 'send' appears far more often in prepositional dative constructions than in double object constructions, while the opposite is true for 'tell' (which occurs almost never in the prepositional dative construction). The verb 'teach' shows a strong bias towards the double object construction, but it is considerably rarer than the other three verbs. This suggests that its corpus statistics are less reliable indicators of the training data used for the language models and therefore not predictive of lexical generalization performance. If taken as a proxy for the relative abundance of these forms in the training corpus for the language models, this could suggest that the points of failure for lexical generalization tasks are correlated with the frequency with which dative verbs appear in the various construction types in the training data.

We find that pretrained language models exhibit robust generalization across voice and construction type in ditransitive constructions when introducing novel THEME- and RECIPIENT-like tokens into their vocabularies. This ability holds across model type, though we do find evidence that performance is lexically conditioned by the ditransitive verb used during the fine-tuning process. This suggests that while the knowledge of the relationship between syntactic position and thematic role is not learned wholly independently for each construction type, it is dependent on the identity of the ditransitive verb involved.

[8] We recognize that the PTB data is not identically distributed to the BooksCorpus and Wikipedia that forms the BERT training set, we expect that the usages of different structures would be reasonably consistent across them, at least at the coarse-grained level we are considering here.

	Double Object				Prepositional Dative			
	Active		Passive		Active		Passive	
	TH.	RE.	TH.	RE.	TH.	RE.	TH.	RE.
RoBERTa								
DO	97.5	100.0	100.0	100.0	97.5	98.8	97.5	100.0
PD	92.5	100.0	90.0	100.0	90.0	100.0	95.0	100.0

Table 6.2: Lexical generalization to 'teach' frames. Columns represent evaluation data while rows represent training contexts. All models were trained on active 'give' sentences. Each reported value is an average over 10 different model runs.

	Double Object				Prepositional Dative			
	Active		Passive		Active		Passive	
	TH.	RE.	TH.	RE.	TH.	RE.	TH.	RE.
RoBERTa								
DO	77.5	100.0	77.5	95.0	91.3	98.8	90.0	100.0
PD	71.3	91.25	72.5	92.5	100.0	100.0	97.5	100.0

Table 6.3: Lexical generalization to 'send' frames. Columns represent evaluation data while rows represent training contexts. All models were trained on active 'give' sentences. Each reported value is an average over 10 different model runs.

	Double Object				Prepositional Dative			
	Active		Passive		Active		Passive	
	TH.	RE.	TH.	RE.	TH.	RE.	TH.	RE.
RoBERTa								
DO	78.5	100.0	97.5	100.0	23.8	98.8	45.0	92.5
PD	62.5	100.0	65.0	100.0	26.3	100.0	15.0	100.0

Table 6.4: Lexical generalization to 'tell' frames. Columns represent evaluation data while rows represent training contexts. All models were trained on active 'give' sentences. Each reported value is an average over 10 different model runs.

6.4 Conclusion

We began by raising the question of how children might acquire position-role mappings, and outlined three possibilities: verb- and syntax-specific learning, innate language-specific biases, and a combination of domain-general biases and evidence in their linguistic input. We have demonstrated here that the third option is a feasible explanation: three language models that contain no explicit linguistic biases regarding possible position-role mappings nevertheless successfully demonstrate knowledge of position-role mappings that largely generalizes across verbs and syntactic structures. The limitations we find do not invalidate this larger conclusion, though they do suggest the importance of further research in this area.

We have shown that pretrained language models (BERT, RoBERTa, and DistilBERT) recognize distributional differences between THEME- and RECIPIENT-expecting positions. This distinction is stable across syntactic (i.e., voice) and structural (i.e., direct object vs prepositional dative) alternations, showing that these well-performing pretrained language models appear to have knowledge of position-role mappings which are preserved between construction type and voice alternations in ditransitive constructions. We have further shown that this knowledge is, in some sense, 'shared' across syntactic and structural alternations. Models that are fine-tuned to learn the novel *thax* (theme-like) and *ricket* (recipient-like) tokens within a single paradigm (e.g., active prepositional dative constructions or active double object constructions) make robust generalizations across voice and construction alternations.

We do however find limitations in the performance of these models with respect to lexical generalization. When the model is exposed to a novel token as the argument of one verb, it generalizes this knowledge to other verbs in an inconsistent fashion. For instance, models trained on 'give'-containing sentences poorly generalize their knowledge of THEME arguments in prepositional dative structures containing the verb 'tell'. Nonetheless, even in such case, models perform well at generalizing knowledge of RECIPIENT arguments. This fits with our earlier observation that RECIPIENT positions have higher animacy confidence than THEME positions do, so that a model's knowledge that a novel token has an animate interpretation will license generalization.

This conclusion suggests a hypothesis concerning how the model may be succeeding in our novel word learning experiments, namely by associating the novel word with a portion of the word embedding space that is appropriate for the selectional restrictions of the verb on which it is trained (i.e., 'give'). This is consistent with the network having learned a distinct and redundant representation of the selectional restrictions across syntactic contexts, so long as they are all characterized in terms of the abstract lexical semantic space represented through the word embeddings. In on-going work, we are exploring other experimental methods to identify knowledge that cuts across structures.

Further directions for work include assessing whether the patterns of generalization we have found here also hold within a broader array of syntactic (e.g., raising) and structural (e.g., causative-inchoative) alternations, as well as better elucidating the computational mechanism by which these models are able to make these kinds of generalizations. We want to better understand whether or not the generalized knowledge observed in these models is derived from the computational mechanisms internal to the language model (i.e., is knowledge of a task like passivisation a property of how BERT encodes sentences and operates on them as data is passed through the model) or is it merely a learned property of the input and output embeddings (i.e., has BERT learned that 'things which are valid subjects of passive sentences' should cluster in some subspace of the embedding space, and that 'things which are the direct object of prepositional dative constructions or the indirect object of ditransitive constructions' should cluster in another subspace, and that nouns with the appropriate distribution are embedded in such a way so as to satisfy both constraints simultaneously). Essentially, this questions probes the degree to which BERT's displayed success at these lexical generalization tasks is a result of some shared computational knowledge of how language works, or whether it is merely a consequence of the learned geometry of its embedding space.

Chapter 7

Conclusion

This thesis takes a small step towards elucidating the connection between the complexity of a generalization task and the learnability of this task by artificial neural networks. We began by introducing the various analyses of generalization that have been employed to describe problems arising in the learning of linguistic data. These descriptions provide very intuitive accounts of the knowledge required to represent language data in a way which generalizes to unseen constructions, but their lack of formal definition precludes a clear comparison of the problems they describe. We adopt the analytical framework of Gordon et al. (2019) and others to connect the notion of generalization to the learning of group-equivariant functions which encode symmetries in training data. Originally used to formalize the notion of compositional generalization (Lake 2019), we extend this formalize to reify G. F. Marcus (1998a)'s notion of algebraic generalization, providing a common formal description for the problems described in these notions. We then show how algebraic generalization describes a subset of the larger task of compositional generalization, and construct a complexity hierarchy of generalization tasks based on the complexity of the learned representation of the task in terms of permutation groups and equivariant functions.

After establishing this theoretical framework, we examine particular problems situated within this hierarchy and their learnability by 'naïve models'—those without any inductive bias for such required generalization. We examine the problems of identity (G. F. Marcus 1998a) and anaphora resolution (Frank, Mathis & Badecker 2013), showing contrary to earlier evidence that these problems are learnable by such naïve recurrent models. We follow up this surprising empirical result, and its extension to transformer models, with an analysis of how these networks have managed to acquire a generalization previously thought beyond their grasp. We determine that, for a subset of the models developed to solve the anaphora resolution task, these networks manage to acquire this generalization by map reflexive inputs to an 'elsewhere condition' in the encoding space, although a full constructive account of how all these models represent their inputs remains elusive. We then present and discuss some more powerful analytical tools, like the tensor product decomposition networks of McCoy et al. (2020), which may help us elucidate exactly how these small, inattentive models acquire and represent algebraic knowledge in a generalized way.

We follow up this examination of small neural networks by examining the capability of large, pretrained models in the BERT family (Devlin et al. 2019) to acquire distributional generalizations, following in the footsteps of Kim & Smolensky (2021) and showing that such generalizations are indeed within the computational capacity of transformer language models.

There of course remains a great deal of work to be done to further understand and characterize the generalizations present in natural language as they pertain in a way which connects to the learnability of such problems by artificial neural networks. While our work here does propose a typological hierarchy of generalization tasks, it does not endeavor to descriptively characterize some of the more vexing tasks posed to linguistic neural networks today; compositionality, in its full form, remains elusive, and it remains to be seen if we can provide a description of the structural and length generalizations noted in Gordon et al. (2019), along with the kinds of generalizations which require the learning of non-locally-equivariant maps. More generally, a connection between this formal complexity hierarchy and the learnability of these problems by neural networks has not been established. Even for neural networks without any kind of inductive bias for solving generalization tasks, it is not clear why some tasks (like the ℓ_3 anaphora resolution task) are solvable by inattentive recurrent models while others (like length generalization on SCAN, see Lake & Baroni (2018)) seem unsolvable. When considering models *with* inductive biases, like the attention mechanisms of Sutskever, Martens & Hinton (2011) or the G-recurrent models designed by Gordon et al. (2019), the scope of what these biases allow us to solve remains open. Empirical failures of these models to generalize well on tasks like SCAN or COGS seem to indicate that something more is required, but what that is remains unknown (Lake 2019, Gordon et al. 2019, Kim & Linzen 2020).

Regarding the empirical results obtained here, we still do not have a constructive characterization of how the simplest models studied in Chapter 5 manage to solve the anaphora resolution task, although the analytical tools of McCoy et al. (2020) point in a promising direction. Since McCoy et al. (2020) found success at modeling the encoding representations of recurrent models with tensor-product decomposition networks, we hope to employ such networks to provide an account of how our models are representing reflexive anaphora, or more generally how such models learn and represent data in a way consistent with the kinds of generalization observed. More broadly, the connection between the group-equivariant properties described in Gordon et al. (2019) and the tensor-product representation of symbolic structures in neural models (Smolensky 1990) has not yet been established, although the intuitive connection between the filler-role paradigm and group-equivariant functions on input data suggests a fruitful avenue of research. Namely, the roles described in Smolensky (1990) seem to function analogously to the induced equivalence classes described by Gordon et al. (2019), while the fillers seem to capture the information that is preserved under equivariant mappings of the input domain to the output domain.

Finally, we hope to extend our analysis of the distribution generalization observed in BERT models to better understand the robustness of the learned patterns. Firstly, we want to more concretely understand if the observed levels of lexical generalization hold in more complicated syntactic constructions to test the degree to which these large language models have learned human-like generalizations on the basis of large amounts of training data. Secondly, we want to know whether or not the displayed generalizational capabilities arise as a result of learning a shared computational mechanism for syntactic or structural transformations, or whether it arises as a latent property of the learned geometry of input embeddings.

Bibliography

Abdou, Mostafa, Vinit Ravishankar, Maria Barrett, Yonatan Belinkov, Desmond Elliott & Anders Søgaard. 2020. The sensitivity of language models and humans to winograd schema perturbations. *arXiv:2005.01348 [cs]*. URL: http://arxiv.org/abs/2005.01348.

Allen, Carl & Timothy Hospedales. 2019. Analogies explained: towards understanding word embeddings. *arXiv:1901.09813 [cs, stat]*. URL: http://arxiv.org/abs/1901.09813.

Alsina, Alex. 1996. Passive types and the theory of object asymmetries. *Natural Language & Linguistic Theory* 14(4). 673–723. URL: https://www.jstor.org/stable/4047832.

Ambridge, Ben, Caroline F. Rowland & Julian M. Pine. 2008. Is structure dependence an innate constraint? new experimental evidence from children's complex-question production. *Cognitive Science* 32(1). 222–255. DOI: 10.1080/03640210701703766.

Bahdanau, Dzmitry, Kyunghyun Cho & Yoshua Bengio. 2016. *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv:1409.0473 [cs, stat]. URL: http://arxiv.org/abs/1409.0473.

Bat-El, Outi. 2003. Semitic verb structure within a universal perspective. In Joseph Shimron (ed.), *Language processing and acquisition in languages of semitic, root-based, morphology*, 29–59. John Benjamins Publishing Company. URL: https://www.jbe-platform.com/content/books/9789027296689-lald.28.02bat.

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major & Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: can language models be too big? &#x1f99c; in *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (FAccT '21), 610–623. New York, NY, USA: Association for Computing Machinery. DOI: 10.1145/3442188.3445922.

Berent, Iris & Gary Marcus. 2019. No integration without structured representations: response to pater. *Language* 95(1). e75–e86. DOI: 10.1353/lan.2019.0011.

Botvinick, Matthew M. & David C. Plaut. 2006. Short-term memory for serial order: a recurrent neural network model. *Psychological Review* 113(2). 201–233. DOI: 10.1037/0033-295X.113.2.201.

Bresnan, Joan & Lioba Moshi. 1990. Object asymmetries in comparative bantu syntax. *Linguistic Inquiry* 21(2). 147–185. URL: https://www.jstor.org/stable/4178668.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever & Dario Amodei. 2020. Language models are few-shot learners. *arXiv:2005.14165 [cs]*. URL: http://arxiv.org/abs/2005.14165.

Cho, Kyunghyun, B. V. Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk & Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1724–1734. Doha, Qatar: Association for Computational Linguistics. DOI: 10.3115/v1/D14-1179.

Chomsky, Noam. 1971. *Problems of knowledge and freedom*. Pantheon Books. 136 pp.

Chomsky, Noam. 1980. Rules and representations. *Behavioral and Brain Sciences* 3(1). 1–15. DOI: 10.1017/S0140525X00001515.

Clackson, Kaili, Claudia Felser & Harald Clahsen. 2011. Children's processing of reflexives and pronouns in english: evidence from eye-movements during listening. *Journal of Memory and Language* 65(2). 128–144. DOI: 10.1016/j.jml.2011.04.007.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics. DOI: 10.18653/v1/N19-1423.

Dodge, Jesse, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell & Matt Gardner. 2021. Documenting large webtext corpora: a case study on the colossal clean crawled corpus. *arXiv:2104.08758 [cs]*. URL: http://arxiv.org/abs/2104.08758.

Dong, Li & Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, vol. 1, 33–43. Berlin, Germany: Association for Computational Linguistics. DOI: 10.18653/v1/P16-1004.

Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive Science* 14(2). 179–211. DOI: 10.1016/0364-0213(90)90002-E.

Fitch, W. Tecumseh, Marc D. Hauser & Noam Chomsky. 2005. The evolution of the language faculty: clarifications and implications. *Cognition* 97(2). 179–210, discussion 211–225. DOI: 10.1016/j.cognition.2005.02.005.

Fodor, Jerry A. & Zenon W. Pylyshyn. 1988. Connectionism and cognitive architecture: a critical analysis. *Cognition* 28(1). 3–71. DOI: 10.1016/0010-0277(88)90031-5.

Frank, Robert & Donald Mathis. 2007. Transformational networks. In *Abstracts from the 3rd meeting of the workshop*. Nashville, Tennessee: Cognitive Science Society. URL: https://blogs.umass.edu/brain-wars/files/2017/06/cogsci-2007.pdf.

Frank, Robert, Donald Mathis & William Badecker. 2013. The acquisition of anaphora by simple recurrent networks. *Language Acquisition* 20(3). 181–227. DOI: 10.1080/10489223.2013.796950.

Frank, Robert & Jackson Petty. 2020. Sequence-to-sequence networks learn the meaning of reflexive anaphora. In *Proceedings of the third workshop on computational models of reference, anaphora and coreference*, 154–164. Barcelona, Spain (online): Association for Computational Linguistics. URL: https://aclanthology.org/2020.crac-1.16.

Gandhi, Kanishk & Brenden M. Lake. 2020. Mutual exclusivity as a challenge for deep neural networks. *arXiv:1906.10197 [cs]*. URL: http://arxiv.org/abs/1906.10197.

Ghomeshi, Jila, Ray Jackendoff, Nicole Rosen & Kevin Russell. 2004. Contrastive focus reduplication in english (the salad-salad paper). *Natural Language & Linguistic Theory* 22(2). 307–357. URL: https://www.jstor.org/stable/4048061.

Gleitman, Lila. 1990. The structural sources of verb meanings. *Language Acquisition* 1(1). 3–55. URL: https://www.jstor.org/stable/20011341.

Goldberg, Yoav. 2019. Assessing BERT's syntactic abilities. *arXiv:1901.05287 [cs]*. URL: http://arxiv.org/abs/1901.05287.

Gordon, Jonathan, David Lopez-Paz, Marco Baroni & Diane Bouchacourt. 2019. Permutation equivariant models for compositional generalization in language. In URL: https://openreview.net/forum?id=SylVNerFvr.

Goyal, Anirudh & Yoshua Bengio. 2021. Inductive biases for deep learning of higher-level cognition. *arXiv:2011.15091 [cs, stat]*. URL: http://arxiv.org/abs/2011.15091.

Graves, Alex & Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks: The Official Journal of the International Neural Network Society* 18(5). 602–610. DOI: 10.1016/j.neunet.2005.06.042.

Gulordava, Kristina, Piotr Bojanowski, Edouard Grave, Tal Linzen & 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)*, 1195–1205. New

Orleans, Louisiana: Association for Computational Linguistics. DOI: 10.18653/v1/N18-1108.

Hadley, Robert F. 1994. Systematicity in connectionist language learning. *Mind & Language* 9(3). 247–272. DOI: 10.1111/j.1468-0017.1994.tb00225.x.

Hahn, Michael. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics* 8. 156–171. DOI: 10.1162/tacl_a_00306.

Hewitt, John & 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)*, 4129–4138. Minneapolis, Minnesota: Association for Computational Linguistics. DOI: 10.18653/v1/N19-1419.

Holmberg, Anders, Michelle Sheehan & Jenneke van der Wal. 2019. Movement from the double object construction is not fully symmetrical. *Linguistic Inquiry* 50(4). 677–722. DOI: 10.1162/ling_a_00322.

Hu, Jennifer, Jon Gauthier, Peng Qian, Ethan Wilcox & 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*, 1725–1744. Online: Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.158.

Kim, Najoung & 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)*, 9087–9105. Online: Association for Computational Linguistics. DOI: 10.18653/v1/2020.emnlp-main.731.

Kim, Najoung & Paul Smolensky. 2021. Testing for grammatical category abstraction in neural language models. *Proceedings of the Society for Computation in Linguistics* 4(1). 467–470. DOI: https://doi.org/10.7275/2nb8-ag59.

Kim, Yoon, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer & Gábor Melis. 2019. Unsupervised recurrent neural network grammars. 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)*, 1105–1117. Minneapolis, Minnesota: Association for Computational Linguistics. DOI: 10.18653/v1/N19-1114.

Kuncoro, Adhiguna, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark & Phil Blunsom. 2018. 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)*, 1426–1436. Melbourne, Australia: Association for Computational Linguistics. DOI: 10.18653/v1/P18-1132.

Lake, Brenden & Marco Baroni. 2018. Generalization without systematicity: on the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings* *of the 35th international conference on machine learning*, 2873–2882. PMLR. URL: https://proceedings.mlr.press/v80/lake18a.html.

Lake, Brenden M. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in neural information processing systems*, vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/hash/f4d0e2e7fc057a58f7ca4a391f01940a-Abstract.html.

Levy, Roger & Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on language resources and evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2006/pdf/513_pdf.pdf.

Li, Yuanpeng, Liang Zhao, Jianyu Wang & Joel Hestness. 2019. Compositional generalization for primitive substitutions. 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)*, 4293–4302. Hong Kong, China: Association for Computational Linguistics. DOI: 10.18653/v1/D19-1438.

Lin, Yongjie, Yi Chern Tan & Robert Frank. 2019. Open sesame: getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, 241–253. Florence, Italy: Association for Computational Linguistics. DOI: 10.18653/v1/W19-4825.

Linzen, Tal, Emmanuel Dupoux & Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics* 4. 521–535. DOI: 10.1162/tacl_a_00115.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer & Veselin Stoyanov. 2019. RoBERTa: a robustly optimized BERT pretraining approach. *arXiv:1907.11692 [cs]*. URL: http://arxiv.org/abs/1907.11692.

Luong, Thang, Hieu Pham & Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1412–1421. Lisbon, Portugal: Association for Computational Linguistics. DOI: 10.18653/v1/D15-1166.

Marcus, Gary. 2020. The next decade in AI: four steps towards robust artificial intelligence. *arXiv:2002.06177 [cs]*. URL: http://arxiv.org/abs/2002.06177.

Marcus, Gary F. 1998a. Can connectionism save constructivism? *Cognition* 66(2). DOI: 10.1016/s0010-0277(98)00018-3.

Marcus, Gary F. 1998b. Rethinking eliminative connectionism. *Cognitive Psychology* 37(3). 243–282. DOI: 10.1006/cogp.1998.0694.

Marcus, Gary F. 2001. *The algebraic mind: integrating connectionism and cognitive science*. Red. by Lila Gleitman, Susan Carey, Elissa L. Newport & Elizabeth S. Spelke (Learning, Development, and Conceptual Change). Cambridge, MA, USA: A Bradford Book. 240 pp.

Marcus, Mitchell P., Beatrice Santorini & Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics* 19(2). 313–330. URL: https://aclanthology.org/J93-2004.

Marvin, Rebecca & Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 1192–1202. Brussels, Belgium: Association for Computational Linguistics. DOI: 10.18653/v1/D18-1151.

McCoy, R. Thomas, Robert Frank & Tal Linzen. 2020. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics* 8. 125–140. DOI: 10.1162/tacl_a_00304.

McCoy, R. Thomas, Tal Linzen, Ewan Dunbar & Paul Smolensky. 2020. Tensor product decomposition networks: uncovering representations of structure learned by neural networks. In *Proceedings of the society for computation in linguistics 2020*, 277–278. New York, New York: Association for Computational Linguistics. URL: https://aclanthology.org/2020.scil-1.34.

McGinnis, M. 2002. Object asymmetries in a phase theory of syntax. In *Proceedings of the 2001 CLA annual conference*, 133–144. PRISM. DOI: 10.11575/PRISM/10006.

Merrill, William. 2019. Sequential neural networks as automata. In *Proceedings of the workshop on deep learning and formal languages: building bridges*, 1–13. Florence: Association for Computational Linguistics. DOI: 10.18653/v1/W19-3901.

Merrill, William, Lenny Khazan, Noah Amsel, Yiding Hao, Simon Mendelsohn & Robert Frank. 2019. Finding hierarchical structure in neural stacks using unsupervised parsing. In *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, 224–232. Florence, Italy: Association for Computational Linguistics. DOI: 10.18653/v1/W19-4823.

Merrill, William, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith & Eran Yahav. 2020. A formal hierarchy of RNN architectures. *arXiv:2004.08500 [cs]*. URL: http://arxiv.org/abs/2004.08500.

Mikolov, Tomas, Kai Chen, Greg Corrado & Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781 [cs]*. URL: http://arxiv.org/abs/1301.3781.

Min, Junghyun, R. Thomas McCoy, Dipanjan Das, Emily Pitler & Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2339–2352. Online: Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.212.

Mulligan, Karl, Robert Frank & Tal Linzen. 2021. Structure here, bias there: hierarchical generalization by jointly learning syntactic transformations. *Proceedings of the Society for Computation in Linguistics* 4(1). 125–135. DOI: 10.7275/j0es-xf97.

O'Grady, William. 2013. Reflexive pronouns in second language acquisition. *Second Langauge*.

Perfors, Amy, Joshua B. Tenenbaum & Terry Regier. 2011. The learnability of abstract syntactic principles. *Cognition* 118(3). 306–338. DOI: 10.1016/j.cognition.2010.11.001.

Pinker, Steven. 1989. *Learnability and cognition: the acquisition of argument structure*. Red. by Lila Gleitman, Susan Carey, Elissa L. Newport & Elizabeth S. Spelke (Learning, Development, and Conceptual Change). Cambridge, MA, USA: A Bradford Book. 432 pp.

Pullum, Geoffrey K. & Barbara C. Scholz. 2002. Empirical assessment of stimulus poverty arguments. *The Linguistic Review* 19. 9–50. URL: https://www.semanticscholar.org/paper/Empirical-assessment-of-stimulus-poverty-arguments-Pullum-Scholz/dc6286fe4cced25c99026f85e01e1b9d3dff36ae.

Resnik, P. 1996. Selectional constraints: an information-theoretic model and its computational realization. *Cognition* 61(1). 127–159. DOI: 10.1016/s0010-0277(96)00722-6.

Rogers, Anna, Olga Kovaleva & Anna Rumshisky. 2020. A primer in BERTology: what we know about how BERT works. *Transactions of the Association for Computational Linguistics* 8. 842–866. DOI: 10.1162/tacl_a_00349.

Safir, Ken. 2013. Syntax, binding, and patterns of anaphora. In Marcel den Dikken (ed.), *Cambridge handbook of generative syntax, the*, 515–576. Cambridge: Cambridge University Press. DOI: 10.1017/CBO9780511804571.020.

Sanh, Victor, Lysandre Debut, Julien Chaumond & Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of the 5th workshop on energy efficient machine learning and cognitive computing*. URL: http://arxiv.org/abs/1910.01108.

van Schijndel, Marten, Aaron Mueller & Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. 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)*, 5831–5837. Hong Kong, China: Association for Computational Linguistics. DOI: 10.18653/v1/D19-1592.

Schlag, Imanol, Paul Smolensky, Roland Fernandez, Nebojsa Jojic, Jürgen Schmidhuber & Jianfeng Gao. 2020. Enhancing the transformer with explicit relational encoding for math problem solving. *arXiv:1910.06611 [cs, stat]*. URL: http://arxiv.org/abs/1910.06611.

Schmidhuber, J H & R Huber. 1990. *Learning to generate focus trajectories for attentive vision*. FKI-128-90. Institut für Informatik: Technische Universität München. 18.

Shen, Yikang, Shawn Tan, Alessandro Sordoni & Aaron Courville. 2018. Ordered neurons: integrating tree structures into recurrent neural networks. In *Proceedings of the 7th international conference on learning representations*. URL: https://openreview.net/forum?id=B1l6qiR5F7.

Smolensky, Paul. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence* 46(1). 159–216. DOI: 10.1016/0004-3702(90)90007-M.

Smolensky, Paul, Moontae Lee, Xiaodong He, Wen-tau Yih, Jianfeng Gao & Li Deng. 2016. Basic reasoning with tensor product representations. *arXiv:1601.02745 [cs]*. URL: http://arxiv.org/abs/1601.02745.

Storoshenko, Dennis. 2008. The distribution of reflexive pronouns in english: a corpus analysis. In *Proceedings of the 24th northwest linguistics conference*, 67–74.

Sutskever, Ilya, James Martens & Geoffrey Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning*, 8. Bellevue, WA.

Sutskever, Ilya, Oriol Vinyals & Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, vol. 27. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html.

Tomasello, Michael. 1992. The social bases of language acquisition. *Social Development* 1(1). 67–87. DOI: 10.1111/j.1467-9507.1992.tb00135.x.

Urbanczyk, Suzanne. 2017. *Phonological and Morphological Aspects of Reduplication*. In *Oxford Research Encyclopedia of Linguistics*. DOI: 10.1093/acrefore/9780199384655.013.80.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser & Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st international conference on neural information processing systems* (NIPS'17), 6000–6010. Red Hook, NY, USA: Curran Associates Inc.

Walther, Géraldine & Benoît Sagot. 2017. Speeding up corpus development for linguistic research: language documentation and acquisition in romansh tuatschin. In *Proceedings of the joint SIGHUM workshop on computational linguistics for cultural heritage, social sciences, humanities and literature*, 89–94. Vancouver, Canada: Association for Computational Linguistics. DOI: 10.18653/v1/W17-2212.

Warstadt, Alex & Samuel R. Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? In *Proceedings of the 42nd annual meeting of the cognitive science society*, 1737–1743. URL: https://arxiv.org/abs/2007.06761v2.

Warstadt, Alex, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic & Samuel R. Bowman. 2019. Investigating BERT's knowledge of language: five analysis methods with NPIs. 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)*, 2877–2887. Hong Kong, China: Association for Computational Linguistics. DOI: 10.18653/v1/D19-1286.

Weißenhorn, Pia, Yuekun Yao, Lucia Donatelli & Alexander Koller. 2022. Compositional generalization requires compositional parsers. *arXiv:2202.11937 [cs]*. URL: http://arxiv.org/abs/2202.11937.

Woolford, Ellen. 1993. Symmetric and asymmetric passives. *Natural Language and Linguistic Theory* 11. 679–728. DOI: 10.1007/BF00993017.