THE UNIVERSITY *of* EDINBURGH

Information Structure in Mappings: An Approach to Learning, Representation, and Generalisation

Henry Coxe Conklin

Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
The University of Edinburgh
2025

Abstract

Mappings relate two different spaces, transforming things of one kind into another; they are ubiquitous across the sciences and the world around us. Mathematical functions map between a domain and range, digital phone systems map waveforms to binaries, ribosomes map DNA sequences to proteins as part of a larger mapping between genotypes and phenotypes. Telegram operators map back and forth between text and morse code, artificial neural networks map inputs to vector representations, and language allows us to map our thoughts to sentences that express them. The structure of these mappings differs widely, having conformed either to the selection pressures of their environment or the concerns of their architects.

Despite the remarkable success of large large-scale neural networks in recent years, we still lack unified notation for thinking about and describing their representational spaces. We lack methods to reliably describe how their representations are structured, how that structure emerges over training, and what kinds of structures are desirable. This thesis introduces quantitative methods for identifying systematic structure in mappings between spaces, and leverages them to understand how deep-learning models learn to represent information, what representational structures drive generalisation, and how design decisions condition the structures that emerge. To do this I identify basic kinds of system-level structures present in a mapping, along with information theoretic quantifications of each of them. I use these to analyse learning, structure, and generalisation across multi-agent reinforcement learning models, sequence-to-sequence models trained on a single task, models trained with meta-learning objectives, and Large Language Models. I also introduce a novel, performant, approach to estimating the entropy of vector space, that allows this analysis to be applied to models ranging in size from 1 million to 12 billion parameters.

The experiments here work to shed light on how large-scale distributed models of cognition learn, while allowing us to draw parallels between those systems and their human analogs. They show how the structures of language and the constraints that give rise to them in many ways parallel the kinds of structures that drive performance of contemporary neural networks.

Lay summary

The world is made up of systems that convert one type of information into another - much like a translator changes words from one language to another. Your phone turns your voice into digital signals, your body turns genetic code into physical traits, and your brain turns thoughts into spoken words. These transformations can be found everywhere in nature and technology, each shaped by different needs and purposes.

In recent years, artificial intelligence systems called neural networks have become incredibly powerful at processing information, but we don't have ways to understand how they organise and structure this information internally. In part because they represent it as huge lists of numbers, that we as humans have trouble reasoning about. This research develops new mathematical tools to look inside these AI systems and understand how they learn to represent information.

These tools work by looking for structure in the relationship between what we show the AI system, and the numbers it transforms them to. The tools look for the same kinds of structure we see in the way human language transforms our thoughts into the things we say.

By applying these tools to various AI systems - from those that play games together to those that process language - this work reveals patterns in how these systems organise information and which patterns help them solve new problems. The research also introduces a new method to measure how efficiently these systems store information, which works on both small and enormous AI models.

Apart from helping us better understand artificial intelligence, these findings also show parallels between how AI systems and human language structures information. This suggests there may be some universal principles in how both natural and artificial systems learn to represent information effectively.

Acknowledgements

I have not done this of my own accord, and so have people in need of thanks.

Kenny Smith my primary supervisor, to whom I owe some substantial debt of gratitude. Thank you for agreeing to supervise my undergraduate dissertation in 2018, and for suggesting I apply for a PhD position a year later, and for making the time to meet with me for an hour every week for the past 5 years. You taught me how to talk about research, and how to pin the big picture down to something testable. I would not have pursued any of this were it not for you, so thank you. Ivan Titov (my secondary supervisor), thank you for encouraging my interest in information theory, for always welcoming me into your group meetings, for introducing me to Bailin, and for supervising our work together. Paul Smolensky (my internship supervisor), thank you for teaching me the elegance of an outer product.

To my PhD Friends; George Carter for their appreciation of Edinburgh's lesser known parks and sad smudges. Tom Hosking, for being such an excellent discussion partner. Seraphina Goldfarb-Tarrant, for the time spent by any and all fires. Annie Holtz, for your appreciation of bakeries, coffee, and bad contemporary art. Laurie Burchell for the many laps of Inverleith park. Rohit Saxena, for not panicking as I cut across 8 lanes of traffic. Matthias Lindemann, for putting up with questions about grammars and linear algebra. Bailin Wang for putting up with questions about meta learning and tensor2struct. Verna Dankers, for putting up with questions about interpretability and for giving excellent feedback. Stella Frank, for helping me get started. Shira and Tomer, for dinner. I'd also like to thank a number of other people for making the PhD time what it was; Marc Meisezahl, Vlad Nedelcu, Elizabeth Pankratz, Aislinn Keogh, Maisy Hallam, Juan Guerrero Montero, Lauren Fletcher, Irene Winther, Dan Wells, Paul Soulos, Anna Kapron-King, Tamar Johnson, Marianne de heer Kloots, Nik

My Edinburgh family; thank you Abby Jackson for the film nights, and politics — and Roddy McDermott for the tunes. Mel Philips and Craig Methven, thank you for the art, and the art of the pal smash. Anna Stewart and Marty McLennan, thank you for fringing. Celia Dugua for the aesthetics — Roxy Cook for dinner. Pedro Leandro and Macleod Stephen for discussion. Izzy Moulder and Caz Elms for the fortress. Eric and Josie Geistfeld for scroobin.

Amy Sheahan for seeing friendship as the serious business it is. Kat Knoerl, go team.

And to my family; thank you Dana Catharine and Susan Buckley for guiding me through the seas of moral turpitude. James Kirby Rogers, thank you for introducing me to Swensen's, contemporary dance, and the Roy St. Parking structure. AKC, for your relentless enthusiasm for my continued existence. PAK & PMC — none of this would have happened without you.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Henry Coxe Conklin)*

for paul.

and everyone else who left the party early

Table of Contents

Chapter 1

How to Represent Information

in Mappings, Language, & Artificial Neural Networks

[redacted] to count leaves is not less meaningful than to count the stars [redacted] It would help them to know whether the world is finite. I discovered one tree that is finite. [redacted]

-David Ignatow

Mappings relate two different spaces, transforming things of one kind into another; they are ubiquitous across the sciences and the world around us. Mathematical functions map between a domain and range, digital phone systems map waveforms to binaries, ribosomes map DNA sequences to proteins as part of a larger mapping between genotypes and phenotypes. Telegram operators map back and forth between text and morse code, artificial neural networks map inputs to vector representations, and language allows us to map our thoughts to sentences that express them.

In each case, some information from one space needs to be transferred to another. Morse code represents 26 letters and 10 digits as a sequence of binary values - dots or dashes; the spanish alphabet uses 27 letters to encode the 19 consonant phonemes used in the language[1]. While in simple cases like these the scope of what information needs to be represented is well defined, what happens

[1] the exact number of phonemes being dependent on dialect

when what we need to encode is much larger: like all of the text on the internet combined with the text of every book written in the past 100 years? As what we need to describe increases in complexity, it becomes less and less clear how to map it to a representation which preserves the structure of the original. How do complex mappings represent information - and are there structural properties that are shared across representational systems that do this effectively?

1.1 Deep Learning

This question is of particular importance when it comes to artificial neural networks[2]. These are models trained to map their inputs to high-dimensional vector representations which preserve enough relevant information from the input for the model to succeed at a task. This is difficult in and of itself, but making it more challenging is that these models are usually trained on only a subset of the space they will need to encode - like a sample of sentences, rather than every sentence possible in a language[3] - making it difficult to know what information in the data they see is part of larger generalisations across the unattested space. Depending on the task this can mean learning to drive on any road - mapping video input to actions like turn/accelerate/brake - from video footage of only a few thousand (Yurtsever et al., 2020), or learning to map any sentence in French to English having been trained on a selection of websites and news articles (Kalchbrenner & Blunsom, 2013).

Despite their success in recent years, large-scale neural networks often fail to learn a mapping which generalises systematically - often failing to learn a representation of their training data that can generalise far outside it. This becomes clear when models are evaluated on a different data distribution than the one they were trained on. In linguistic tasks this can mean their performance degrades substantially when words they have seen before appear in novel contexts, or appear in sentences longer than they see during training (e.g. Keysers et al., 2020; Kim & Linzen, 2020; Lake & Baroni, 2018). Vision models can struggle with tiny changes to a small subset of pixels which doesn't change the image to the human eye (Goodfellow et al., 2014) or have difficulty identifying a horse standing

[2]Throughout I use the terms artificial neural networks, neural networks, deep-learning, and connectionist models broadly interchangeably.

[3]Given language contains a functionally infinite number of possible sentences, training on all of them is intractable.

on anything other than grass, given the frequency of pastured-horses in training data (Dagaev et al., 2021).

In response data scaling & augmentation has become the prevailing strategy - making a model's training data sufficiently large that it is unlikely to encounter something it hasn't seen before - reducing the amount that it needs to generalise. But models struggle even when trained on more data than a human hears in 200 lifetimes (Furrer et al., 2021; Griffiths, 2020b). A dataset can never cover the entire space of possible examples; there are an infinite number of grammatical sentences that have yet to be said (Chomsky, 1969) — ultimately data underspecifies for the generalisations that produced it (Goodman, 1955). Despite this, humans reliably learn their native language from a small fraction of the data shown to a large language model. What is missing from the representations learned by models trained orders of magnitude more data than we are, that affords them some generalisation but no more?

Building models that generalise robustly out-of-distribution remains a core goal of machine learning (Bishop, 2006), despite this we continue to have a limited understanding of what kinds of representational structures are needed enable generalisation. Some work identifies the kinds of shallow heuristics models learn instead of the underlying structure of the data (McCoy, 2019), but these are rarely *intrinsic* measures — they're not based on models' internal representations but rather their (*extrinsic*) downstream performance. It proves challenging to relate representational properties to behaviours, in fact in some contexts it's been shown that no existing intrinsic measures can predict model behaviour (Goldfarb-Tarrant et al., 2021).

1.2 The Problem of Interpretability

The challenge of understanding structure in deep-learning models is driven in no small part by their scale. The past two decades have seen a striking shift in the tractability of training neural architectures with gradient descent. Early models performed digit recognition with 9760 trainable parameters (LeCun et al., 1989), or learned sentence dependency structures with only 50 (Elman, 1990). By contrast the 'small' model used in chapter 3 of this thesis uses just over 1,000,000 parameters, and the large language models used in chapter 5 have 12,000,000,000 (even these are comparatively small by current standards - with state-of-the-art

LLMs exceeding 400 billion parameters Dubey et al. (2024)). As they scale up these models represent information as increasingly high-dimensional vectors, something about which humans tend not to have strong intuitions. This makes it hard for us to decipher or reason about how a model represents information, how it learns to do so, or predict which representations might be best.

We lack a clear understanding of what kinds of representational structures are desirable, or if there are domain-general, quantifiable properties of a representational system that enable systematic generalisation.

1.3 How to Interpret Representation Spaces

This lays out a core set of problems -

- Models fail to represent their training data in a way that allows them to generalise systematically
- Their representations are high-dimensional vectors that are hard for us to interpret
- We lack a framework for defining how those representations are structured that lets us understand what kinds of structures drive behaviours like generalisation.

The remainder of this thesis works to address these issues, introducing a framework for thinking about representation spaces grounded in existing work in cognitive science and information theory. First though, it is worth considering what it means to interpret a representation space, and by extension what any successful approach should do. I break this question into two parts, **what phenomena an approach to interpretability should give an account for**, and **the properties that approach should have.** At a minimum a framework for interpreting deep-learning models needs to be able to give an account of

- how representations are structured
- how representations change over the course of training
- how different design decisions (e.g. hidden size, choice of optimiser, learning rate, dropout ...) affect representation space

- what kinds of representation structures generalise best

These requirements already constrain some properties our desired approach can have. To characterise representation structure an approach needs to deal with representations directly, rather than making inferences about them based on downstream performance. Additionally to give an account over the timecourse of training an approach ideally needs to be sufficiently fast and resource efficient that it can be run at each training step, rather than only once as a post-hoc analysis. More than that, it is worth remembering that interpretability has an audience: humans. As such it is not enough to provide just any account of the above phenomena, it needs to be an account that is intuitive for us - relating the structures found here to things we already have an understanding of, like existing work on representations in other areas of science. In summary, our desiderada for an approach to interpretability are that it:

- deals directly with representations rather than their downstream effects
- is efficient enough to leverage throughout training
- accounts for representational structure in models in a way that can be clearly related to work on representations in existing areas of science.

While existing approaches to interpretability have shed light on a variety of empirical phenomena in deep-learning models, they often meet only some of the criteria laid out above. A prominent existing set of approaches leverages behavioural evidence, treating models as akin to psycholinguistic subjects (Futrell et al., 2018, 2019). By treating model outputs as behaviours, experiments enable conclusions about what kinds of information a model may have learned. Like looking at whether models assign higher probability to grammatical sentences, to reason about whether their representations encode syntactic information (Hu et al., 2020; Marvin & Linzen, 2018; Tucker et al., 2022; Warstadt et al., 2019). While valuable, this line of work is removed from the models' representations themselves - characterising downstream behaviours rather than characterising the representational structures that drive them.

Probing represents another form of interpretability with closer ties to representational structure (Hupkes et al., 2018; Müller-Eberstein et al., 2023; Pimentel et al., 2020). It relies on training a probe — a smaller model, like a linear classifier

— to predict certain properties from representations. If a model can take a representation for a sentence, and predict the correct part of speech labels, or constituency parses for that sentence, it acts as some evidence that that information is encoded in those representations (e.g. Voita & Titov, 2020). Although this, again, does not directly characterise the structures in representation space but the information that can be predicted from them - and in some cases how complex a classifier is required for that prediction. Additionally as it requires training a secondary model, its computational complexity can limit the contexts where it is applied.

Mechanistic interpretability (Elhage et al., 2021), tries to offer explanations more tightly tied to what happens model internally. However it often relies on training unsupervised probes (termed *sparse auto encoders* Elhage et al., 2022). This enables some analysis of which parts of a model correspond to different words or concepts from the training data (Bricken et al., 2023), but again this relies on training a secondary model, meaning it can have similar compute cost to other forms of probing and gives a limited understanding of how representations are structured or how those structures relate to work on representations in other areas of science.

Having discussed what an approach to interpretability needs, and the limited ways in which this is addressed by existing work, I now introduce the approach taken here. This thesis introduces quantitative methods for identifying systematic structure in mappings between discrete and continuous spaces, and leverages them to interpret how neural networks learn and when & why they generalise successfully. These methods are predictive of downstream performance, grounded in information theory, and fast to compute enabling analysis of even large models throughout training.

1.4 An Analogical Approach

As stated above, a goal here is to develop an approach to interpretability that lets us leverage intuitions from other disciplines. In particular disciplines with existing work on what representation structures are likely to be learnable, expressive, and can enable generalisation. To do this we can start by identifying a representational mapping, which is well studied, and which bears some resemblance to representation systems learned by the models studied here. By defining measures of structure applicable to both we can draw analogies between the exemplar and

the structures that emerge in a deep learning model. This kind of analogical interpretability helps us understand something novel, through how it relates to something similar that is well understood.

In our case an ideal exemplar would have examples of structures that enable the kinds of sample efficient learning and generalisation which neural networks likely need to succeed in the general case. It would also be one about which we have strong intuitions for what kinds of structures are desirable - unlike high-dimensional vector spaces - and for which there is an substantive body of work analysing & describing those structures. Given these desiderata, the obvious choice is natural language.

At its core language is a mapping - relating objects, concepts, and events, to words, constructions, and phrases which refer to them (de Saussure, 1916). While many natural communication systems fit this bill, language is unique amongst them (Hockett, 1960). It's acquired, rather than being built-in from birth. It generalises readily to novel concepts and contexts, instead of containing a finite repertoire of calls - we can readily interpret sentences we have never heard before[4]. Its units are meaningful despite being arbitrary, with its systematic structure providing us a representation system simple enough to be learned by children, but complex enough to describe the universe.

A growing body of work presents an account of how structures in language may result from language evolving to conform to domain-general cognitive constraints, and the dynamics of transmission and use rather than reflecting properties of some innate language faculty (e.g. Brighton et al., 2005; Chater et al., 2009; Christiansen, 1994; Culbertson & Kirby, 2016; Fedorenko, 2014; Kirby, 2001; Kirby et al., 2008; Kirby et al., 2014, 2015; Smith, 2011; Smith & Kirby, 2008; Wehbe et al., 2021; Zuidema, 2002). Given this, the structures present in language, and how we think they originated, may have explanatory power domain-generally, giving us an exemplar of how mappings become structured in response to their environment.

We expect neural networks to learn a mapping from inputs to representations from a finite sample of data, and generalise to examples not seen during training, by learning to encode structural properties of the world from which its training data is drawn rather than relying on heuristics. These expectations are in parallel to design features of language, which suggests if we develop sufficiently general ways

[4]e.g. "At the airport I smiled myself an upgrade" —(Goldberg, 2006)

of quantifying structures that underpin language we may be able to assess whether or not those structures are present in other domains - like vector spaces internal to models. While human language is clearly distinct from the representations inside a deep-learning model, their teleological similarities - and arguments about the domain-generality of language - make it a reasonable exemplar for our approach to interpretability. Building an understanding of deep-learning representations by drawing analogies between their structures and the structure of natural language.

1.5 For Our Purposes, What is Structure?

If we're going to look for structure in mappings, and draw analogies with language in the process, we need to be clear what we mean by structure. Part of what makes language unique as a mapping is its systematic structure (sometimes referred to as systematicity). Structure that exists not just at the item level - like at the level of an individual word or sentence - but across the entire language. In this thesis we consider two inter-related notions of structure present in language: compositionality and regularity.

1.5.1 Compositionality

Compositionality describes how language builds the meaning of a whole as a product of the meaning of its parts (Cann, 1993; Chomsky, 1969; Hockett, 1960; Partee et al., 1995). It allows us as speakers and learners to make 'infinite use of finite means' (Von Humboldt, 1863), in that from a finite set of words and constructions we can generate or understand a potentially infinite number of sentences. By composing together known morphemes, words, or sentences in novel combinations we can produce new words, sentences, and paragraphs, where the meaning of the larger construction is a predictable function of the meaning of the parts and they way they are combined. Compositionality represents a core building block of the syntactic system, as exemplified by the minimalist programme, which boils the innate component of syntactic knowledge down to primarily a merge operation [5] which takes two arguments and composes them (Chomsky, 1995, 2014). The origins of compositionality in language have been the subject of study in linguistics (e.g. Bickerton, 1984; Kirby, 2001; Kirby et al., 2008;

[5] Many instantiations of minimalism also rely on a slightly broader assortment of operations, like agreement, and transfer.

Kirby et al., 2015), evolutionary biology (Nowak et al., 2000) and more recently in mutli-agent reinforcement learning (e.g. Chaabouni et al., 2020; Lazaridou et al., 2018; Resnick et al., 2020).

in deep learning

A major driver of the continued interest in compositionality across disciplines is how essential it seems to generalisation; if we want a system to generalise, it's difficult to conceive of how it could do so non-compositionally[6]. How can we understand something novel, which we haven't encountered before, except by breaking it into parts we already know? Whether or not neural networks are capable of learning compositional representations, and generalising compositionality is the subject of longstanding debate, most notably with Fodor and Pylyshyn (1988) arguing that artificial neural networks are structurally unable to compose their representations. More than that, Fodor (1975) argues that human thought is compositional & symbolic to its core - neural networks operate in continuous vector space making symbolic processing impossible (see Symons & Calvo, 2014, for discussion). This criticism is oft repeated, even now, despite the fact that in the years immediately following, Smolensky (1990) showed how compositional, symbolic structures can be losslessly embedded in vector space, Elman (1990) introduced the recurrent neural network which has explicit composition, and Chalmers (1993) pointed out structural flaws in Fodor and Pyshlyn's argument.

Today, a substantive body of work still questions the compositional abilities of contemporary neural networks, and large language models (Akyürek & Andreas, 2022; Andreas, 2020; Csordás et al., 2021; Hupkes et al., 2019; Keysers et al., 2020), with many introducing benchmarking datasets (e.g. Kim & Linzen, 2020; Lake & Baroni, 2018) intended to evaluate models' compositional abilities, by testing them under *distributional shift*, where a model is trained on data sampled from one distribution, and evaluated on data sampled from another. In practice what it means for distributions to differ can depend on the domain, but on text-based tasks data is usually synthetic, generated by a context free grammar, with training and evaluation splits created by subsampling different portions of the entire space that grammar covers. As a result any evaluation split would be

[6]There's an argument to be made that some generalisation may be iconic instead, but such systems are unlikely to empower generalisation at the same order of magnitude as compositionality - that is enable generalisation to tens of thousands of unseen examples rather than a handful.

trivial for the model had it learned the underlying grammar used to generate the data. In reality contemporary architectures like LSTMs and Transformers perform well when training and evaluation splits are sampled at random (referred to as in-distribution generalisation, or independent and identically distributed *i.i.d.*), but where this is not the case - as when evaluation contains longer sentences as mentioned above- those same architectures perform remarkably poorly. Large language models, like T5 (Raffel et al., 2019), pretrained on vast amounts of data then finetuned on these tasks fare substantively better than standard architectures but still below ceiling (Furrer et al., 2021).

if not compositional then what?

This work often paints a confusing picture, asserting that a model's limited ability to generalise out of distribution, to examples generated by the same underlying grammar as the training data, provides evidence of non-compositionality; evidence that models have in fact induced heuristics via statistical learning rather than inducing the rules of the grammar. But this fails to reckon with the fact that these models *can* generalise, to tens of thousands of examples provided those examples and the training data were sampled i.i.d.. Short of an explanation for how generalisation of that scale can happen non-compositionality we instead need an explanation for how a compositional system can generalise systematically to some examples but not others.

1.5.2 Regularity (& Variation)

Unlike compositionality, regularity is a property of language that has received less attention outside of linguistics. Compositionality enables predictable mappings between meanings and forms by building wholes out of reusable parts - like words - which makes it so that human's best friend will reliably be referred to as a *dog* regardless of the context. However language is also rich with *variation* (Weinreich et al., 1968), affording us as speakers an enormous variety of ways to express ourselves dependent on the given context. Regularity - in the form of predictable system-level structures - underpins language's generalisability, but is interwoven with *variation* which gives us robust tools for conveying meaning, ambiguity, & intention in context - making ourselves clear with precision. It allows our collective best friend to sometimes be a *dog*, but also to be a *canine, pup, good*

boy, good boi or *mutt* as the situation demands. In language regularity refers to how predictable realisations of the same property are across a system. This is the inverse of variation with respect to a given property which, somewhat intuitively, describes how much that property varies A substantive body of work looks at how humans regularise their input during learning (Hudson Kam & Chang, 2009; Senghas et al., 1997), and how languages undergo regularisation over time (Reali and Griffiths (2009), Smith and Wonnacott (2010), see Ferdinand et al. (2019) for review) often quantifying it probabilistically. For our purposes we'll say that:

Definitions 1 & 2

Regularity: How predictable realisations of the same property are across a system

Variation: How much realisations of the same property vary in that system dependent on context

---

At a computational level compositionality is a binary property - either a merge operation is performed somewhere in a system or it isn't [7]. I argue in chapter 3, that regularity and variation are quantities best suited to assessing whether a system is structured, in no small part because they are naturally graded - offering degrees of variation - rather than binary. Although the two concepts are intrinsically related, I point out that what is often discussed as compositionality is in reality compositionality + maximal regularity[8] likely predicated on an assumption that variation impairs generalisation.

Variation is not an accident. It proves ubiquitous across languages because it enables us to express effectively the kinds of complex context-dependent information we encounter every day. In extremity variation can impair generalisation - imagine having a different word for dog depending on what it holds in its mouth, like a bone or a tennis ball. This would allow us to be maximally efficient and

[7] Martins and Boeckx (2019) make a case that the binary analogy may not extend to the hardware level, but I leave arguments about the origins of merge to other work, and here for the sake of argument consider systems where merge exists.

[8] You can also think of it as regularity reflecting the predictability or frequency of merge operations in a system, which is likely what is implied when work discusses the 'degree of compositionality'. However the predictability of compositions is distinct from their existence.

precise in the rare context of picking one out of a lineup of dogs each holding a different object. But becomes problematic in the far more likely scenario that we encounter a dog holding something novel, like a smartphone, or the collected works of David Foster Wallace.

While we could describe a word without separate parts referring to [dog] and [what is held] as non-compositional, this elides the fact that the resulting word can go on to be used compositionally in a sentence — *a lack of compositionality at the item-level does not preclude it at the system-level.* Similarly a set of synonyms (e.g. best, baddest, leading, terrific) are often best suited to subtly different contexts (best: a blogpost about what coffee maker to buy; baddest: a TikTok about the best coffee maker to buy; leading: the description of the coffee maker on the manufacturer's website; terrific: the review you leave of the coffee maker). The fact that each of these conveys both a notion of 'excellence' along with contextual information doesn't undermine the compositionality of language. Were we to stipulate that a truly compositional system represent everything context independently the result would be substantively removed from the realities of human language. In chapter 2 I introduce methods for quantifying regularity, variation, and disentanglement (a particular kind of variation), that can in principle be applied to any discrete-discrete or discrete-continuous mapping that I apply in the remainder of the thesis to a variety of artificial neural network models.

1.5.3 Information Structure

Across domains information theory (Shannon, 1949) is a tool of choice for analysing how information is packaged and mapped - finding explanatory power from genetics (Schneider, 2010; Vinga, 2014), to cognitive science (Chater & Vitányi, 2003; Smith & Wonnacott, 2010) and machine learning (MacKay, 2004). Additionally as a discipline it rests on a similar analogy to language as the one I make here, with Shannon introducing the field as a mathematical model of communication. In the general case Information Theory considers the mapping between a message from an information source and a signal that represents it, and presents quantitative methods for describing the relationships between spaces and the mapping that relates them. In this thesis I build on this analogy using basic information theoretic quantities to quantify regularity, variation, and disentanglement in a mapping between spaces. These linguistic concepts are intuitively related to basic

information theoretic quantities, links I make explicit in chapter 2.

There are three basic kinds of structure I consider in a mapping between two spaces: one-to-one, one-to-many, and many-to-one — related to regularity, variation, and disentanglement respectively. In reality, at a system level, a mapping can be comprised of a combination of these three structures, so we quantify the prevalence of each of them probabilistically; defining quantitative measures reflecting the probability of each basic structure across a system. An approach first introduced in chapter 2, then built on in chapters 4 and 5. To distinguish this approach to understanding representational structure from previous work I refer to it as *information structure*[9], given it aims to quantify structure in the way information is mapped between spaces. It's worth noting that this approach is not the first to emphasise the interplay between structure and probability with usage-based approaches to language (e.g. Croft, 2001; Goldberg, 1995; Tomasello, 2005), eroding the binary distinction between grammar and lexicon in favour of constructions that unify meaning and form and are learned probabilistically on the basis of experience (Goldberg, 2003). The relationship between usage-based approaches and an information structure approach are discussed further in chapter 6.

1.6 Leveraging Work on Language to Understand Mappings

We've discussed how making analogies with language can help us form a strong intuition about what a structured, learnable, generalising mapping looks like - formalising these intuitions quantitively is a core goal of this thesis. But by drawing this analogy we can also leverage intuitions from the cognitive sciences about the conditions needed for structure to emerge in a mapping, what kinds of structures can drive or hinder generalisation, and the constraints or pressures which condition the prevalence of those structures. Building on existing theories and intuitions is a major advantage of contextualising mappings found in neural networks in terms of existing areas of science - like language and information theory - rather than approaching them with methods and terminology which make

[9]This is unrelated to exiting linguistic notions of information structure, which focus on different ways of communicating the same information (e.g. to draw focus to a particular part). The name is adopted here to describe our information-theoretic approach to structure.

them out to be something wholly alien.

When Structure Emerges Structural regularities can emerge as a result of the meaning space speakers need to describe expanding such that substantively greater fitness is given to systems with structural regularity (Nowak et al., 2000). Or they can arise from repeated iterated chains of learning, applying a pressure for simplicity making the system easier to learn (Kirby, 2001; Kirby et al., 2008).

What Structures are Desirable Compositionality is essential for generalisation (Cann, 1993; Chomsky, 1965), enabling predictable, regular structures that are recombinable. Variation is equally essential to enable expressivity, giving speakers sufficient fidelity to describe what they need to (Kirby et al., 2015). Other structures, like homonymy, make a system more compressible but at the expense of introducing ambiguities that can be difficult interpret (Piantadosi et al., 2012). Often languages mitigate these ambiguities by collapsing over concepts that are contextually mutually exclusive and unlikely to co-occur (Winters et al., 2018).

What Conditions Structure As mentioned above, pressures relating to the needs of learners and speakers have major effects, with speakers introducing pressure for variation to express themselves, and learners introducing pressure for regularity to aid acquisition (Kirby et al., 2015). These pressures can also be introduced to the system via population dynamics, with prevalence of second language learners, number of speakers, and geographic spread of a population having potentially regularising effects (Dale & Lupyan, 2012; Lupyan & Dale, 2010). Competing needs of speakers and listeners can drive the mapping from meanings to forms to become more efficient in an information theoretic sense, with languages often evolving to be optimally compressed for the amount of information they encode (Kemp et al., 2018; Zaslavsky et al., 2018). More general cognitive constraints are also thought to be a major driver of regularity, with limitations like our finite memory having a regularising effect by placing an upper bound on the amount of variation we can faithfully remember and reproduce (Griffiths, 2020b; Lieder & Griffiths, 2020).

1.7 Capacity's Role in Shaping Structure

Across chapters of this thesis particular attention is paid to the role capacity plays in how structure develops. A wide array of work has looked at how the finite memory of human learners can drive the kinds of regularities ubiquitous across languages (e.g. Ferdinand et al., 2019; Griffiths, 2020b; Newport, 1990; Smith & Wonnacott, 2010). In part because we are likely limited in the number of low probability forms we can recall (Hudson Kam & Newport, 2005). In work on humans however, it can to be difficult to directly modulate the capacity of learners as an independent variable. In models, by contrast, this is a hyper-parameter we set.

Each experimental chapter looks at the effect capacity has on measures of representational structure. Chapter 3, varies the size of representational spaces[10] learned by agents in an emergent communication model, showing populations with smaller agents develop more regular systems. Chapters 4 and 5 look at continuous representations inside models ranging from 1 million parameters to 12 billion — finding larger models can accommodate more contextual variation. Finally Chapter 6 introduces a way to manipulate a model's capacity via optimisation, using a meta-learning objective instead of manipulating a model's parameter count. Models trained with this objective generalise more robustly out of distribution, in line with expectations from work in cognitive science.

The through-line of capacity here allows us to consider how general effects of capacity are on representational structure — looking at the degree to which conclusions from work on humans, can apply to learners in general. Throughout we show that reduced capacity almost always has a regularising effect of some kind, in accordance with expectations from existing work. However, in the experiments looking at model-internal representations, the story becomes more complicated - we analyse regularity with respect to both words and the context they occur in. Larger models are less regular at the word level but can accommodate far greater regularity with respect to context, and it is this contextual regularity that proves predictive of model performance. This nested, multi-level approach to regularity (introduced in chapter 4) potentially offers a way for thinking about certain linguistic phenomena - like iconicity - where languages seem to exhibit regularities with respect to event structure.

[10] I also look at dropout and l2 regularisation as ways of modulating model capacity.

A note on other approaches in this direction

It's important to point out that I'm not the first to notice the potential for language to help us understand other complex systems. In fact much of early cognitive science leverages analogies with language in discussion of other aspects of cognition (e.g. Lashley, 1951; G. A. Miller, 1951). To focus on a few examples of previous approaches, of relevance to the work presented here: Fodor (1975) asserted that human thought is best understood as a language, with our cognition functioning as a system of signs mapping between the world and our thoughts about it. Smolensky (1990) showed how vector spaces, particularly those learned by early connectionist models, can be understood in terms of a generative grammar. Beckner et al. (2009) draws parallels between language and complex systems in physics. More recently analysis of multi-agent deep-learning models has leveraged tools from emergent communication (Brighton and Kirby, 2006 used in Lazaridou et al., 2017). In evaluating different deep-learning models for vision classification Lu et al. (2022) instantiate a method for measuring quantities related to Kirby et al. (2015)'s notions of expressivity and learnability. The long history of work along these lines, makes clear the utility of alluding to language in understanding representational systems.

It's also important to point out that the approach taken here differs from previous work. I make a point of looking at, discussing, and quantifying structure in *mappings*. As discussed in the next chapter this is a fairly general set of functions, at a relatively high level of abstraction. I argue that language is a mapping and can be used as an exemplar against which to make analogies about structures in other mappings like connectionist models – which is distinct from claiming that other mappings are themself a language. This may seem like a needlessly fine hair to split but its an important one. I make no assertions that representations in a mapping need to be symbolic (a la Fodor, 1975), nor do I focus on embedding discrete structures in representations space like Smolensky (1990). I also explicitly quantify structures (e.g. regularity) in a mapping using general-purpose methods, instead of looking at behavioural properties like learnability which is necessarily relative to a learner, I focus on clear, self-contained formalisations that are computationally efficient. Some of my engagement with linguistics at only a high level is also out of respect for the complexity of language and awareness that talking about it in terms of regularity, and variation abstracts much of that away.

Part of the focus on mappings in the abstract is that, to me, some of the beauty found in language's domain generality, is that our understanding of language can underpin our understanding of the world, in general: the information in it, and the structures that define it.

1.8 Thesis Outline

To business. This thesis falls across 7 chapters, and broadly revolves around three core themes.

1. Structural properties found in language are domain-generally useful for understanding mappings that need to be learned, structured, and generalise
2. Quantifying information structure in the mapping learned by a neural network can allow us to describe their learning process, and when and why they generalise
3. Capacity's effect on the emergence of structure in neural networks

---

To summarise the chapters below

1. **How to Represent Information:** A general introduction to the core concepts of this thesis
2. **Information Structure:** Introduces 3 basic structures present in a mapping between two spaces and relates them to information theoretic quantities. The remainder of the chapter provides a brief introduction to discrete information theory.
3. **What We Talk About When We Talk About Compositionality:** This chapter discusses challenges in quantifying structure, and looks at the relationship between compositionality and regularity. I introduce methods for quantifying

variation in a discrete → discrete mapping, showing how previous measures of compositionality implicitly assess regularity. This distinction allows us to make sense of previous results suggesting compositionality isn't related to generalisation. Finally I vary model capacity showing how capacity to have a regularising effect in line with what's predicted by work in linguistics. Work in this chapter is based around Conklin and Smith (2022).

4. **Regularity and Variation in Vector Space**: I use the structural quantities defined in chapter 2 to understand what happens when training a neural network. Transformer models trained on a sequence-to-sequence task go through distinct patterns of expansion, compression, and disentanglement. Based on quantifications introduced in this chapter I can predict how well a model will generalise out of distribution, laying out the kinds of structures that seem critical for generalisation. Work in this chapter is based around Conklin and Smith (2024).

5. **Information, Generalisation and Scale, in Large Language Models**: Here I apply the information structure analysis from earlier chapters to large language models. Showing how they follow a similar training trajectory to their smaller counterparts. As with models trained on a single task we find correlations between particular representational structures and downstream performance, further showing what representational structures drive generalisation.

6. **Biasing Representational Structure with Meta-Learning**: In a final chapter, I look at how to bias representational structure using a meta-learning objective. Optimizing a model's update steps to be beneficial to similar examples, and showing that this improves out-of-distribution generalisation ability. This chapter also starts with some discussion of how the information structure framing used throughout the thesis relates to more behavioural properties like memorisation and generalisation. Experiments presented here are based around Conklin et al. (2021).

7. **Conclusion:** Here we revisit the core themes of the thesis, highlighting common threads between the preceding chapters and laying out directions for future work.

Reproducibility Unless stated otherwise, code and data are available at https://github.com/hcoxec/h.

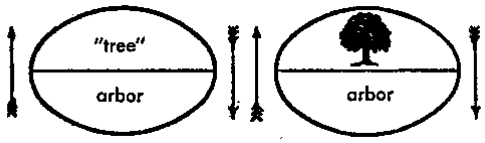
Chapter 2

Information Structure

a primer on information theory for quantifying structure

[redacted] feeling herself change painfully cell by cell into a shadow, a laurel, you, a constellation.

- James Richardson



Mappings relate two spaces - alphabet/morse code, input/encoding, meaning/form - and their structures can vary widely depending on where and how they're used. We want a general-purpose way to describe their structure quantitatively, so we consider three kinds of primitive structure present in a mapping: one-to-one, one-to-many, and many-to-one. By assessing each of these quantities continuously, we can describe a mapping in terms of how much of each structure is present. Each of our primitive structures relates intuitively to basic information theoretic quantities, the majority of this chapter is a primer on information theory (in the discrete case) and the quantities relevant to the chapters that follow. Before that, I give a quick overview of the primitive mapping structures we look at later. The

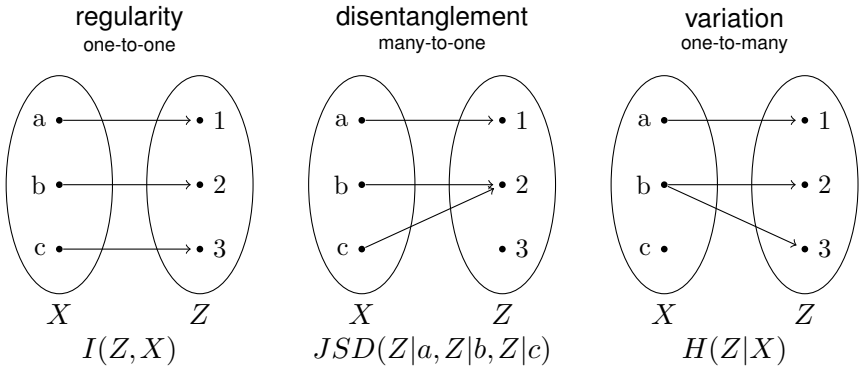


Figure 2.1: Three basic kinds of mapping structure we consider here, labelled with their linguistic analog, and the information theoretic quantity we introduce to measure them in chapter 4. Note that we show part of the mapping ($a \rightarrow 1$) as regular in all cases because the mappings we consider exhibit a combination of all 3 structures. As such we assess the *degree* of each structure, not whether or not it exists. Variation (one-to-many) is possible here because our X contains instances of the same label (like a word) in different contexts (like sentences), meaning $b \rightarrow 2$ and $b \rightarrow 3$ reflects b in different contexts which are not shown here for brevity.

next chapter presents initial experiments quantifying specific linguistic structures in the discrete-to-discrete mapping learned by a multi-agent model. Chapters 4 & 5 build on this, introducing the more general framework for thinking about structure described briefly in the next section and applying it to discrete-to-continuous mappings learned by Neural Networks.

2.1 Structural Primitives

Consider 3 basic kinds of structure that can exist in a relational mapping between two spaces: one-to-one, one-to-many, and many-to-one. These are depicted visually for a mapping between spaces X and Z in figure 2.1, along with the information theoretic quantities we relate them to later on. At a high-level these are:

One-to-one (Regularity)

Maximised when each unique $x_i \in X$ maps to a unique $z \in Z$ and each $x_i^k \in x_i$ maps to the same z regardless of context. This reflects how predictable a mapping between spaces is, or the degree to which the spaces X and Z are monotonically aligned. If we were to think about this in terms of a function f that maps $f(x) \rightarrow z$ regularity is related to how *injective* f is. A regular mapping is structure preserving - we can recover the input x from the corresponding output z

Many-to-one (Entanglement[1])

Maximised when all $x \in X$, map to the same z_i, regardless of which x it is or the context in which it occurs. Reflects the degree of ambiguity in the mapping - how hard it is to infer which input x has been mapped to a given z_i. Given $f(x) \rightarrow z$ this is related to how *non-injective* f is. Linguistically this is most clearly related at a lexical level to *homonymy* where multiple meanings have the same surface form - an analogy discussed at length in chapter 3. A entangled mapping is not structure preserving - a given output z could have many corresponding input xs.

One-to-many (Variation)

Maximised when each input has a different representation for each context where it occurs — i.e. each x_i^k maps to a unique z_i. This is the inverse of regularity, reflecting the degree of contextual variation in the mapping. A function which maps the same input x to different outputs z violates the general definition of a function. But we can think of this as a kind of reciprocal of entanglement, and say given $g(z) \rightarrow x$ this quantity reflects how non-injective the mapping from z's to x's is. This highlights that entanglement and variation are virtually identical except for their directionality (one-to-many vs. many-to-one). Lexically this is related to *synonymy* in natural language, where the same meaning has multiple different realisations in form often dependent on context. Structurally we can relate this to word-order freedom, or the degree of variability in the mapping between semantic roles and linear order in form.

[1]Note that we often quantify disentanglement, rather than entanglement because it better aligns with the information theoretic divergences used to measure it. For our purposes these are the same quantity, inverted.

These are the structures of interest in brief and, while basic, later chapters show that evaluating them at different levels of abstraction can give substantial insight into the structure present in a wide array of mappings. In order to quantify these we need to be able to quantify the information present in each space, and what information is preserved or compressed as we map between them. For a quantitative approach to information, we turn to information theory.

2.2 Discrete Entropy

Information theory describes relationships between spaces, and is built upon a 'mathematical theory of communication' (Shannon, 1948). Shannon considers an information source producing messages that are encoded in a signal by a transmitter, to be later decoded back to the original message by a receiver; built on a broad analogy to language where a speaker encodes their thought in a sentence decoded by a listener. It's worth noting this is similar to the analogy I make throughout this thesis, relating different kinds of mappings to language as a point of reference. Originally concerned with how to optimally map messages to signals, information theory has found explanatory power across a wide array of disciplines from genetics (Lezon et al., 2006), to neuroscience (Paninski, 2003) and machine learning (MacKay, 2004). In later work Shannon looks at ambiguity in encoding schemes, quantifying the allowable degree of ambiguity in a mapping from messages to signals that still allows the structure of the original message to be recovered (Shannon, 1949) - this work eventually forms the basis of lossy compression, and means this area of mathematics has extensive tools for thinking about information quantitatively. At it's core information theory considers data probabilistically, and so like probability itself has different instantiations for discrete and continuous cases. Here we introduce the discrete case, which is easier to reason about intuitively, and is used in the next chapter. Later in Chapter 4 we formalise the same quantities for continuous cases.

2.2.1 Quantifying Information

How can we tell how much information is in a sample of data? Not how many gigabytes it is, or how many entries it has, but how much *information* it contains. Let's say for a moment that we had two datasets, one of which contains all

26,145 words of the full text of King Lear, the other containing 26,145 words comprised of just "king" and "lear" repeated over and over again. Clearly the former contains far more information despite the fact that both are identical in size - what we want is a way to reliably quantify this difference. We can do this using information entropy (Shannon, 1948) which quantifies information by looking at data probabilistically. It follows from the intuition that the more frequent something is, the less informative it is. Consider the 5 most frequent words in English *the, be, to, of, and*, most of which have a primarily grammatical function, conveying little semantic content of their own; this becomes even more clear were we to take a passage

(a) We went outside. He adjusted the shutter. He told me where to stand, and we got down to it. We moved around the house. Systematic. Sometimes I'd look sideways. Sometimes I'd look straight ahead. "Good," he'd say. "That's good," he'd say, until we'd circled the house and were back in the front again. "That's twenty. That's enough." "No," I said. "On the roof," I said. - (Carver, 1981)

and remove any of the 100 most frequent words in English according to the Oxford English Corpus (Stevenson, 2010).

(b) [redacted] outside. [redacted] adjusted [redacted] shutter. [redacted] told [redacted] where [redacted] stand, [redacted] down [redacted] moved around [redacted] house. Systematic. Sometimes [redacted] sideways. Sometimes [redacted] straight ahead. [redacted] [redacted] until [redacted] circled [redacted] house [redacted] front again. [redacted] twenty. [redacted] enough [redacted] roof [redacted]

Which immediately makes the text ungrammatical, but we can still recover quite a lot of what the passage is about - taking pictures outside ('outside adjusted shutter') while circling a house ('moved around house.. until circled house'). By contrast, removing any words not in 100 most frequent

(c) We went [redacted] He [redacted] the [redacted] He [redacted] me [redacted] to [redacted] and we got [redacted] to it. We [redacted] the [redacted] I'd look [redacted] I'd look [redacted] "Good," he'd say. "That's good," he'd say, [redacted] we'd [redacted] the [redacted] and were back in the [redacted] "That's [redacted] That's [redacted] ." "No," I said. "On the [redacted] ," I said.

we end up with text, where the meaning of the original is essentially unrecoverable. We can still piece together that two or more people ('we went' 'he' 'me') are doing something that goes well ("Good," he'd say. "That's good," he'd say,") but nothing more. While there is information in both edits of the sentence, there is considerably more in the version that retains lower-probability words.

2.2.2 Self Information/Surprisal

Armed with this intuition, that the amount of information in a piece of data is related to how likely it is, Shannon quantifies the amount of information in an event as the *self information*, also termed surprisal. Because we're talking about information in terms of probability, given some data we need a probability distribution that describes it. For text data there are a number of ways of describing it probabilistically - for simplicity we create random variable $\mathcal{X}$ that describes our data at the word level, where each event in the distribution x_i is a word that occurs in the text, and its probability refers to the frequency of that word. Given this, the self-information of each word is the negative log of its probability.

$$s(x_i) = -\log p(x_i) \tag{2.1}$$

When the probability of an event $p(x_i)$ is 1.0 its log is 0, and as $p(x_i)$ approaches 0 $s(x_i)$ monotonically increases (shown figure 2.2 left). This definition and its use of a logarithm are intended to satisfy:

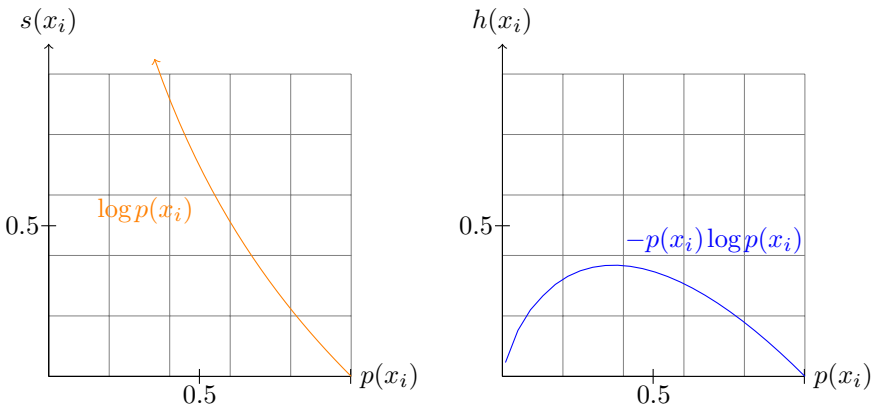


Figure 2.2: Plots showing suprisal (left) and entropy (right) values for a single event. The x axis is the event probability, while the y axis shows the infromation theoretic quantity. Note that suprisal continues off the plot approaching infinity as $p(x_i)$ approaches 1.

---

i. A constant event, with probability 1.0, conveys no information; it's completely unsurprising

ii. An unattested event, with probability 0.0, is infinitely suprising

iii. The less likely an event, the more information it contains

---

Note that the 'unit' of entropy depends on the base of the logarithm used. In what follows I use the natural logarithm unless otherwise noted, which means all entropies are in *nats* (for bits use base 2, for dits use base 10).

2.2.3 Entropy: Expected Information

To get the amount of information contained in the whole distribution $\mathcal{X}$, rather than just one event, we aggregate the self information of the events it contains — for instance, we might aggregate across the information contained in each word in a vocabulary for text data. Information Entropy aggregates using the expected value operator, a weighted mean where the weight is determined by the probability of the event. Figure 2.2 (right) shows this quantity for a single event.

$$h(\mathcal{X}) = \sum_{x_i \in \mathcal{X}} -p(x_i) \log p(x_i) \tag{2.2}$$

Compared with self-information, entropy assigns proportionally less information to less likely events. In practice this can be useful in cases with many low probability events - whose self-information will approach infinity which can make estimates numerically unstable.

At the distribution level, entropy describes how peaked a distribution over events is. As a single event in the distribution becomes increasingly probable the overall entropy decreases. This is shown below for a random variable with four events. The uniform distribution achieves highest entropy, which decreases as the distribution becomes more peaked.

We can also see it as reflecting the number of samples needed from a distribution in order to tell its shape and the probability of the events it contains. When one

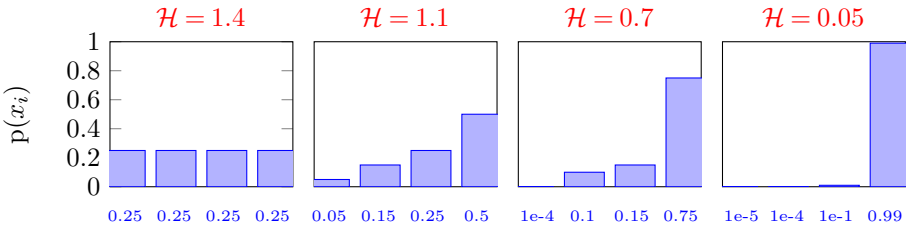


Figure 2.3: A distribution with 4 events is shown above in 4 different versions. On the x axis is the probability of each event, also labelled below each bar. Above each plot is the entropy of the distribution. As events become less uniformly distributed - more peaky - entropy decreases.

event occurs 100% of the time, and entropy approaches 0, we hardly need any samples at all. But as the distribution becomes more uniform we need more and more samples in order to have all possible events be attested, and to get a good estimate of their probability.

This leads us to a final intuitive way of thinking about entropy most relevant to later chapters: as the amount of variation in data. If $\mathcal{X}$ is a distribution over words, then $\mathcal{H}(\mathcal{X})$ is minimised when only one word is used - meaning there's no variation in word-choice. As the words used in the text vary more and more, the distribution over them becomes more uniform, and entropy increases. To summarise the perspectives, entropy reflects:

---

1. the amount of information in a random variable
2. the expected level of surprise from any sample from a distribution
3. how peaked a distribution over events is
4. the relative number of samples needed from a distribution in order to estimate it
5. **the amount of variation in the data a distribution describes**

With this in mind we can return to the task we started with - telling apart our two documents, one with the text of king lear, the other with the words 'king' and 'lear' repeated for the same number of words. We build a vocabulary for each

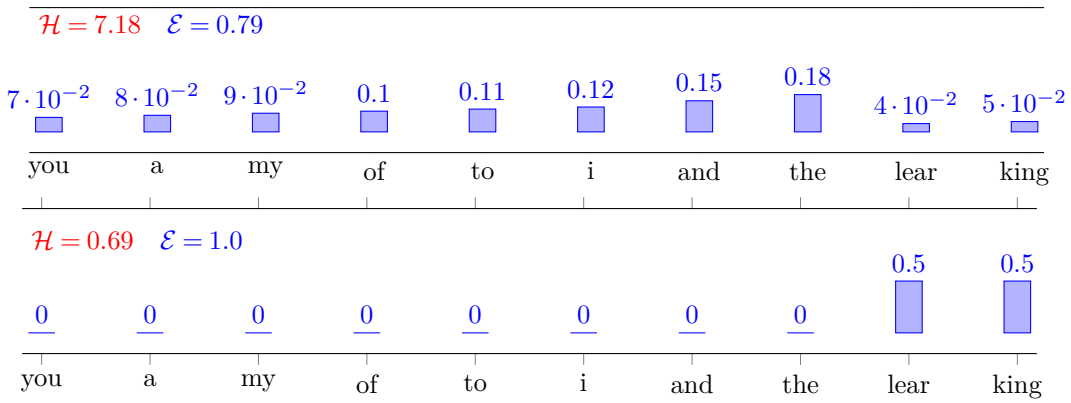


Figure 2.4: The probabilities of the 10 most frequent words from both documents plotted, along with their respective entropies and efficiencies. As shown the repeated document has considerably lower entropy than the full text, but higher efficiency as it is more uniformly distributed over events it contains. Note that the lower distribution is only comprised of the two words with non-zero probability, non-existent events are shown for continuity with zero probability but do not factor into the efficiency calculation.

document containing all words that occur in it, then create a random variable for each of them with the events in the distribution reflecting the probability of a given word (summarised in figure 2.4). With these distributions we can say the entropy of the full text of kind lear is 7.18 nats, while the document of the same length with two repeated words is only 0.69 - the full text has more information.

2.2.4 Efficiency: Normalised Entropy

> Degree of non-uniformity, independent of distribution size. Bounded $0 < \mathcal{E}(\mathcal{X}) < 1$: the more uniform it is, the closer its efficiency is to 1. The more peaked it is, the closer to 0.

An issue in interpreting entropy values is that the quantity itself is, in principle, unbounded. You could have longer, and longer documents with greater complexity so entropy is bounded $0 < \mathcal{H}(\mathcal{X}) < \inf$. This can make it difficult to compare entropy values for different distributions - a uniform distribution with 10 events will have lower entropy than a uniform distribution with 20 events. While this makes sense given the intuitions we've discussed so far - 20 equiprobable events encode more information than 10 - often we want a relative quantity that can be compared across distributions of different sizes. To get this we focus on the degree of non-uniformity in a distribution, or peak-iness, by normalising the entropy of a

random variable by the entropy of a same sized uniform distribution - remember that a uniform distribution represents the highest possible entropy. Helpfully, the entropy of a uniform distribution is equivalent to the logarithm of the number of events it contains.

$$\mathcal{E}(\mathcal{X}) = \frac{\mathcal{H}(\mathcal{X})}{\log(|\mathcal{X}|)} \tag{2.3}$$

The resulting quantity is called efficiency and is bounded between 0 and 1, such that an efficiency of 0 indicates a one-hot distribution, and 1.0 indicates a uniform. Shannon terms this efficiency because it reflects what proportion of a distribution's maximum possible entropy is actually used.

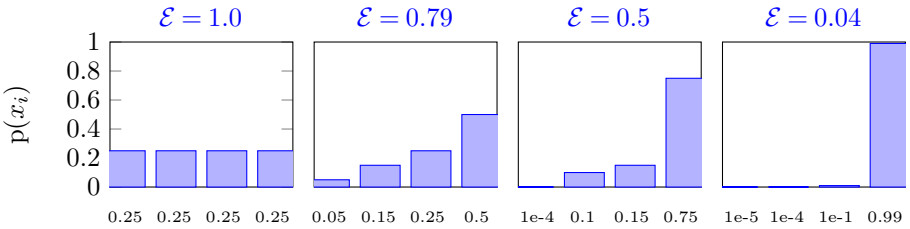


Figure 2.5: The same distributions shown in figure 2.3 but now labelled with the efficiency of each distribution.

Referring back to figure 2.4, note that the document with just the words 'king lear' repeated has lower entropy than the full text of the play, but higher efficiency. The two words present in the document are perfectly uniformly distributed so the repeated document scores an efficiency of 1.0, higher than the full text's efficiency of 0.79. The repeated text contains as much information as is possible for a document with only 2 words. Even if a random variable contains less information than another, it may be more efficient by virtue of being more uniformly distributed over the events it has.

2.2.5 Conditional Entropy

> The amount of variation with respect to a single feature in data. Analogous to the entropy of a distribution when a certain feature is true – it tells us how much information about that feature is contained in a distribution

Conditional entropy builds on top of conditional probability. If we know that our data exhibits certain features we can calculate the probability distribution over events, when a feature is true. For a distribution X that describes word

frequencies in text data which is a mixture of fiction and non-fiction documents, the conditional distribution $P(X|\text{fiction})$ gives word probabilities using only the fictional portion of the data. Accordingly the entropy $\mathcal{H}(X|\text{fiction})$ tells us how much information is in the fiction data alone, separate from non-fiction. The entropy of a conditional distribution is a conditional entropy:

$$\mathcal{H}(X|\text{label}) = \sum_{x_i \in \mathcal{X}} -p(x_i|\text{label}) \log p(x_i|\text{label}) \tag{2.4}$$

Where *label* refers to a known feature label for the data. This quantity is useful because it allows you to understand how parts fit together into a whole – it's a key building block of the approach taken in later chapters. To get a better intuition of how it works, let's say that we have a library containing selected texts from the following authors:

- William Shakespeare (1564-1616): *King Lear, Hamlet, Titus Andronicus, Twelfth Night, As You Like It*
- Christopher Marlow (1564-1593): *Doctor Faustus, Tamburlaine*
- Raymond Carver (1938-1988): *What We Talk About When We Talk About Love*

We want a single distribution that describes the entire library. Again, there are many ways to do this, for simplicity we opt to create a distribution reflecting word-level information, where each event is a word and its probability reflects word frequency. We'll call this distribution for the entire library $P(words)$.

The entropy $\mathcal{H}(words)$ is 7.2 nats, its efficiency is 0.76 (shown in table 2.1). This reflects the degree of variation in word use across the entire library. If all the words in the library were more uniformly used efficiency would be closer to 1, if only a few of the words were reused over and over the efficiency would be closer to 0. Computing the conditional entropy of words given each each of the individual books $\mathcal{H}(words|title)$ reflects how much information is encoded in each title, or how much word use varies in a given title. As shown in table 2.1 the entropy of each title is less than the overall $\mathcal{H}(words)$, but not by much - with the lowest entropy text $\mathcal{H}(words|\text{what we talk about...}) = 6.457$. This tells us that most word frequency information is shared across texts, which makes sense given these are all in the same language and we wouldn't necessarily expect words like

the, be, a, and to appear dramatically less often in any of them. But the fact that the overall entropy is higher than any individual text tells us that there is information that they don't share which is added by combining the texts together. Importantly we also know more than one thing about the source documents - for example, we know their authors. We can just as easily compute the conditional entropy $\mathcal{H}(words|author)$. This groups together the 5 Shakespeare plays into a single distribution and tells us how much word choice varies in the collection of plays we have by them for each author, or how much information is that collection.

In the general case we can define sets of labels that describe different values for a feature in the data a distribution describes. These are used in conditional entropies, and we take the entropy of a set as the average entropy across the constituent labels.

$$h(\mathcal{X}|\text{set}) = \frac{1}{|\text{set}|} \sum_{\text{label} \in \text{set}} \mathcal{H}(\mathcal{X}|\text{label}) \tag{2.5}$$

We can use this to look at variation in data at different levels of abstraction based on what we know about the texts that comprise the library. Some examples of sets of labels we could consider:

- Title: *King Lear, Hamlet, Titus Andronicus, Twelfth Night, As You Like It, Doctor Faustus, What We Talk About When We Talk About Love*
- Author: *Shakespeare, Marlow, Carver*
- Genre: *Tragedy, Comedy, History*
- Century: *16th, 20th*

2.3 Mutual Information

How much variation we can explain in terms of a property of the data. Reflecting the reduction in entropy when a given label is true, or how much knowing a label tells us about data.

Library	$\mathcal{H}(words)$	$\mathcal{E}(words)$	
library	7.1591	0.7685	
Title	$\mathcal{H}(words\|title)$	$\mathcal{E}(words\|title)$	$I(words, title)$
King Lear	6.4703	0.6946	0.6888
Hamlet	6.4269	0.6899	0.7322
Titus Andronicus	6.4310	0.6904	0.7280
Twelfth Night	6.3090	0.6773	0.8501
As You Like It	6.2584	0.6718	0.9007
Doctor Faustus	6.3583	0.6826	0.8008
Tamburlaine	6.3995	0.6870	0.7596
What We Talk About ...	6.0382	0.6482	1.1209
Author	$\mathcal{H}(words\|author)$	$\mathcal{E}(words\|author)$	$I(words, author)$
William Shakespeare	6.4603	0.6935	0.6987
Christopher Marlow	6.4683	0.6944	0.6908
Raymond Carver	6.0382	0.6482	1.1209

Table 2.1: Entropies, efficiencies, and mutual informations for our library of texts. Conditional entropies are shown for two different kinds of conditioning labels, title and author.

Mutual information is related to both entropy and conditional entropy. It tells us how much we reduce the overall entropy by knowing a conditioning label. It's computed by taking the difference between the overall entropy and the conditional.

$$I(\mathcal{X}, \text{label}) = \mathcal{H}(\mathcal{X}) - \mathcal{H}(\mathcal{X}|\text{label}) \tag{2.6}$$

This tells us the relationship between a distribution and its subset. Given we can look at conditional entropy as reflecting the degree to which a property varies, mutual information quantifies the inverse - how regular the data is with respect to a property, or how aligned a distribution is with respect to a label. In the library example the distribution contains word level information, so a mutual information $I(words, shakespeare)$ reflects how predictable Shakespeare's word choice is. $I(words, shakespeare)$ would be maximised if Shakespeare used only one word in all his plays, meaning just knowing the play was by Shakespeare would tell us everything there is to know about which words it contains. When

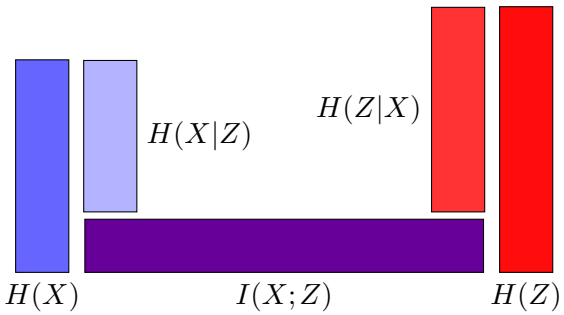


Figure 2.6: Relationships between basic information-theoretic quantities. Given two spaces X and Z, their entropies $H(X)$ and $H(Z)$ are the information each of them contains. Mutual information $I(X,Y)$ reflects the information they share - how monotonically aligned they are. The conditional $H(X|Z)$ reflects the information unique to X not contained in Z and $H(Z|X)$ is the information unique to Z.

maximised the author label *Shakespeare* would be monotonically aligned with a single word, with degree of alignment decreasing as the author's word choice varies more.

In practice mutual information for all 3 authors in our library is relatively low, indicating somewhat unsurprisingly that they each use well more than one word in their writing. Raymond Carver has higher mutual information $I(words, carver)$, than the other authors indicating that his word choice is more predictable (less variable) than the library in general. This could reflect something stylistically about modernist writing of the later 20th century being less verbose than iambic verse from 1608. Alternately it could be driven by the fact that the library is predominantly texts from 400 years before Carver. As a result $P(words)$ likely reflects Shakespeare and Marlow's use of words like *anon, assay, dost, doth, hark, thee*, and the lower $I(words, carver)$ may reflect Carver being more aligned with a subset of words still in use in the late 20th century.

2.4 Relationships Between Entropy, Conditional Entropy, and Mutual Information

These three quantities fit together to describe how two distributions relate to each other. Shown in figure 2.6, for two distributions $P(X)$ and $P(Z)$, or in the current example a distribution over words based on the entire library $P(words)$,

and a distribution over authors $P(author)$ where each event is an author's name and probability reflects the number of words in the library written by that author. $H(words)$ and $H(author)$ describe the amount of information in each of them. $I(words, author)$ describes the amount of information they share - how predictive knowing the author of a text is of the words it contains. The conditional $H(words|author)$ - discussed above - reflects the variation in each author's word choice, or - as a reciprocal to mutual information - the amount of information about an author's word choice we can't determine just by knowing the author. We can also compute a condition in the other direction $H(author|words)$ which tells us for a given word the variation in which author uses it - maximised when that word is equiprobably used by all authors.

We can look at this as an example of a simple mapping, between authors names and the words they write. Using these basic concepts we can tell how much information is preserved moving between spaces (mutual information), and how much information is unique to each space (conditional entropy). In some cases though we want to be able to tell how much information is unique to each in a set of labels - like how different the word choices are for different authors - without computing the conditional $H(author|words)$. For this we can use a divergence.

2.5 Jensen-Shannon/Lambda Divergence

> How separable a set of distributions are from each other. How much the information in different distributions overlap.

Often we have a number of different distributions and we want to know how similar they are to each other; if their information overlaps or is fully separable. There are a number of ways of assessing this; here because we need to tell apart a number of distributions we opt for the Multivariate Jensen Shannon Divergence, sometimes called the Lambda Divergence. This computes a mixture of the distributions M by taking a weighted mean of the distributions we're comparing. In the general case laid out above for a set of labels this is the sum of the conditionals $P(Z|\text{label})$ each weighted by the probability of the label $P(\text{label})$.

$$M \propto \sum_{\text{label}}^{\text{set}} P(\text{label})P(Z|\text{label}) \tag{2.7a}$$

Given this mixture we try to explain the information it contains $H(M)$ in terms of the information in each component of the mixture $H(Z|\text{label})$ weighted again by

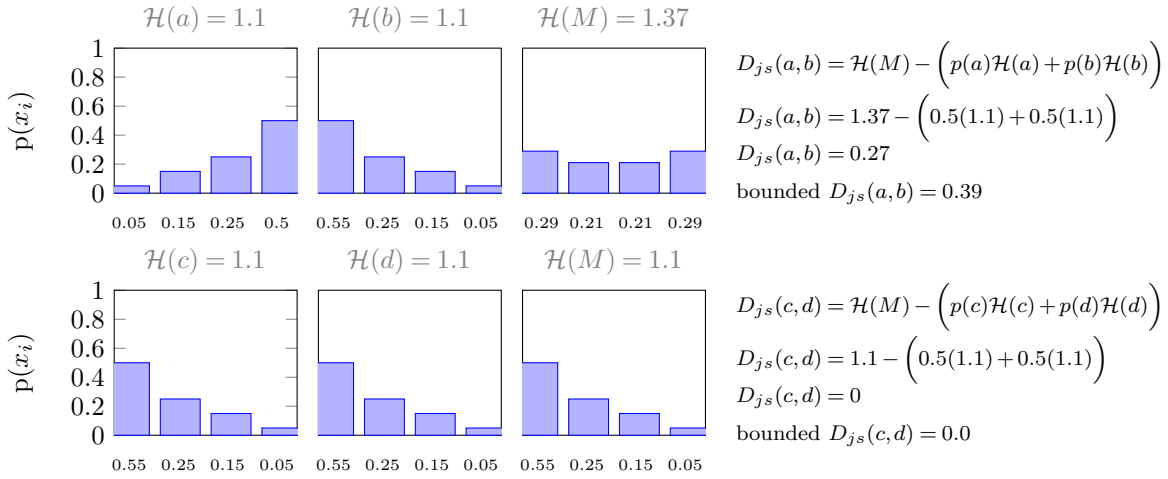


Figure 2.7: Jensen Shannon divergences computed for two sets of distributions {a,b} and {c,d}. In both cases we compute a mixture distribution M by taking an unbiased mean — as noted in body text, this mixture can be weighted by the probability of each of the component distributions, we opt to weight them equally here for simplicity. The JS divergence then looks to see if we can explain the information in the mixture distribution in terms on the component distributions. a and b, overlap somewhat but are distinct - as a result the JS divergence is 0.27, or 0.39 when bounded to lie between 0 and 1. By contrast c and d are identical, and so their JS divergence is 0.0.

how much that component contributed $P(\text{label})$ — shown below. The resulting quantity is bounded by the entropy of the mixture $H(M)$ and so can be bounded to lie between 0 and 1. As values approach 1 component distributions overlap less, and as the divergence approaches 0 the component distributions become identical. This is implicitly the mutual information between the mixture distribution M and the weights used to combine distributions $P(\text{label})$. If the components of the mixture don't overlap then the mixture weights will explain all the information in the mixture - if components do overlap then M will be less aligned with the weights used to compute it. Two example cases are visualised in figure 2.7.

$$D_{JS}(Z||\text{set}) = H(M) - \sum_{\text{label}}^{\text{set}} P(\text{label})H(Z|\text{label}) \tag{2.7b}$$

It's worth noting that this computes the divergence between the component distributions and their mixture, rather than comparing the distributions individually. We can use this to tell how separable sets of labels are from each other, taking

the example above we can look at how separable the word frequency distributions are for each author. $D_{JS}(Z||[\text{shakespeare, marlow, carver}]) = 0.534$, indicating the distributions for the 3 authors are relatively separable. Note that in this case because the conditionals include all the data in the library the mixture ends up equalling the library distribution $P(M) = P(words)$. If we instead estimate $D_{JS}(Z||[\text{shakespeare, marlow}]) = 0.5263$ or $D_{JS}(Z||[\text{shakespeare, carver}]) = 0.5845$ we can see word choice is more similar for Shakespeare and Marlow, than for Shakespeare and Carver.

These are the 4 information theoretic quantities we need for the remainder of the thesis, so to summarise:

---

i **Entropy:** The amount of variation in data

ii **Conditional Entropy:** The amount of variation in a subset of data

iii **Mutual Information:** The reciprocal of Conditional Entropy, reflecting how much less variable a subset is than overall. By extension how predictable a subset of data is.

iv **Jensen Shannon Divergence:** How separable subsets of data are from one another. The degree to which their information does not overlap (1 indicating no overlap).

I have introduced these quantities here in the discrete case; chapters 4, and 5, consider information theory for continuous spaces. First though, the next chapter draws explicit parallels between different conditional entropies in a discrete-to-discrete mapping and different kinds of linguistic variation.

Chapter 3

What We Talk About When We Talk About Compositionality

Variation in Discrete → Discrete

[redacted] you say there are no haloes around the streetlights [redacted]
what I see is an aberration [redacted] I tell you it has
taken [redacted] all my life [redacted] to
soften and blur and finally banish the edges you regret I don't see [redacted]

- Lisel Mueller

Neural Networks are known for finding solutions to complex problems, but not always the solutions we'd expect. A model trained to predict whether or not a patient had pneumonia based on their chest x-ray appeared to do so with remarkable accuracy, until a meta-analysis (Zech et al., 2018) noticed that each x-ray has information in it indicating which scanner and which hospital it came from (most notably from a metal tag radiographers place on the patient's shoulder). In the training data different hospitals had different prevalences of pneumonia, meaning you can predict whether or not a patient had the disease with relatively high fidelity based on where the x-ray was performed. Rather than learning to identify if a patient's lungs had damage consistent with pneumonia, the model

learned a much simpler solution: identify what hospital performed the scan. Deep-Learning models are most often optimised with back-propagation of error via gradient descent (Rumelhart et al., 1986). This tries to minimise the model's error with respect to an objective - like classifying scans as healthy or diseased - but provides no supervision for how the model solves that problem. As a result it's often difficult to work out if a model's behaviour is reflective of it having learned a mapping that identifies and preserves the necessary information from its input, or it having found some simpler solution that can mimic that behaviour. Models can easily rely on heuristics - like the co-occurence probability of different input features - to perform well on a task without learning the properties of their training data we'd expect them to (McCoy, 2019). Understanding what representational information drives models' behaviour remains a major challenge - across domains - when trying to draw conclusions from experiments with deep-learning.

As training large-scale neural networks became more tractable in the past decade, a series of papers started using them to replicate earlier work on the origins of human language (Choi et al., 2018; Kottur et al., 2017; Lazaridou et al., 2017, 2018; Mordatch & Abbeel, 2018). Given that language leaves behind no fossil record, linguists often turn to computational simulations to study how linguistic systems can emerge in a population. Previously, simple probabilistic models gave an account of how structural properties of language like compositionality can emerge in response to the dynamics of transmission and use rather than natural selection on the language faculty (e.g. Brighton et al., 2005; Kirby, 2001) or by processes of biological evolution (Nowak et al., 2000) or gene-culture co-evolution (Smith et al., 2003). Lazaridou et al. (2018) implemented a multi-agent model with two neural networks playing a signalling game where a sender network maps a meaning to a discrete signal, a receiver network then tries to map this signal back to the original meeting, in a high-level analogue to communication. Both are then optimised for 'communicative success' - to have the receiver's reconstruction match the original meaning as often as possible. Using this setup senders and receivers could reliably converge to near-perfect communication on both the examples they saw during training, and on thousands of unseen examples.

Despite this, the mappings that emerged showed limited evidence of compositional structure – an essential component of generalisation in natural language. Qualitative analyses (Choi et al., 2018; Havrylov & Titov, 2017), try to identify certain subunits of of signals that corresponded to specific features in the input space

- but it's difficult to manually look for structure in thousands of strings of characters. Quantitative assessments of compositionality showed the mappings scored well below any 'idealised' compositional systems used for reference (Brighton and Kirby (2006) used in Lazaridou et al. (2018), and Resnick et al. (2020)), and that 'degree of compositionality' had no correlation with generalisation performance (Chaabouni et al., 2020). This raises a real question as to whether language-like structure had emerged in the sender's mapping from meaning's to signals, or the model had found a heuristic solution to the communicative task.

Implicitly, when you look for structure in a system, you make some assertions about what structure is and what it should look like. Each method instantiates its own definition of structure which makes proving a null-result difficult: does a system lack compositional structure or does your quantification look for something else? The remainder of this chapter looks at existing methods for identifying compositional structure in models of language emergence, and shows that they actually assess the degree of regularity in a system, not whether or not the system is compositional. As a result they discount mappings with variation as being non-compositional, and by extension indicative of a in-human approach to communication, despite the fact that variation is typologically ubiquitous.

This chapter represents initial experiments in quantifying structure in a mapping. It approaches the problem with less generality than later chapters, focusing on models designed to be directly relatable to human language. As a result it introduces information theoretic measures of four specific kinds of linguistic structure, two structural and two lexical. These are essentially specific instantiations of the one-to-many (variation) and many-to-one (disentanglement) kinds of structure mentioned at the start of the previous chapter (section 2.1). By starting with a discrete-to-discrete mapping in a model of language emergence we can draw clear parallels between the structures we quantify and their analogs in linguistics, which prove useful when we apply these measures to discrete-to-continuous mappings in the next chapter.

The remainder of this chapter is based around a paper **Compositionality with Variation Reliably Emerges in Neural Networks** that appeared at the International Conference on Learning Representations in 2023. Authors are myself and Kenny Smith - I conceived of and ran experiments myself, and wrote the paper - Kenny gave writing feedback prior to submission to the conference. The paper is presented here minimally

changed from the conference version that underwent peer-review. Changes are largely related to formatting to make the content more readable outside of the original conference paper template.

3.1 Compositionality with Variation Reliably Emerges in Neural Networks

Compositionality is a defining feature of natural language; the meaning of a phrase is composed from the meaning of its parts and the way they're combined (Cann, 1993). This underpins the powerful generalization abilities of the average speaker allowing us to readily interpret novel sentences and express novel concepts.

Robust generalization like this is a core goal of machine-learning: central to how we evaluate our models is seeing how well they generalize to examples that were withheld during training (Bishop, 2006). Deep neural networks show remarkable aptitude for generalization in-distribution (Dong & Lapata, 2016; Vaswani et al., 2017), but a growing body of work questions whether or not these networks are generalizing compositionally (Kim & Linzen, 2020; Lake & Baroni, 2018), highlighting contexts where models consistently fail to generalize (e.g. in cases of distributional shift; Keysers et al., 2020).

Recent work has looked at whether compositional representations emerge between neural networks placed in conditions analogous to those that gave rise to human language (e.g. Choi et al., 2018; Kottur et al., 2017). In these simulations, multiple separate networks need to learn to communicate with one another about concepts, environmental information, instructions, or goals via discrete signals - like sequences of letters - but are given no prior information about how to do so. A common setup is a 'reconstruction game' modelled after a Lewisian signalling game (Lewis, 1970), where a sender network describes a meaning using a signal, and a receiver network needs to reconstruct that meaning given the signal alone. The resulting set of mappings from meanings to signals can be thought of as a language.

Previous work has shown that in this setup models reliably develop a language that succeeds not only in describing the examples seen during training but also

successfully generalizes to a held-out test set, allowing accurate communication about novel meanings. Despite this capacity to generalize, which is a product of compositionality in natural languages, existing analyses of those emergent languages provide little evidence of reliable compositional structure (see Lazaridou & Baroni, 2020, for a review), leading some to suggest that compositionality is not required in order to generalise robustly (Andreas, 2019; Chaabouni et al., 2020; Kharitonov & Baroni, 2020).

If not compositional, then what? This interpretation leaves us with a major puzzle: if the languages that emerge in these models are non-compositional, how do they allow successful communication about thousands of unseen examples (e.g. Havrylov & Titov, 2017; Lazaridou et al., 2018)? If the meaning of a form is arbitrary rather than being in some way composed from its parts there should be no reliable way to use such a mapping to generalize to novel examples (Brighton, 2002). Here we provide an answer to this question showing that emergent languages are characterised by *variation*, which masks their compositionality from many of the measures used in the existing literature. Existing measures take regularity as the defining feature of a compositional system, assuming that in order to be compositional separate semantic roles need to be represented separately in the signal (Chaabouni et al., 2020), or that symbols in the signal must have the same meaning regardless of the context they occur in (Kottur et al., 2017; Resnick et al., 2020). Alternately they expect that each part of meaning will be encoded in only one way, or that the resulting languages will have a strict canonical word order (Brighton and Kirby (2006) used in Lazaridou et al. (2018)). However, natural languages exhibit rich patterns of variation (Goldberg, 2006; Weinreich et al., 1968), frequently violating these four properties: forms often encode multiple elements of meaning (e.g. fusional inflection of person and number or gender and case), language is rife with homonymy (where the meaning of a form depends on context) and synonymy (where there are many ways of encoding a meaning in form), and many natural languages exhibit relatively free word order.

This offers us a different explanation of previous results: compositional systems may emerge, just with variation. If so that doesn't necessarily undermine their compositionality, natural languages show us that systems can have considerable variation while retaining the generalizability that makes compositionality so desirable. We focus on explicitly assessing variation independent of composi-

tionality and illustrate how emergent languages can generalize robustly even with substantial variation. Our core contributions are as follows:

- We introduce 4 measures of natural language-like variation
- We show that the languages which emerge tend to exhibit a high degree of variation which explains why previous metrics would classify them as non-compositional.
- We find that a language's degree of regularity correlates strongly with generalization early in training, but as the emergent language becomes *regular enough* to generalize reliably this correlation goes away.
- We reduce the capacity of our models by reducing the size of the hidden layers, and show that lower capacity models develop more regular languages, as predicted by accounts linking cognitive capacity and regularity in natural language

3.2 Variation, Regularity, & Compositionality

Variation and compositionality in language are related but distinct. We look at them separately, taking a language's generalization performance as an indication of whether or not it is compositional (in line with Brighton, 2002; Kottur et al., 2017). Linguistic regularity - the absence of variation - has been studied in broad array of contexts (see Ferdinand et al., 2019, for discussion). At a high-level it describes how predictable a mapping from meaning to form is; if there's only one way of encoding a meaning that mapping is highly-regular (Smith & Wonnacott, 2010). Conversely if there's a variety of different ways of encoding a meaning that mapping likely has high variation (low regularity). In our context - mapping meanings to discrete signals - regularity is maximized by a language of one-to-one mappings. For example where each position in the signal encodes one part of the meaning – position 1 → Subject – and each character in that position refers to only one possible subject – *A* in position 1 → Subject: Ollie – and is the only character ever used to refer to that subject. A maximally regular language encodes the same (part of) meaning with the same (part of) form every time, rather than affording a speaker a variety of ways to encode a meaning.

This kind of maximally regular system is intuitively compostional, given the meaning of a signal would be composed from the parts of meaning its characters map to and the position they're in (in line with Cann, 1993) but it's by no means the *only* kind of compositional system. To better characterise the space of possible languages in section 3.2.1 we introduce four kinds of variation - drawn from kinds of variation attested in natural language - and ways of quantifying each of them individually. Then in section 3.2.2 we look at some of the most relevant existing measures of 'compositionality' and discuss how they could be interpreted in terms of regularity. Results from a standard emergent communication model in section 3.3 show that every run results in a highly-generalizing (and therefore compositional) language but with varying degrees of variation. To better understand the relationship between variation and generalization we look over the time-course of training and find regularity is a strong predictor of how well a language generalizes early on but this effect goes away as the models approach ceiling i.i.d. generalization. We take this as an indication that while a language needs to be more regular than a random mapping in order to generalize, it doesn't need to minimize variation in order to do so – a point made clear by natural languages. At the end of training when the emergent languages have become sufficiently regular for the task at hand, whether one is more regular than another doesn't necessarily correspond to better generalization.

In a final set of experiments we look at how to decrease the amount of variation in an emergent language. Limitations on humans' memory and cognitive capacity are thought to be a driving force in the emergence of compositional structure and regularity in natural language (Hudson Kam & Chang, 2009; Kirby, 2001; Smith & Wonnacott, 2010). Learners with less memory are believed to regularize their input because they are more constrained in their ability to store low-frequency forms (Ferdinand et al., 2019; Newport, 1990). We reduce the capacity of our models by reducing the size of the hidden layers, and show that lower capacity models develop more regular languages, as predicted by accounts linking learner capacity and regularity in natural language and in line with previous work in this area (Resnick et al., 2020).

Model	i.i.d. acc		o.o.d. acc		synonymy		entanglement		freedom		homonomy		variation		topsim		posdis	
ideal					0.00		0.00		0.00		0.12		0.03		0.62		1.00	
random					0.99		1.00		0.99		0.99		0.99		0.00		0.00	
small	**97.54**	±0.49	**72.86**	±7.07	**0.46**	±0.02	**0.54**	±0.03	**0.49**	±0.03	**0.53**	±0.03	**0.50**	±0.03	**0.21**	±0.01	**0.24**	±0.03
Δ *best o.o.d.*					−0.20	±0.03	−0.42	±0.04	−0.19	±0.03	−0.18	±0.03	−0.25	±0.03	0.12	±0.01	0.22	±0.03
medium	**97.73**	±0.59	**82.13**	±3.62	**0.52**	±0.05	**0.60**	±0.07	**0.54**	±0.05	**0.58**	±0.05	**0.56**	±0.05	**0.19**	±0.02	**0.19**	±0.05
Δ *best o.o.d.*					−0.13	±0.05	−0.35	±0.07	−0.12	±0.05	−0.13	±0.05	−0.20	±0.05	0.10	±0.02	0.17	±0.04
large	**97.53**	±0.52	**81.33**	±3.03	**0.63**	±0.02	**0.80**	±0.04	**0.66**	±0.02	**0.69**	±0.02	**0.69**	±0.02	**0.14**	±0.01	**0.08**	±0.02
Δ *best o.o.d.*					−0.01	±0.03	−0.13	±0.05	−0.01	±0.03	−0.03	±0.02	−0.08	±0.02	0.03	±0.02	0.05	±0.02

Table 3.1: Mean accuracy and variation with 95% confidence interval across 20 runs, taken from the epoch with the best o.o.d. generalization performance, along with the change in measures $\Delta best$ between the least regular language that occurs between epochs 1 and 10 and the best generalizing one. Also included are the variation measures applied to a perfectly regular and a maximally variable language one as well as an average across all 4 variation measures. Two measures of regularity from previous work (topsim and posdis) are included in the grey cells.

3.2.1 Quantifying Variation

We quantify four kinds of linguistic variation two lexical Synonymy & Homonymy and two structural Entanglement & Word-Order Freedom. This is not intended to be an exhaustive list, but offers a starting point for thinking about linguistic variation in this context. Each of these measures is bounded between 0 and 1, where 0 indicates a perfectly regular language with no variation, and 1 represents a maximally variable language. For comparison we generate a maximally regular compositional language which scores near 0 across our measures, and maximally irregular non-compositional language (where each meaning maps to a unique randomly-generated signal) which scores near 1, as shown in table 3.1. Our task (described fully in section 3.3) asks models to map meanings to signals. With meanings comprised of roles - e.g. Subject, Verb, and Object - and semantic atoms which can occur in each role (e.g. Subject: *Ollie, Isla ...* Verb: *loves, hates,* ...). Prior work in this area sometimes refers to these as attribute-value pairs (see Lazaridou & Baroni, 2020, for a review including some mention of attribute-value pairs, p. 11). Similarly signals are comprised of positions (indices), and the character that occurs in each. We can frame linguistic concepts of variation in terms of how semantics (roles & atoms) map to signals (positions & characters).

All four measures start with a probability table that describes the mapping between meanings and signals probabilistically, in terms of a distribution over characters in each signal position given a semantic atom in a role. This encodes,

for example, how likely character 'A' is in signal position 1 given that 'Ollie' is in the subject role of the signal's meaning. We can quantify this as a straightforward conditional probability using maximum likelihood estimation, shown in equation 3.1. We estimate this for every atom ($\forall atom_{r,i} \in A_r$) in every role ($\forall r \in R$), looking at every character ($\forall char_{p,j} \in C$) in every position of every signal ($\forall p \in P$).

$$\mathbb{P}(char_{p,j}|atom_{r,i}) = \frac{count(char_{p,j}, atom_{r,i})}{count(atom_{r,i})} \tag{3.1}$$

The resulting tensor describes how often each letter occurs in a position, given a certain atom in a role in the meaning (like Subject: Ollie)[1]. This tensor has dimensions semantic roles × semantic atoms × max signal length × characters[2], where the last axis is a probability distribution over all possible characters in a given position - here denoted by $\mathbb{P}(char_p|atom_{r,i})$.

Synonymy & Homonymy: Synonymy is minimised when each atom in a meaning maps to a single character in a position. Homonymy is minimised when each character in a position maps back to a single atom (Hurford, 2003). While a perfectly regular compositional language minimises these, natural language is rife with both synonymy and homonymy (e.g. 'loves', 'adores', 'fancies' all map to approximately the same concept; the homonymous 'bank' maps to a financial institution, the act of turning a plane, and the land at the side of a river). One-to-many mappings (synonymy) aren't a problem for compositionality, as each different synonym can still be composed with the rest of a signal. Similarly many-to-one mappings (homonymy) can be used compositionally, with meaning disambiguated by context. Homonyms in natural languages also tend to be contextually mutually exclusive meaning the ambiguity they introduce at a system level tends not to impair communication (thinking of 'bank' - it's rare to cash a check while piloting a plane into the side of a river).

In our setting synonymy is how many different characters can refer to an atom in a role. For example when $r = Subject$ and $atom_{r,i} = Ollie$ how many characters have non-zero probability in each signal position? A perfectly regular language where 'Ollie' is always encoded by 'A' in position 1 would have a probability of 1.0 on 'A' in position 1. A maximally variable language would have a uniform

[1]Here we use semantic roles given the meanings are sentences, this can be generalised to any analogous attributes a dataset exhibits.

[2]For all experiments reported here these values are $3 \times 25 \times 6 \times 26$

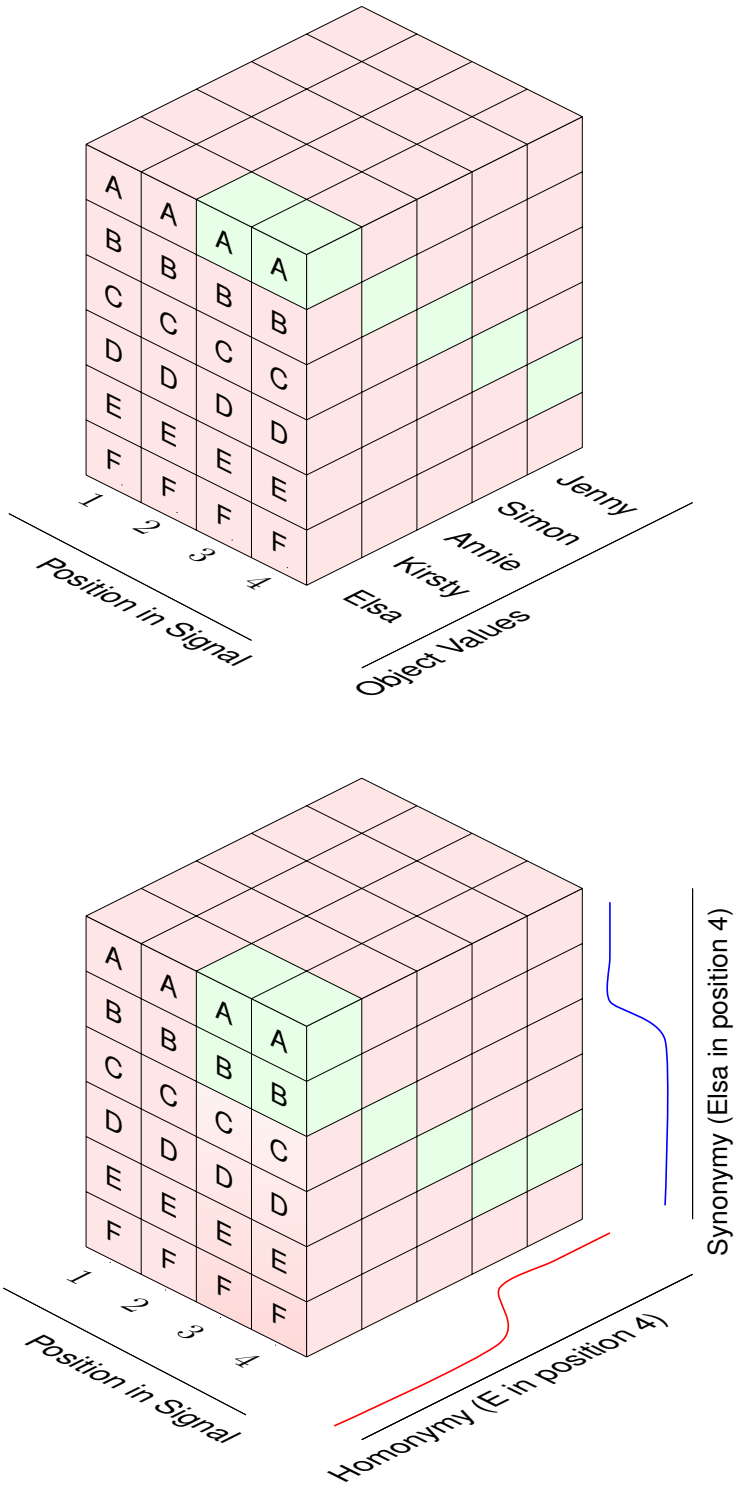


Figure 3.1: A depiction of the probability tensor built with equation 3.1 where $r = Object$. Green indicates high probability and red low. **(Top:)** A perfectly regular language, Elsa is always encoded by 'AA' in the final two positions, Kirsty by 'BB' etc. **(Bottom:)** The same cube is shown (object labels removed) for a language with basic synonymy (Elsa can be encoded by 'A' or 'B') and homonymy (Jenny and Simon are both encoded by 'E'). We quantify the degree of synonymy by taking the entropy of each column (equation 3.2) and the degree of homonymy by taking the entropy of each row (equation 3.3)

distribution over all characters. We can take the entropy over characters in a position $\mathcal{H}(char_p|atom_{r,i})$ as a measure of synonymy in that position (illustrated in figure 3.1). We take the position with the lowest entropy as a lower-bound estimate of synonymy for that $atom_{r,i}$. We bound this measure dividing it by the entropy of a same-sized uniform distribution $\log(n_{char})$. The resulting quantity is an efficiency (Shannon, 1948) where a uniform distribution scores 1.0 and a one-hot distribution scores 0.0. A language with no synonymy where each atom is encoded by a single character in a position achieving close to 0, and maximal synonymy where any character can refer to each atom achieving close to 1 (shown empirically in table 3.1). The synonymy of an entire language ($\mathcal{L}$) is obtained by averaging across all atoms in a role, then across all roles.

$$Synonymy(atom) = \frac{\mathcal{H}(char|atom)}{\log(n_{chars})} \tag{3.2}$$

We measure homonymy in a similar way, looking at how many semantic atoms a character in a position can refer to. As depicted in figure 3.1 this is akin to applying the synonymy measure to a different axis of the probability tensor $\mathbb{P}$. We estimate $\mathbb{P}(atom_r|char_{p,j})$ to get a distribution over atoms given characters in a position[3]. To get a lower-bound estimate of language-level homonymy we take the position with the lowest entropy over atoms, again bound between 0 and 1, then average across all characters and roles. When the resulting value is close to 1 each character maps to every atom. Approaching 0 each character uniquely refers to a single atom.

$$Homonomy(char) = \frac{\mathcal{H}(atom|char)}{\log(n_{atoms})} \tag{3.3}$$

Word Order Freedom is minimized when each role in the meaning is always encoded in the same position(s) in the signal, resulting in a single canonical word order. Looking at a language like Basque we see that a compositional language can support a number of different grammatical word orders (Laka Mugarza, 1996), with at least two equivalently valid translations of 'Ollie saw Ernest:' *Ollie Ernest ikusi zuen, Ollie ikusi zuen Ernest.* Even in English which has relatively strict word order we see processes like topicalization that result in alternate orders that are equally acceptable *Let's go down to the lake for some fun; For some fun, let's*

[3]For simplicity we re-normalize $\mathbb{P}$ to create a probability distribution over atoms in a role which is equivalent to directly computing $\mathbb{P}(atom_r|char_{p,j})$ see appendix for further discussion.

go down to the lake, or even more commonly dative alternations (Chomsky, 1957) like *Ollie gave Orson a book; Ollie gave a book to Orson.* While many languages have some constraints on word-order, even when there is maximal word order freedom the resulting language can still be clearly compositional, with characters encoding the meaning and their order conveying little information.

A language with free word order is equally likely to encode any $role \in R$ in any position, while a maximally regular language always encodes atoms from the same role in the same position(s). If a given $atom_{r,i}$ is not encoded in a position we expect its distribution over characters to be roughly uniform. So we can take the entropy for each position $(\mathbb{P}(char_p|atom_{r,i})) : \forall p \in P)$ (also computed as part of equation 3.2), and average across all atoms in that role $\forall i \in A_r$. If all the atoms in a role are encoded in the same position the distribution resulting from the mean will be non-uniform, with some positions having lower mean entropy than others (illustrated in figure 3.2).

$$\mathbb{F}(role_r) = \frac{1}{|r|}\sum_{i=1}^{|r|}\mathcal{H}(char|atom_{r,i}) \tag{3.4a}$$

To get a lower-bound estimate of the language-level word-order freedom we take the minimum from the mean distribution $\mathbb{F}(role_r)$ and bound between 0 and 1, then average across all roles:

$$Freedom(\mathcal{L}) = \frac{1}{|R|}\sum_{r=1}^{|R|}\frac{min\,\mathbb{F}(role_r)}{\log(n_{char})} \tag{3.4b}$$

Entanglement is minimised when each role is encoded in different positions in the signal. While a dis-entangled language is likely compositional, consider the English past tense form of 'go.' 'Went' is irregular, encoding action and tense together, in contrast to the hypothetical regular form 'goed' where action and tense are encoded in separate parts of form (Anderson, 1992, p. 55). Despite this we can go on to use the entangled form 'went' compositionally in a sentence: *Ollie went down to the shore* (for discussion O'Donnell, 2015, p. 105). While maximal entanglement where every role is encoded in every position would be non-compositional, the existence of even a high degree of entanglement does not preclude compositionality, given the entangled forms can be straight-forwardly recomposed with others. We can quantify this by seeing if two roles are consistently

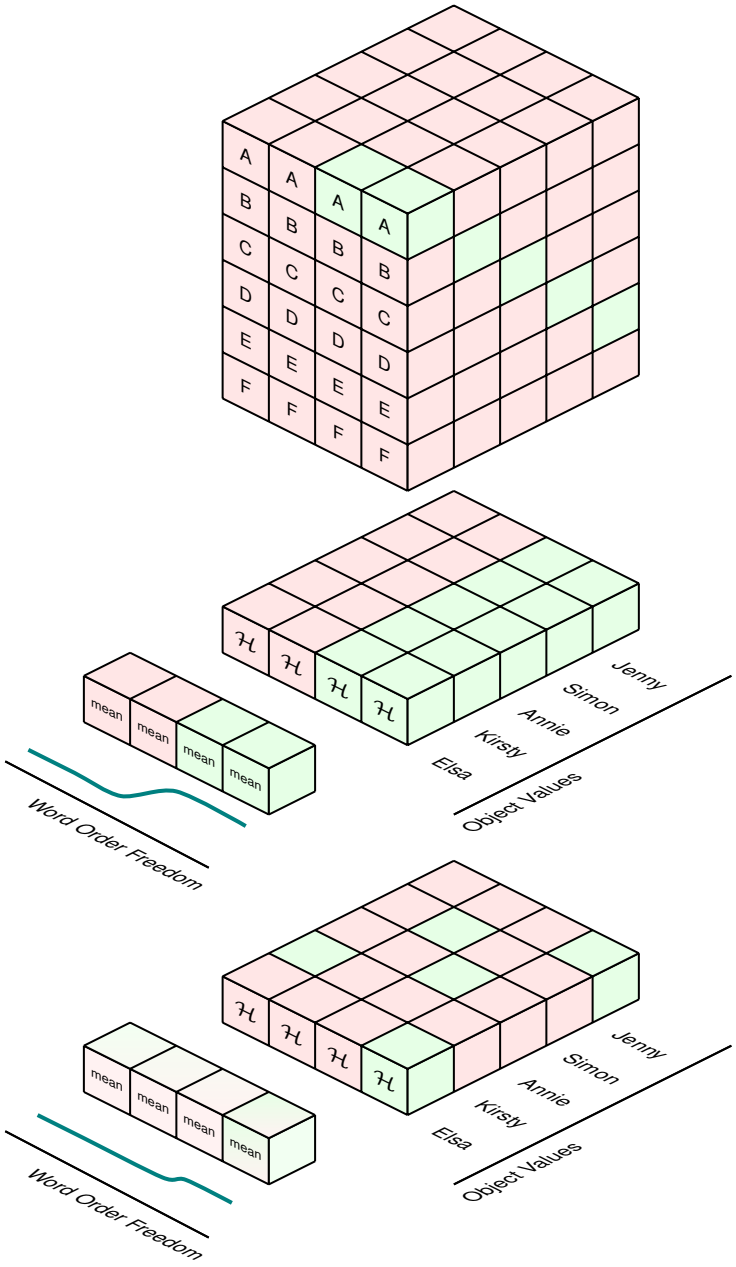


Figure 3.2: The Freedom measure applied to a regular and variable language. At top the cube from figure 3.1 red indicates low probability (high entropy), and green high probability (low entropy). Directly below it is the entropy of each column - when we take the mean of these column entropies across atoms in a role a non-uniform distribution indicates semantic roles are disentangled.

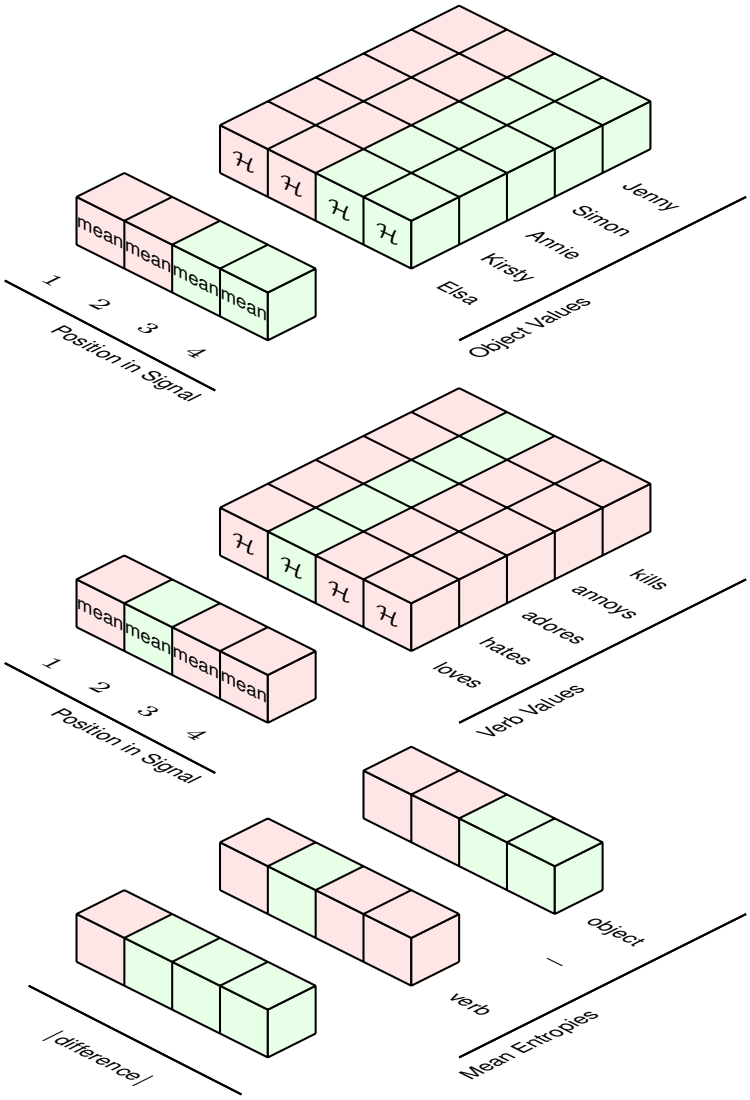


Figure 3.3: Column entropies for two different roles, object and verb. The mean of both are taken and then compared by taking the absolute value of the difference of the two averaged entropies. If two roles are encoded in the same part of the signal then their difference will be close to zero: they both have consistently similar in the same parts of the signal. The maximum of the difference is then divided by the maximum of the mean distribution for the respective roles before they were subtracted - this bounds the measure between 0 and 1 and makes the quantity reflect the proportion of overlap between roles.

encoded in the same (or different) positions. We compare the means $\mathbb{F}(role)$ from equation 3.4a for each possible pair of roles $r_i, r_j \in {}^{R}C_2$ by taking the magnitude of their difference, if two roles are encoded in the same position the result will be close to zero. If the roles are maximally disentangled then the result will be close to the $max(\mathbb{F}(role_i), \mathbb{F}(role_j))$ for that position. To get a lower bound estimate of two roles' entanglement we take the maximum of the difference and divide by the pre-difference max. When the resulting value approaches 0 all roles are mapped to different parts of the signal, as it approaches 1 all roles are encoded in the same positions (illustrated in figure 3.3).

$$Entanglement(\mathcal{L}) = 1 - \frac{1}{|{}^{R}C_2|} \sum_{r_i, r_j}^{{}^{R}C_2} \frac{max\Big(|\mathbb{F}(role_i) - \mathbb{F}(role_j)|\Big)}{max\Big(\mathbb{F}(role_i), \mathbb{F}(role_j)\Big)} \tag{3.5}$$

3.2.2 Existing Measures

Topographic Similarity (Topsim) (Brighton & Kirby, 2006) has been used as a measure of compositionality in a wide array of contexts (e.g. Kirby et al., 2008; Lazaridou et al., 2018; Smith et al., 2003). It assumes that in a compositional system where a whole signal is composed from reusable parts, similar meanings will map to similar signals. This can be assessed by measuring the correlation between pairwise meaning-distances and edit distances between their associated signals: a perfectly regular compositional language without variation achieves a correlation score close to 1, while a non-compositional (random) mapping between meanings and signals achieves a correlation close to 0. While languages that score highly are likely to be compositional synonymy and word order freedom can reduce the score for this measure, as they can result in similar meanings having dissimilar signals. Synonymy can mean two meanings with the same subject encode it with different characters. Freedom can mean signals for similar meanings with different word orders have high edit-distance despite containing many of the same letters.

Posdis & Residual Entropy (Chaabouni et al., 2020) & (Resnick et al., 2020) provide entropy-based measures of 'compositionality.' Posdis captures the extent to which each position of the signal univocally refers to a role in the meaning (e.g. subject, object, verb) and looks for each signal position to refer to only one role. This is similar to what our entanglement measure assess (though computed differently). Similarly, residual entropy assesses the degree to which

a sub-string of the signal encodes a single atom in a role (e.g. Ollie in the Subject) and is minimized when a sub-string refers to only one atom in a role. This requires there to be minimal homonymy and entanglement in a subset of the signal (across 1 or more positions), with each unique sub-string in those positions referring to only one atom in a role. As discussed above natural language shows us that even a high degree of homonymy and entanglement in a language doesn't preclude its compositionality. We show empirically in table 3.1 that a maximally regular language maximizes topsim and posdis while minimizing residual entropy (for brevity residual entropy results are deferred to appendix A.6). Like topsim, languages that score highly on these measures are very likely to be compositional - the issue is that they take some kinds of variation as evidence of non-compositionality.

3.3 Methods

Models We implement a reconstruction game with a sender network and receiver network. The overall architecture used is intentionally similar to Chaabouni et al. (2020) and Resnick et al. (2020) and Guo et al. (2021) to allow comparison of results. The sender network is comprised of an embedding layer, linear layer, and a GRU (Cho, van Merrienboer, et al., 2014) - the receiver architecture is the inverse. A linear layer is used as the input is of fixed length, so can be presented at once as a one-hot encoding - while a GRU spells out the variable-length signal a character at a time. The maximum signal length used here is 6, with 26 characters available to the model in each position. The sender is optimized using REINFORCE (Williams, 1992) due to the discrete channel, while the receiver is optimised using ADAM (Kingma & Ba, 2014). Models are implemented using pytorch (Paszke et al., 2019), and make use of portions of code from the EGG repository (Kharitonov et al., 2019). Full hyperparameters for the experiments presented here can be found in appendix A.8.1.

Data The sender is shown examples drawn from a meaning space of two place predicates (e.g. *Ollie loves Osgood*) generated using a context free grammar, with three roles: subject, verb, and object and 25 atoms per role, resulting in a total of 15625 examples. This is equivalent to the attribute, value setup used in previous work (Chaabouni et al., 2020; Resnick et al., 2020). Data is divided into 4 splits

for training: 60%, validation 10%, i.i.d. testing 10%, and o.o.d. testing 20%.

O.O.D. Evaluation Previous emergent communication work typically evaluates generalization on an in-distribution held out test-set. In order to better align our findings with the broader literature on compostional generalization in neural networks (e.g. Kim & Linzen, 2020; Lake & Baroni, 2018) we implement a version of the maximum compound divergence (MCD) algorithm from Keysers et al. (2020), and report results for both in-distribution generalization, and out-of-distribution generalization to an MCD split. Additionally we use an O.O.D. split because models often converge to ceiling i.i.d. performance, which potentially makes it difficult to look for correlations between generalization performance and attributes of the language, like regularity. For our *small* model i.i.d. performance 95% confidence intervals are $\pm 0.49\%$ while o.o.d. performance is $\pm 7.07\%$, allowing a broader range of values with which to look for correlations (we include the same analyses on i.i.d. performance in appendix A.7 and in practice i.i.d. and o.o.d. results are very similar).

Capacity We look to see if models with less capacity arrive at more regular languages than their larger counterparts as predicted by work in natural language (e.g. Hudson Kam & Chang, 2009). We vary model capacity by varying the size of the hidden layers used by the model reporting and comparing results of three different model sizes *small, medium,* and *large* with hidden layer sizes 250, 500, and 800 respectively.

Results and discussion

Our results for all model sizes are summarised in table 3.1. As stated in section 3.1 a language must be compositional in order to generalize, in line with previous work in this area (Kottur et al., 2017) and in linguistics (Brighton & Kirby, 2006). All versions of our model get near ceiling i.i.d generalization and robust o.o.d. generalization indicating a compositional system. Compositionality and variation are related, but distinct; while a system needs to be more regular than a completely random mapping in order to generalize compositionally it does not need to be perfectly regular. Natural languages show us that a system can support a high-degree of variation while remaining compositional. In line with this in all

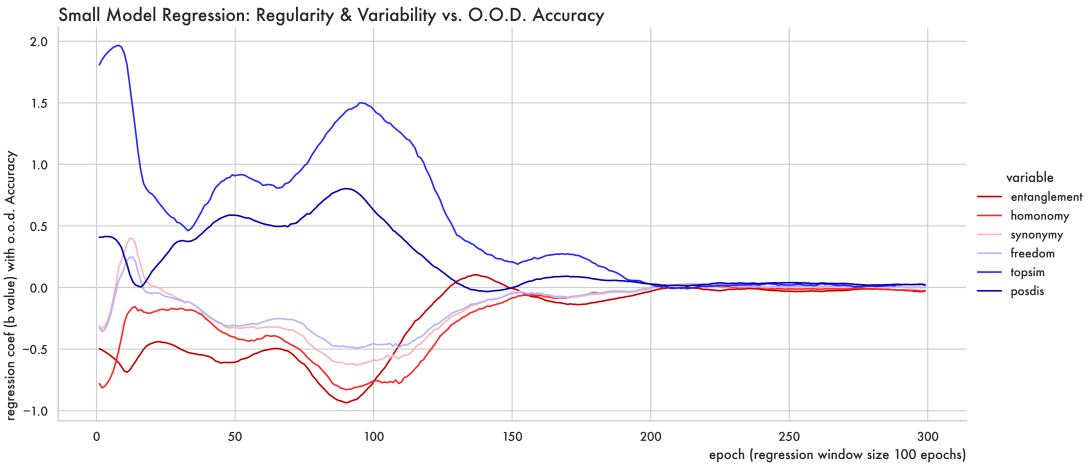


Figure 3.4: A model is fit to a sliding window of data from 100 epochs at a time across 20 initializations between o.o.d. accuracy and each measure of variation. Shown are the regression coefficients (b values) of our four measures of variation, and two previous measures of regularity (topsim and posdis) with o.o.d. generalization accuracy for the *small* model for each window.

conditions of our model the language that emerges is substantially more regular than a random mapping, but more variable than a perfectly regular language of one-to-one mappings.

The relationship between regularity and generalization We use linear mixed effect models to evaluate the relationship between our four measures of variation and o.o.d. performance, fitting a model on rolling windows covering the time course of training (implementation details in appendix A.2). The resulting regression coefficient (b value) for a window indicates how strong a predictor our measures are of generalization performance over that period of training. As shown in figure 3.4 early on a language's regularity is a strong predictor of how well it generalizes, but later in training this effect goes away. This is consistent with the idea that some regularity is needed for generalization, but maximal regularity is not required. Later in training, as a language emerges that is *regular enough* to succeed at the task (achieving ceiling i.i.d. generalization performance), the relationship between regularity and generalization trends toward non-significance. Supporting this we see languages become more regular over time with a negative relationship between training step and variation ($b = -0.038, p < 1e-10$) – in table 3.1 we also see that in every condition the model decreases the variation

in its language between early training and the best generalizing epoch indicated by a negative value for Δ *best o.o.d.*. A limitation of these results is that the language for every run is still highly-variable (with the lowest mean variation score of any run being 0.43), possibly because the task here is quite simple in comparison to compositional generalization datasets in other domains (e.g. Kim & Linzen, 2020). As languages approach maximal regularity, regularity may again be a strong predictor of generalization performance - but given none of our models approach minimal variation this remains an open question (further discussion of these results in appendix A.4).

How can a variable language still be compositional? Figure 3.5 helps us to understand what these highly variable but robustly generalizing languages look like. It visualizes the word order for the run of our *small* model with the *highest* word order freedom - meaning all other runs of that model exhibit even stricter word order. It shows that while the language is still much more variable than a perfectly regular one (this language has $freedom = 0.57$, a compositional language with fixed word order has $freedom = 0$), it nonetheless exhibits a high degree of word order regularity, with verbs most likely encoded at the start of the signal, subjects in the middle, and objects at the end, but with each individual atom sometimes being encoded slightly differently. Given compositionality requires the meaning of a whole to be a function of its parts the pattern seen here where each role is encoded in part of the signal appears to meet that threshold despite its high variation.

Capacity effects regularity with an increase in the number of trainable parameters resulting in an increase in variation across all measures with $large$ arriving at significantly more variable languages than *small* or *medium* ($p < 0.05$). Spearman correlations show model size does not correlate significantly with o.o.d. accuracy ($p = 0.24$) but correlates with synonymy ($r = 0.67$), word order freedom ($r = 0.69$), entanglement ($r = 0.68$), and homonymy ($r = 0.68$) indicating larger models develop more variable languages (all of which are significant $p < 0.00001$). This result is in line with work that points to constraints on human cognition as a key driver of regularization in natural language, suggesting that similar factors shape the regularity of emergent communication in neural networks. Previous work studying the effect of network capacity on emergent languages (Resnick et al.,

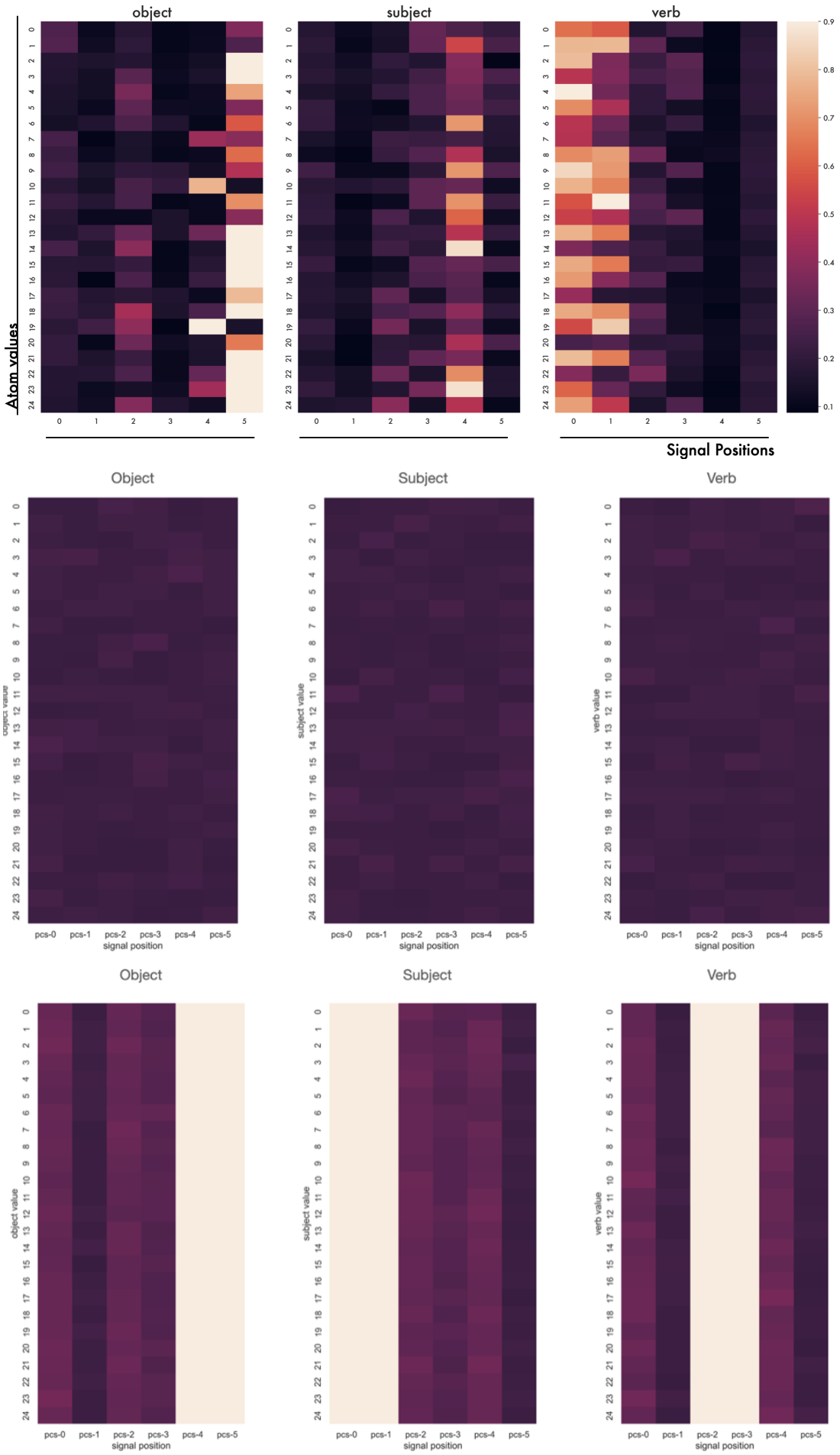


Figure 3.5: *Caption Opposite*

Figure 3.5: Plots showing the max from the distribution over characters for each atom in each position, with a plot for each separate role (object, subject, verb). x axis: positions, y axis: id for each $atom_i \in A_r$. Shown to the left are these plots for the synthetic ideally regular compositional language (with SVO order), and the maximally variable random mapping. The large plot shows data from the run of the $small$ model with the *highest* variation This run's variation scores: $freedom = 0.57, entanglement = 0.61, homonymy = 0.61, synonymy = 0.51, topsim = 0.28, posdis = 0.26$

2020) found that while most model sizes could generalize well, larger models could do so using a 'non-compositional code' indicated by a higher residual entropy measure (which has similarities to our measures of homonymy and entanglement). This is consistent with our results, although we believe this indicates that larger models develop a language characterised by greater variation rather than non-compositionality (residual entropy scores for our model can be found in appendix A.6).

Framing prior results in terms of regularity Existing measures (topsim and posdis) correlate negatively with model size ($r = -0.63$, $r = -0.71$) strongly suggesting that rather than tracking compositionality these measures implicitly track the degree of regularity in a language, especially given that the magnitude of their correlation coefficient is similar to that of our measures that explicitly assess variation. This helps us to interpret results suggesting compositionality doesn't correlate with generalization: if these measures assess regularity instead we know a wide array of languages can be regular enough to generalize well without needing to maximize regularity to do so.

3.4 Conclusion

Neural networks reliably arrive at compositional languages when natural language-like variation is taken into account. Previously these languages' compositionality has been assessed on the basis of their regularity, but natural languages show us a system can be rich with variation while retaining the generalizability that makes compositionality so desirable. Similar to natural language the capacity of learners is a key driver of the degree of regularity that emerges. By accounting for

variation we can see striking similarities between the structure of the languages that emerge and structures in natural language.

3.5 Epilogue

The experiments in this chapter lay out a couple of key premises that are built on moving forward. Namely that

- Conditional entropies can describe variation in a mapping
- Applying conditional entropies at different levels of abstraction, like lexical or structural, can shed light on how a mapping is structured.
- Capacity has a regularising effect on model behaviour, with smaller models producing more regular languages.
- Generalisation performance isn't directly related to regularity - a system doesn't need to minimise variation in order to generalise.

The model in this chapter is intended to have a clear relationship to communication in humans. As a result the structures discussed are introduced in terms of their linguistic analogs. While the remainder of the thesis focusses on mapping structure in a more general way, I think starting with linguistic examples arms us with intuitions for what different structural properties are good for. Synonymy increases the complexity of a language, by increasing the number of forms it contains. However this also increases the complexity that a language can faithfully describe. By having different forms for different contextualisations of a concept we end up being able to convey fine-grained contextual information. Homonymy makes the system as a whole more compressible, by reducing the number of forms - but does so with a cost of ambiguity. Reusing the same form for different meanings requires reconstructing which meaning a form maps to from context. While this isn't an issue if homonyms are contextually mutually exclusive - often the case in natural language - when homonyms appear in the same context this ambiguity can be a problem. In the next chapter synonymy and homonymy are generalised

to *Variation* and *Disentanglement*[4], but at a high-level intuitions about the utility and costs of each kind of structure broadly hold.

Having started with an illustrative set of experiments, using a small model and dataset with discrete mappings, we turn now to mappings less clearly related to human language. The next chapter takes a step towards studying mappings learned by deep learning models, instantiating information theoretic measures for studying relationships between discrete and vector spaces that can be applied to models trained on more complex tasks than used in this chapter. Despite the shift in domain we'll continue to look at whether linguistic intuitions about different mapping structures hold, and the role that capacity plays in conditioning them.

[4]Homonomy is directly related to entanglement, but as noted previously to better align the measures with the divergence they use we opt to quantify disentanglement, which for us is equivalent to 1-entanglement.

Chapter 4

Regularity, Variation & Disentanglement in Vector Space

Discrete → Dimension-Wise Continuous

> The Era of Having Famous Painter Parents [redacted] of Making a Myth of Oneself [redacted] of Patenting International Klein Blue [redacted] of Needing More and More of a Crowd [redacted] of One's Friends' Suspicion That One Had Arranged to Vanish Not Actually Died [redacted]
>
> *- Anne Carson*

Chapter 1 discussed how deep-learning models put up impressive performance across a broadening array of tasks. Despite this we have a limited understanding of how they learn to represent information, and what information structures are desirable. In the experiments in this chapter, we look at models trained on large-scale semantic parsing tasks (like mapping 100,000 questions to SQL queries that answer them) and measure structure between their inputs and representation spaces. We want to see if a model's vector representations develop the same kinds of system-level structures present in the language data on which they're trained. While there's a growing body of work on *interpretability*, this area tends not to focus on models' representation spaces directly. Instead, approaches

often focus on behavioural evidence using a model's outputs to reason about what may be happening internally — like looking at when models assign higher probability to grammatical sentences (Marvin & Linzen, 2018). Another popular approach is probing (Hupkes et al., 2018; Veldhoen et al., 2016) - also called diagnostic classification - which trains a classifier to predict properties based on representations from a trained model. If it's possible to predict part-of-speech tags for a sentence from a model's representation of that sentence, it suggests the model has learned something about syntax. MDL probing (Voita & Titov, 2020) extends this line of work, by also quantifying how complex of a classifier is required to predict the diagnostic labels. While this kind of interpretability has yielded valuable insights it remains removed from directly assessing structure in representation space - relying on model performance on downstream tasks, or the performance of a diagnostic classifier.

A major reason for the limited work on representations themselves, is that representations are high-dimensional vectors about which humans tend not to have strong intuitions. In their work on probing Hupkes et al. (2018) note that earlier work would visualise models' hidden states and look for patterns in an effort to make representation spaces more intuitive to analyse, it's an approach that proves difficult to scale:

> [...] studying hidden layer activations is an interesting puzzle and can – especially for relatively low dimensional network such as ours – give pointers to which aspects should be studied in more depth. However, it is hard to draw definite (and quantitative) conclusions, and the usefulness of the method decreases with higher dimensionality of the networks. [...] Disentangling the behaviour of networks through visual inspection of activations is searching for a needle in a haystack.
>
> — Hupkes, Veldhoen, and Zuidema (2018) | p.918

What we need is a method for talking about representations directly, quantitatively, that scales to the kinds of models in use today whose latent spaces easily exceed 1000s of dimensions. The next two chapters propose methods for doing exactly this, information theoretically, treating a model as a mapping between inputs and representations, and quantifying the degree of our 3 base structural properties in a mapping — regularity, variation, and disentanglement (one-to-one, one-to-many, many-to-one).

The major challenge in this approach is quantifying entropy in vector space. Entropy estimation for continuous spaces is a notoriously challenging problem (Paninski, 2003). In Chapter 2 I looked at discrete entropy, where each event in a categorical distribution refers to the probability of something discrete, like the probability of a word. Now though, we want a probabilistic interpretation of vector space, where probabilities refer to how likely it is for representations to occur in a certain region of space. Shannon proposed differentiable entropy as the continuous analog of discrete entropy, swapping the summation over events with an integral - unfortunately as Jaynes (1957) notes this quantity is not in-fact the true continuous analog of discrete entropy and lacks many of the properties which make entropy desirable[1]. Most critically, differential entropy $\mathcal{D}(Z)$ is unbounded ($-\inf < \mathcal{D}(Z) < \inf$) making it difficult to interpret, and is not invariant to linear transformations, meaning two uniform distributions can have substantively different differential entropies depending on what region of space they cover. There are also practical challenges in estimating the quantity given the difficulty of integration.

Chapter 5 engages with this problem at length, discussing limitations of *differential entropy* and proposing a new theoretically grounded approach to estimating entropy of continuous space. The remainder of the current chapter takes initial steps in that direction, opting to discretise representations, cutting attested space into bins and using bin probabilities to estimate discrete entropy. This approach is used in previous work looking at smaller networks trained on simpler tasks (Goldfeld et al., 2018; Saxe et al., 2019; Shwartz-Ziv & Tishby, 2017), and is instantiated similarly here. While this approach does allow meaningful quantification of structure in continuous space it has a number of drawbacks discussed in the next chapter. Most relevant here is that it is exceedingly memory intensive, limiting it's applicability to contemporary models. In this chapter we circumvent this by performing dimension-wise discretisation; discretising and analysing dimensions one at a time before aggregating across them. While this limits our ability to track cross-dimensional dependencies, it does allow the approach to be applied to 256 dimensional spaces, and enables direct comparison between models of different dimensionalities. It also allows us to relate the analysis

[1]According to Jaynes (1957) Shannon merely assumed use of an integral was the correct continuous analog of discrete entropy (it is not), and did not derive Differential Entropy from Discrete Entropy. A rare moment of fallibility from Shannon who introduced Information Entropy in his masters thesis.

here to the analysis performed in the previous chapter.

The previous chapter looked at structure in a mapping between meanings and signals, where signals were a sequence of symbols. Signals have a maximum length (how many symbols can occur in sequence) and an alphabet (how many different symbols are possible at each position). Dimension-wise discretisation treats a representation as a sequence of random variables, where number of dimensions is analogous to maximum signal length, with number of bins as the alphabet size. The approach in this chapter therefore acts as a middle ground between the preceding chapter which considers structure in discrete signals, and the next chapter which works to quantify entropy while remaining more faithful to continuous space. In all three cases I show how structure develops over the course of training, and that representational systems end up looking - in many regards - a lot like natural language: regular enough to generalise, but retaining substantial variation to represent the contextual complexity of the world their training data describes.

The remainder of this chapter is based around a paper **Representations as Language: An Information Theoretic Framework for Interpretability** that appeared at the International Meeting of the Cognitive Science Society in 2024. Authors are myself and Kenny Smith - Kenny and I conceived the experiments together which I then implemented and wrote up - Kenny gave writing feedback prior to submission to the conference. The paper is presented here minimally changed from the conference version that underwent peer-review. Changes are largely related to formatting to make the content more readable outside of the original conference paper template, and to make notation consistent across different chapters. Here I've also added some visualisations of this chapter's measures of structure applied to example distributions in 2-dimensions to help give an intuition for how they work. This addition comes just after the measures are defined and added text is marked, like here, by horizontal lines above and below.

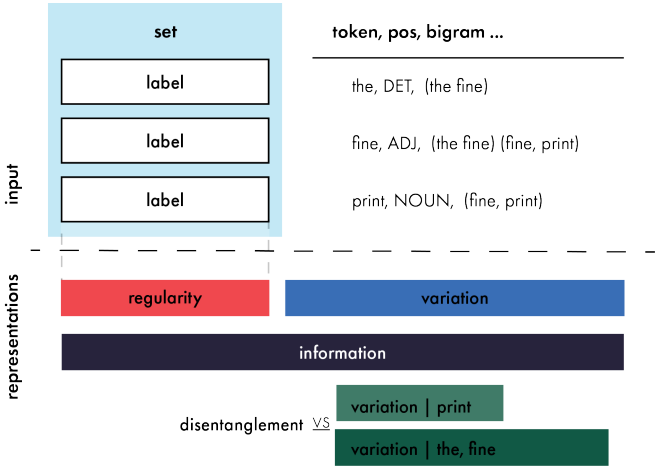


Figure 4.1: a depiction of basic quantities we measure and how they relate to each other. We measure structure in the mapping between labels for the dataset, and latent representations inside of a transformer. Here some example labels are given for the sentence "the fine print."

4.1 Representations as Language: An Information Theoretic Framework for Interpretability

Deep-Learning models achieve remarkable performance across a broad range of natural-language tasks (Vaswani et al., 2017), but we still have a limited understanding of the learning process they undertake, and how they come to represent information so effectively. This is in part because these models are black-boxes (Shwartz-Ziv & Tishby, 2017; Tishby & Zaslavsky, 2015). They learn representations of their training data that are high-dimensional vectors, gigantic lists of numbers that are hard to interpret. While there is a growing body of work on interpretability, offering techniques for predicting what is encoded in a model's representations (e.g. Pimentel et al., 2020; Voita & Titov, 2020), there's still lack of clarity about how representations themselves are structured, how that structure emerges, and what kinds of structures are desirable.

Central to language's ability to generalise is its regularity, exemplified by syntactic structure (Partee et al., 1995), which allows predictable & regular encoding of meanings across the entire system. Languages are also rich with variation which can make them more expressive (Hurford, 2003) and structured ambiguities that can make them more compressible (Piantadosi et al., 2012). Do the representations learned by a transformer model Vaswani et al., 2017 exhibit

similar *system level-structures*? To answer this we look at the representations that emerge over the course of training as a kind of language in their own right. At a high-level we can think of language as a mapping between spaces, like between meaning and form (de Saussure, 1916). A multi-layered neural model needs to learn to map a sentence to a vector representation that later layers can successfully map to the output; encoder-decoder models (Cho, Van Merriënboer, et al., 2014) even more explicitly use separate parts of a model to map in and out of vector space. We draw an analogy between these two mappings, quantifying different kinds of regularity and variation in a model's mapping between inputs and representations. While there has long been interest in the kinds of representations learned by deep-models (Bengio et al., 2013; Locatello et al., 2019), there has been little work quantifying systematic structure in the representations learned by transformers or relating them to the kinds of structures that characterise natural language. It's worth noting our approach is in contrast to some existing work that draws parallels between model weights and formal languages (like trying to infer functions or 'source code' from model weights; as in Elhage et al. (2021)) — we think an approach grounded in natural language is more scalable and better suited to characterising the kinds of systematic structure, variation, & ambiguity that can emerge in deep-learning models, especially those trained on data from natural languages.

We introduce a novel information-theoretic framework for assessing whether the representations learned by a model are systematic. In order to do this we first discretise vector representations into a sequence of symbols, then quantify 4 properties of the learned mapping from sentences to symbols: the degree of compression, regularity, variation and disentanglement. By doing this at different levels of abstraction we show when lexical and syntactic information are learned. We identify two clear phases of training, the first characterised by the model rapidly learning to disentangle and align representations with token and part of speech information, the second (far longer) phase of training characterised by representations becoming more robust to noise. During this second phase models compress their representations, with larger models compressing considerably more; at the same time, generalisation performance begins to slowly improve, showing a link between robustness to noise and generalisation. Finally we discuss what kinds of representational structure are desirable, using our measures to predict which models will perform best on a generalisation set.

4.2 Methods

Our experiments use a Transformer (Vaswani et al., 2017) encoder-decoder model, with a two layer encoder, and single layer decoder. The model's encoder maps each input sentence to a vector representation, then the decoder uses this representation to generate the corresponding output, in our case a formal semantic representation of the input sentence. We look at the encoder's mapping between sentences and representations, and quantify the degree to which it exhibits systematic structure. We train each model (from scratch) on two different semantic parsing datasets, designed to evaluate a model's ability to systematically generalise: SLOG (B. Li et al., 2023) where the task is to generate lambda expressions for a sentence, and CFQ-MCD (Keysers et al., 2020) where questions about movies need to be mapped to SQL queries that answer them. Both of these datasets come with an out-of-distribution generalisation set containing examples generated by the same grammar as the training data, but purposefully designed to be challenging. We also look at whether the capacity of a model affects the kinds of structure that emerges, training three different model sizes (with hidden dimensions of 64, 128, or 256).

4.2.1 Estimating Entropy in Vector space

Shannon Entropy describes the amount of information contained in a random variable (Shannon, 1948). While methods exist for estimating entropy of continuous variables, these approaches are difficult to compare across representational spaces and often require strong assumptions about the underlying probability distribution (Jaynes, 1957). Instead we discretise the hidden representations into a sequence of random variables, enabling us to directly estimate the Shannon entropy of our latent space. Our method is analogous to converting each vector into a sequence of discrete symbols, with a symbol for each dimension of vector space. Previous information theoretic analyses of deep learning have performed a similar estimation (e.g. Shwartz-Ziv and Tishby (2017), Saxe et al. (2019)), although it's been noted this approach is more reflective of non-uniformity, like clustering behaviour, than it is of the true entropy of the space (Goldfeld et al., 2018). For our purposes identifying the degree to which a variable is uniformly distributed, or tightly clustered is sufficient to draw substantive conclusions.

For a given vector in a set of vectors $v_i \in V$ with dimensions $d \in D$ we cut

each dimension V_d, into N equal-width bins between the attested maximum and minimum values of that V_d. This enables a straightforward maximum-likelihood estimate of the entropy of V by counting the frequency of each bin and normalising by number of representations in V. Resulting bin probabilities $p(V_{dn})$ are used to estimate the entropy of each dimension, then averaged across dimensions to give us an overall estimate of the dimension-wise entropy of V.

$$H_{dw}(V) = \frac{1}{|D|} \sum_{d}^{D} \sum_{n}^{N} -p(V_{dn}) \log(p(V_{dn})) \tag{4.1}$$

On the right in 4.1 is the equation for shannon entropy (Shannon, 1948), as this estimate is an approximation we also use the Miller-Meadow correction in order to smooth the estimate based on sample size and improve its accuracy (G. Miller, 1955). No method of estimating discrete entropy in continuous spaces is perfect (see Paninski (2003) for extensive discussion), but our estimator is invariant to linear transformations while making minimal assumptions about the underlying distribution. Note that while in the results presented here we estimate entropy per dimension we can in principle just as easily estimate entropy per pair or set of dimensions (akin to modelling at the unigram vs n-gram level); in practice the memory demands of the full-discretisation approach used here and in previous work make this intractable. Our use of a dimension-wise estimate simplifies our analysis but limits its ability to track cross-dimensional dependencies. Although analysis at the dimension level allows us some insight into the role of different subspaces of representational space, by letting us break estimates down dimension by dimension.

4.2.2 Measuring Structure

We are interested in whether a model's representations become systematically structured during training, reflecting the system-level structures of the data they're trained on. Using our entropy estimator we assess 4 different quantities at different levels of abstraction, which allow us to describe the degree to which the representations a model learns are structured with respect to structure in a given dataset. Here we walk through our measures for describing the representational system that emerges over the course of training, quantifying the amount of Information, Variation, Regularity, and Disentanglement.

Information (Entropy): We have a model f that maps a set of sentences X to representational space Y. For each sentence $S^k \in X$, the model takes as input a sequence of tokens — usually words — $t_a^k, t_b^k, t_c^k ... \in S^k$ and returns a sequence of vectors $v_a^k, v_b^k, v_c^k ... \in V^k$ where v_a^k is the vector corresponding to token a when it occurs in sentence k. While each sequence V^k is of variable length, the individual vectors are the same size. We can therefore create a list Y of all token representations from all sentences in the dataset

$$Y = [v_a^k : \forall v_a^k \in f(S^k) : \forall S^k \in X] \tag{4.2}$$

and calculate its dimension-wise entropy. The result gives us a measure of the average amount of information encoded in each dimension of the representation, $H_{dw}(Y)$. Given that the amount of information the model needs to encode is constant (the dataset doesn't change during training) this also tells us how compressed the model's representations are. As the dimension-wise entropy goes down, the model uses less of its available representational space. Information is minimised (i.e. compression is maximised) as all tokens are mapped to the same vector regardless of the token and sentence they correspond to, and information is maximised when token representations are spread out uniformly across representational space. To aid interpretation we normalise this measure, as well as Variation and Regularity, so that 1.0 indicates a uniform distribution and 0.0 is one-hot (this makes the quantity an efficiency). Our estimator is *invariant to linear transformations,* which means it ignores how numerically large a representational space is used. That is, this score is maximised if representations are spread out uniformly between the interval -2, 2 or -10,10 — what matters is representations' uniformity, not their magnitude.

Variation (Conditional Entropy) captures how much a property varies in representation space. Given a class of labels, like tokens, or parts of speech, it reflects whether the model learns a single global representation of each label invariant to context, or if each representation is completely unique to the sentence it occurs in. We quantify this in terms of the conditional entropy of representations, given a label, creating a list of all instances of that label $Y|label$, across all contexts where it occurs

$$Y|label = [v_a^k \ if \ a = label : \forall v_a^k \in Y] \tag{4.3a}$$

Labels for the tokens fed into a model are virtually always known, so we can easily estimate the conditional dimension-wise entropy of Y given a specific token $H_{dw}(Y|token)$. This is minimised when all instances of a token map to the same vector regardless of the sentence they occur in, and maximised when $H_{dw}(Y|token) = H_{dw}(Y)$ indicating instances of the same token are no more likely to be similar than two tokens chosen at random. The mean variation across the set S of all tokens gives us a general sense of how much the model encodes context in its internal representations.

$$variation(Y|Set) = \frac{1}{|S|}\sum_{label}^{S} H_{dw}(Y|label) \tag{4.3b}$$

We can also calculate variation with respect to any features we have a set of labels for. For example, if we know the part of speech for each of the input tokens $variation(Y|POS)$ could tell us if members of the same syntactic class share more information with each other than expected by chance. In the general case we just need a set of labels to condition on (e.g. part of speech, morphological case, tense etc.) when estimating $H_{dw}(Y|Set)$.

Regularity (Mutual Information) measures how structured a model's representations are with respect to a feature in the input — in particular, whether the mapping between a label and its representation is monotonic (one-to-one). The inverse of variation, Regularity quantifies how much knowing something about a token is going to tell us about its representation; quantifiable as the dimension-wise mutual information between a label and its representations.

$$regularity(Y, Set) = \frac{1}{|S|}\sum_{label}^{S} H_{dw}(Y) - H_{dw}(Y|label) \tag{4.4}$$

This is maximised when a label and its representations are monotonically aligned — knowing the label tells us everything there is to know about the representation. As with variation we can quantify regularity with respect to individual labels in a set and mean across them to get a general notion of how aligned representations are with e.g. tokens, POS tags, or the bigrams a token is part of.

Disentanglement (JS Divergence) measures how separable different labels within a set are from one another, e.g. whether separate tokens are represented in distinct regions of representational space, rather than overlapping. We measure this by

assessing the Jensen-Shannon divergence between $P(Y|label)$ and all other labels in the set $P(Y|Set\neg label)$; if tokens are distributed uniformly across a space their disentanglement will be 0, while if they are entirely separable it approaches 1.

$$dis(Y, Set) = D_{JS}(P(Y|label); P(Y|Set\neg label)) \tag{4.5}$$

As with previous measures we aggregate this to get an assessment of how disentangled the class of labels is. This measure is related to previous assessments of entanglement (Chen et al., 2018; Conklin & Smith, 2022) but is implemented quite differently, and requires no pair-wise comparison of different labels.

Measure Visualisations

I have included visualisations for each of these measures, that were not included in the original publication in figure 4.2. These help build an intuition for how different scores across the measures correspond to actual representations in vector space. Each facet contains 2 to 4 different distributions - each corresponding to a 'label' in the analysis - indicated by colour. For each distribution either a uniform, or multivariate normal distribution is selected at random, then randomly parameterised. 100 samples are drawn from each distribution, and the 4 measures introduced above are applied to these samples. To enable straight-forward visualisation each distribution is 2 dimensional. Each figure includes 3 different examples (each column) with 2-4 distributions (each row), note that the rows are additive with each plot including the plots above in its column. These visuals help us to link each measure to properties of clusters in real-valued space.

- Regularity: How predictable a region of space samples from a cluster will lie in.
- Variation: Cluster size - how much of the attested space is occupied by that cluster.
- Disentanglement: How separable clusters are from each other.

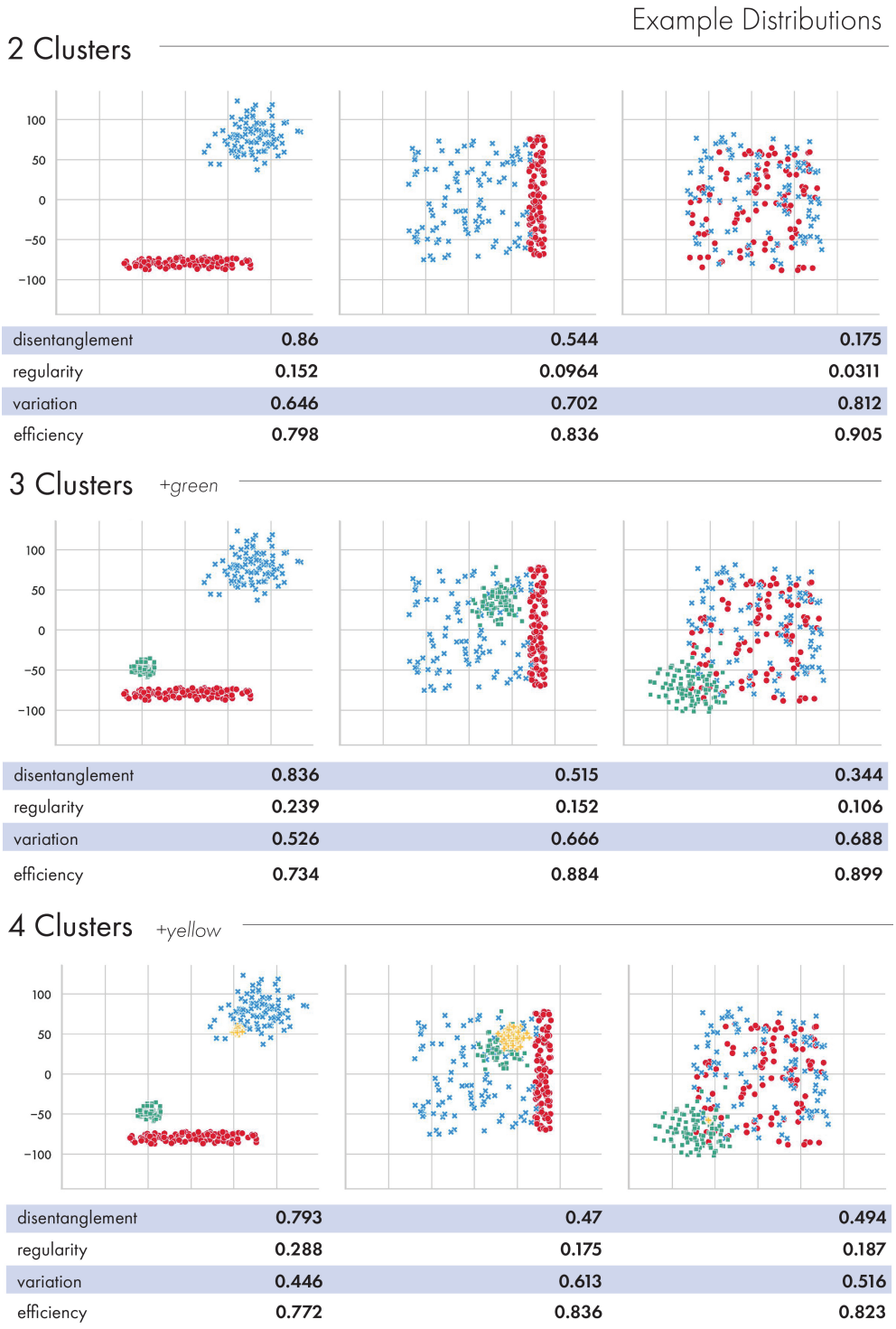


Figure 4.2: Examples of Measures defined here using the dimension-wise discretisation leveraged in this chapter. Each plot contains different clusters indicated by colour, with the scores across 4 measures below each facet. Disentanglement indicates cluster separability, regularity is the average mutual information with cluster label, variation is the average conditional entropy given a cluster label. Efficiency is the normalised entropy for the entire facet.

4.3 Results

We report results on two different datasets designed to assess compositional generalisation. Our measures allow us to characterise the trajectory of training, which we identify as having two distinct phases. We also compare models of different sizes to see how capacity changes representational space. Summary results are found in table 4.1. It is worth noting that some of our results may be particular to the hyperparameters used for training. We use hyperparameters recommended by the authors of the datasets we use, or that the transformer was introduced with Vaswani et al. (2017). We believe this means that our design choices are representative of common ones for training sequence-to-sequence transformers, our code itemises all parameters used. We compute measures with respect to labels for Tokens, Parts of Speech, and Bigrams in the input and for brevity report the values with the clearest effect on model performance. We also focus discussion on results from the MCD CFQ dataset, as it's the larger of the two (100,000 training examples) and is a more realistic task — mapping questions to SQL queries. We report results for the most challenging split of this dataset, known as MCD2. We include some discussion of SLOG, but an exhaustive listing of all results, across all datasets levels of analysis and model sizes can be found with the released code.

4.3.1 Two Distinct Phases of Training

We see 2 distinct phases of training, similar to Shwartz-Ziv and Tishby (2017), despite using rather different methods (studying classification with a feed-forward network rather than a linguistic task with a transformer). This suggests some generality to this characterisation of deep-learning, though our results point to different analyses of each phase (particularly the second, much longer one), likely due in large part to the difference in model and domain. While overall trajectories are consistent across conditions when different model sizes move between phases differs, for clarity here we refer to specific steps in the training timeline for the mid-size model on CFQ.

4.3.1.1 Phase 1 | In-Distribution Learning

Alignment & Disentanglement. In Phase 1 the model achieves high in-distribution accuracy, climbing to ceiling performance on the training data by step 1,000. This increase in accuracy is driven by an increase in token and POS regularity between

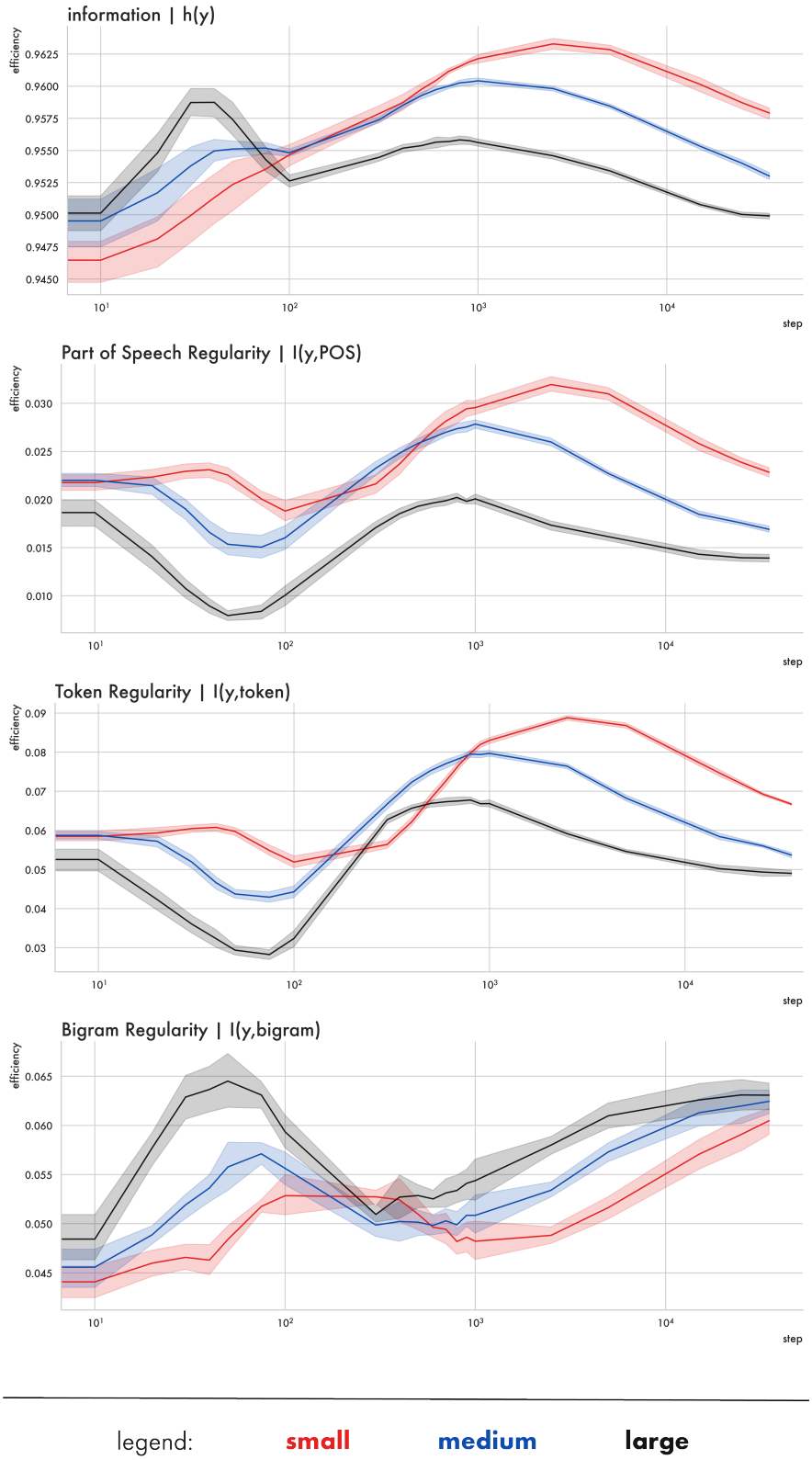


Figure 4.3: Each facet shows a different measure (along the y axis) against training steps (log scaled). Lines and shading give means and 95% CIs; line colours give results for 3 different model sizes. Values are calculated across the entire training set for 10 different random seeds. Efficiency (normalised entropy) is bounded such that 1.0 indicates a uniform distribution and 0.0 one-hot.

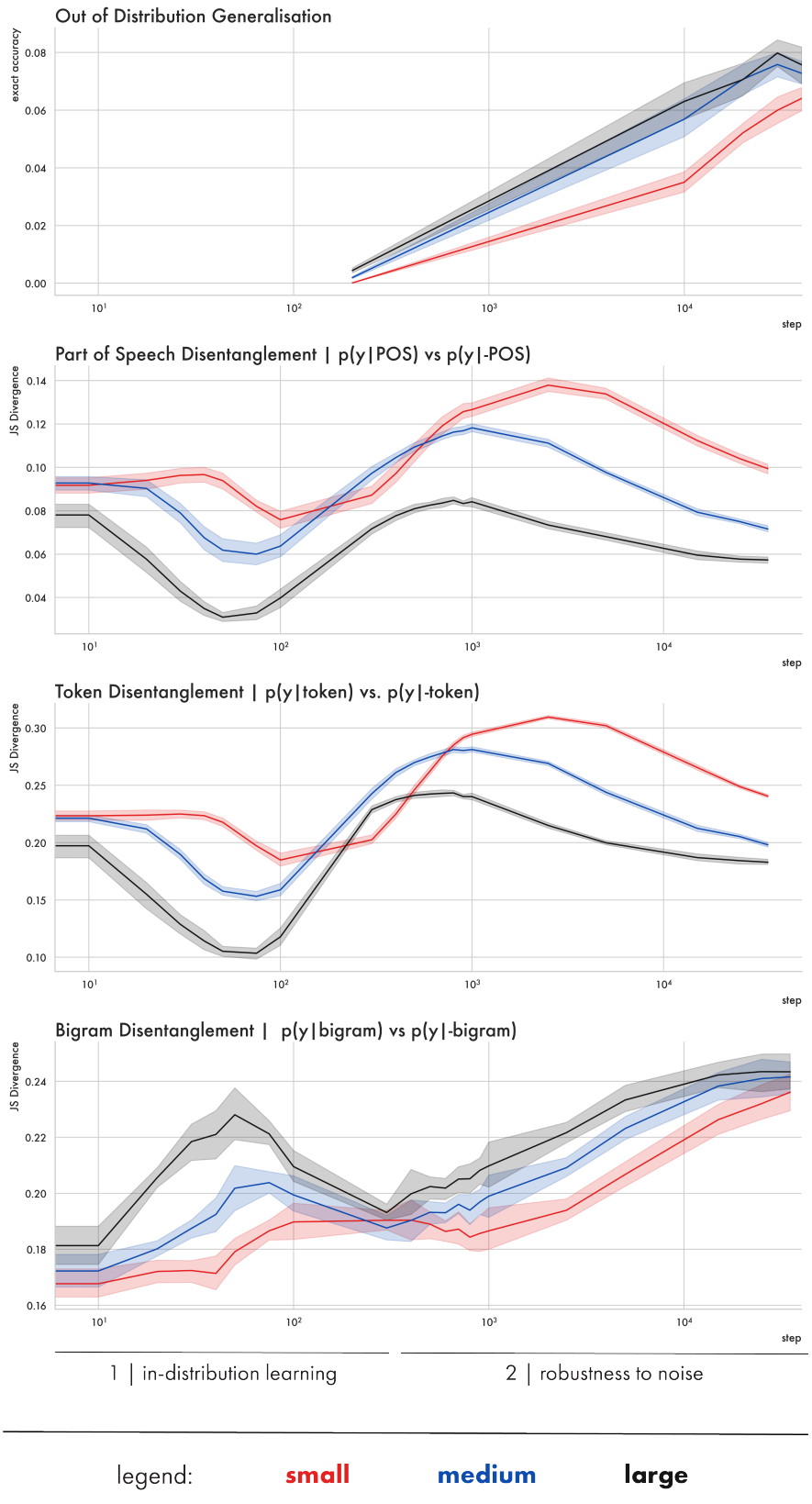


Figure 4.3: **(continued):** The top facet shows out of distribution generalisation performance. Note that this begins to increase as disentanglement increases across all levels of analysis. In particular though, the point at which generalisation performance increases closely aligns with when bigram regularity and disentanglement increase. Given the task requires models to interpret known words in new contexts, more disentangled contextual information (here bigram information acts as a proxy for contextual information) may allow the model to correctly decode the token in a broader range of sentences.

steps 100 and 1000 as representations become more monotonically aligned with the corresponding input token and its part of speech (fig 4.3, top left). This period also reflects an increase in POS and Token disentanglement indicating different tokens are represented in increasingly distinct regions of representational space. Conversely bigram regularity and disentanglement are reduced over the same interval as different token representations in the same bigram become more uniformly distributed over the support of $Y|token$.

Does training select for structure? During training the model tries to minimise a loss function, here the cross entropy between its predicted semantic representation for the input sentence and the correct one. During phase 1 we find a lower loss on the task (indicating better performance) correlates with our measures, suggesting the objective selects for certain structural properties in representation space. The timecourse of this is shown in figure 4.4, with correlations between structure measures and the loss for 100 different runs of the medium model on SLOG. From steps 100 to 200 all four of our token-level measures correlate negatively with task loss ($p < 0.001$), This dynamic shifts slightly from steps 200-600, where higher token disentanglement ($p < 0.001$) and regularity (indicative of a more monotonic alignment) ($p < 0.001$) continue to correlate with lower task loss but now with less variation ($p < 0.001$, steps 280-600).

Past this point all measures cease correlating with the task loss, which is also the point where empirical error begins to saturate — as the model approaches ceiling performance on the training set, the loss asymptotically approaches its floor. Figure 4.4 also shows the correlation between loss and our measures conditioned on part of speech tags. Similarly, greater regularity and disentanglement with respect to part of speech labels and less variation correlate strongly with a better task loss from step 100 until 600 ($p < 0.001$). The peak spearman coefficient for disentanglement reaches -0.71 indicating the objective optimizes more strongly for disentanglement of parts of speech than tokens (which peaks at -0.58 and fades from significance faster).

4.3.1.2 Phase 2 | Robustness to Noise

Contextualisation & Compression This is the dominant dynamic of training, taking place from step 1000 onwards. During this period the representational space slowly compresses, with dimension-wise entropy decreasing. This is coupled

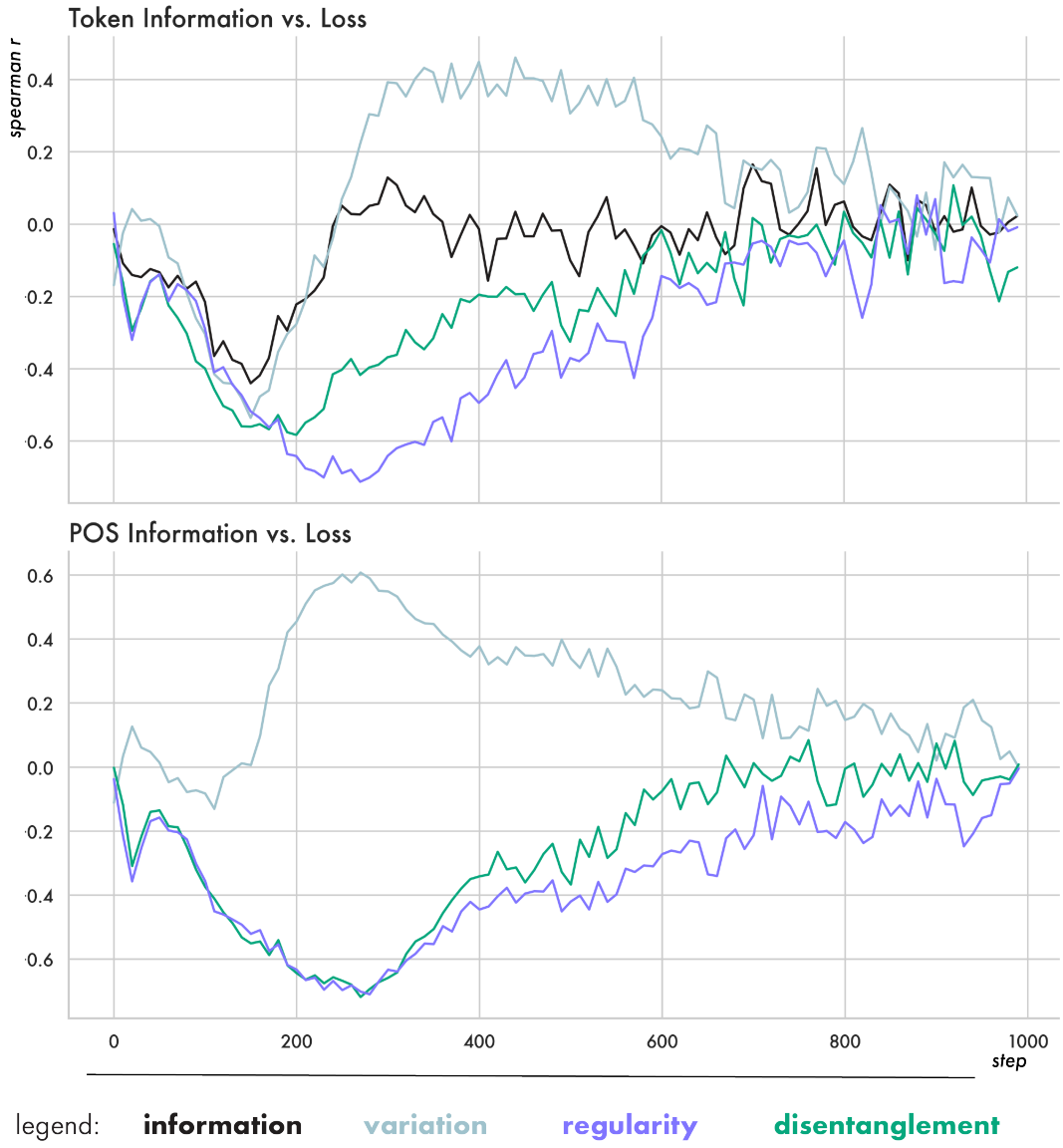


Figure 4.4: Spearman correlation coefficients between the loss minimised during training and our measures. Negative coefficients suggest the objective increases a quantity. Results for 100 runs of the medium model on SLOG. When exactly significance fades is noted in body text. **Top:** measures conditioned on token labels. **Bottom:** measures conditioned on part of speech tags for the tokens.

SLOG	POS	Token	Bigram
Information	0.96 ± 0.0003	0.96 ± 0.0003	0.96 ± 0.0003
Variation	0.92 ± 0.0008	0.85 ± 0.0006	0.75 ± 0.0004
Regularity	0.04 ± 0.0006	0.11 ± 0.0008	0.10 ± 0.0008
Disentanglement	0.15 ± 0.0022	0.35 ± 0.0017	0.37 ± 0.0018
			Accuracy: 27.84 ± 0.7613
CFQ			
Information	0.95 ± 0.0002	0.95 ± 0.0002	0.95 ± 0.0002
Variation	0.94 ± 0.0003	0.90 ± 0.0005	0.84 ± 0.0017
Regularity	0.02 ± 0.0003	0.05 ± 0.0005	0.06 ± 0.0012
Disentanglement	0.07 ± 0.0012	0.20 ± 0.0015	0.24 ± 0.0042
			Accuracy: 7.59 ± 0.4130

Table 4.1: Summary results for measures at the POS, Token, and Bigram level, across 10 runs of the medium model on both datasets, with 95%CIs. Measures are computed at the last step of training across the entire training set. Models' accuracy % reported on the held out generalisation set.

with an increase in bigram regularity as clusters for different contextualizations become more distinct in representational space, respecting contextual structure forces the overall token regularity down as representations become less aligned with token information and more aligned with bigram information. These shifts happen slowly taking thousands of training steps.

Shwartz-Ziv and Tishby (2017) note that later in training, after the loss has reached its floor, the update steps the model takes begin to behave like 'gaussian noise with very small means.' This aligns with what we see here, as measures of structure cease to consistently correlate with the task objective by phase 2. This suggests that a major dynamic of the latter period of training is representations becoming increasingly robust to noise. The model's mapping from sentences to representations needs to continue to encode the input, but do so robustly enough that the mapping won't be undermined by constant noisy updates, otherwise the task loss will begin to increase. Unlike previous work we note mutual information increases between inputs and representations later in training, just at higher level of granularity — here, bigrams.

It's also worth noting that while the model achieves ceiling performance on the training and validation data during phase 1, it only begins to succeed on the more challenging out-of-distribution generalisation task 10,000 steps later (see figure 4.3 top right). This means robust generalisation ability begins to appear only after a sustained period of representations becoming more robust to noise. This is related to the double descent phenomenon (Nakkiran et al., 2021), where models begin to exhibit strong generalisation performance long after the initial learning of in-distribution data. Voita et al. (2021) also note that in machine translation a transformer starts by learning individual token probabilities before acquiring more complex sentential structure. Our results give a mechanistic account of how this may happen, with token alignment increasing first, then a much longer phase where representations become more contextualised. Though our task is simpler than large-scale translation, in future we aim to apply this analysis to that context.

What kinds of representations generalise best? We also look within conditions to see if representational structure correlates with generalisation across different runs of the same model. We take the middle-sized model on CFQ and correlate across 10 runs at the final step of training. This analysis shows that runs with higher bigram disentanglement ($r = 0.65$, $p = 0.04$), and higher bigram regularity generalise better ($r = 0.61$, $p = 0.06$). The generalisation set of CFQ contains tokens seen during training as part of novel contexts. In order to do well our model needs to correctly encode tokens it has seen before, in contexts it hasn't. Higher bigram regularity and disentanglement indicates different contextualisations for the same token are more tightly clustered in space and that those clusters are more pure (being separable from other contextualisations of that token). More predictably and separably encoding different bigrams may help novel contextualisations of a token to be decoded correctly.

4.3.2 Model Size Clearly Affects Representational Space

While the overall phases of training are remarkably consistent across datasets and model sizes, there is a clear influence of model size on representational structure. Figure 4.3 shows trajectories for our three different model sizes over the course of training. Smaller models are less compressed, and have greater regularity and disentanglement with respect to tokens and parts of speech. They also perform

worse on both tasks than their larger counterparts. Larger models are more entangled at the POS and token level, but have more disentangled bigrams — indicating larger models learn more pure clusters for different contextualisations of the same token.

Why Models Compress & Larger Models Compress More It's common to think of connectionist models as cognitive models, and expect them to be governed by similar constraints (Futrell et al., 2018). Humans may generalise robustly because constraints on our cognitive capacity force us to learn generalisable solutions rather than memorizing every possible outcome (Griffiths, 2020a; Hahn et al., 2022). The fact that larger models (with greater capacity) compress more per-dimension, would seem at odds with this framing. While we agree that drawing cognitive parallels can be useful, on a representational level looking at models as a language can help us to reason about the effects of scale and the phases of training.

Specifically, our interpretation is that larger models are able to exploit their higher-dimensional internal representations to develop representations more robust to noise. An obvious analogy in communication is mapping an input to a discrete signal, where the signal space is defined by an alphabet of characters and a maximal signal length. If the signal length is low, a larger alphabet is needed to encode the input unambiguously. In contrast, if longer signals are allowed a smaller alphabet is required, the limiting case being a binary alphabet (like morse code) where sentences are encoded in comparatively long signals. Signals composed from a smaller alphabet are more resilient to noise [2] for instance, when an operator interprets morse code, at each point in the sequence they only need to differentiate between two possibilities, dot or dash, which is easier than distinguishing between e.g. 26 different outcomes, particularly on signals transmitted over copper wire. We have shown how, during the second phase of training, transformers compress their representations in response to noisy update steps. This is directly analogous to models using a progressively smaller vocabulary for each dimension of hidden space. Larger models have more dimensions, which in our analysis is akin to having a longer maximum signal length, enabling them to learn a mapping more robust to noise, like morse code, converging to a smaller alphabet but longer signal.

[2] This is implicit in Shannon's definition of entropy, as the maximum uncertainty of a binary distribution is lower than one with more outcomes ($log(2) < log(3)$)

4.4 Conclusion

We have introduced a linguistically-motivated approach to interpreting transformer models. By looking for system-level structure in the model's representations, we characterise two-distinct phases of training, and show how representational structure develops during those phases and how this explains model's ability to generalise. This is enabled by an efficient approach to estimating the entropy of transformers' latent space, that allows for non-parametric analysis of representational structure. Our findings help shed light on what the learning process looks like in deep-learning models, and makes a case that intuitions from linguistics and cognitive science about what makes for a 'good' representation may meaningfully transfer here.

Chapter 5

Information Structure in Large Language Models with Soft Entropy

A Continuous Approach to Entropy Estimation.

> Up the coast a few miles north [redacted] there are a lot of rock pools. You can visit them when the tide is out. Each pool is separate and different [redacted] individual entities [redacted] throughout the day of the ebb tide, they know no other. But [redacted] the waters of the ocean come flooding [redacted] Can they tell us, in any manner, about their journey? Is there, indeed, anything for them to tell– except that the waters of the ocean are not really other than the waters of the pool?
>
> *- Christopher Isherwood*

Chapter 4 studied how information structure emerges in a model trained on a single task. This chapter looks instead at large language models (LLMs), which are pre-trained on huge volumes of data scraped from the internet or from digitised collections of books (Raffel et al., 2019). Starting with BERT (Devlin et al., 2019), LLMs have put up state of the art performance across a broad range of natural language tasks (e.g. Brown, 2020; Devlin et al., 2019), often generalising significantly more robustly than smaller models trained on a single task (Furrer

et al., 2021). This chapter analyses large language models from an information structure perspective.

While scaling up models has seemed to lead to greater performance, scale also makes analysis more challenging. The analysis from the previous chapter relies on binning representations in order to estimate entropy, but with the larger models studied in this chapter having a hidden dimension of 5120 this approach becomes intractable. To address this, this chapter also introduces a novel approach to entropy estimation — **soft entropy**. Soft entropy splits the difference between discrete and differential entropy, estimating a quantity that behaves like discrete entropy but which is differentiable and works to respect properties of continuous space. This approach primarily relies on matrix multiplication, making it highly parallelisable which allows us to apply our analysis to models of arbitrary size.

Given large language model's significant improvements in performance compared with models trained on a single task, we study the information structures they converge to and the training dynamics that lead them there. Broadly the results in this chapter paint a similar picture to the training dynamics of the single task model in chapter 4, namely early learning of lexical information followed by a slow contextualisation phase - suggesting some generality to this characterisation of deep-learning in the language domain.

The remainder of this chapter is a paper that is currently under review. Authors are myself and Kenny Smith - Kenny and I conceived the experiments together which I then implemented and wrote up. The paper is presented here minimally changed from the submitted version. Changes are largely related to formatting to make the content more readable outside of the original conference paper template, and to make notation consistent across different chapters.

5.1 Information Structure in Large Language Models

Despite the remarkable performance of large language models (Brown, 2020; Dubey et al., 2024), and their widespread use we still lack unified notation for thinking about and describing their representational spaces. We lack methods

to reliably describe how their representations are structured, how that structure emerges over training, and what kinds of structures are desirable. This should be of concern to us for practical reasons - it makes it difficult to make design decisions when we don't have a clear picture of how they effect representational space - but also for broader social reasons. Most people in the US and UK come into contact with an NLP system multiple times a day without realising (Kennedy et al., 2023). Given their increasing ubiquity we should be able to account for the information they have learned and how that information is structured.

Our lack of tools for understanding representations in networks is in part because their representations are continuous, and we as humans tend not to have strong intuitions about high-dimensional vector spaces. Existing work interpreting large language models describes phases of training in terms of model behaviour (e.g. Blevins et al., 2018; Dziri et al., 2024; Marvin & Linzen, 2018), for example analysing when they begin to generalise robustly - or *grok* (Merrill et al., 2023; Power et al., 2022). Alternately work uses parametric methods like probing, leveraging a separate model to describe the first (Hupkes et al., 2019; Pimentel et al., 2020; Voita & Titov, 2020). We focus instead on giving a representational account of what training looks like, using information theoretic measures of representational space to quantify how structured representation spaces are in large language models, and what kinds of structure matter for generalisation. Ideally we need a way of thinking about deep-learning models in the general case that allows us to:

1. Describe structure in representation space, and what structures drive generalisation

2. Clearly relate these to relevant work in linguistics and the cognitive sciences

3. Quantify structure with methods that are efficient enough to apply the same analyses to models of any size, throughout training

4. Meaningfully compare models of different sizes, trained with different objectives

In an effort to do this, we look at deep-learning models as member of a more general class: mappings. Models map between their inputs and representational space, and are comprised of a sequence of linear and non-linear mappings. Here

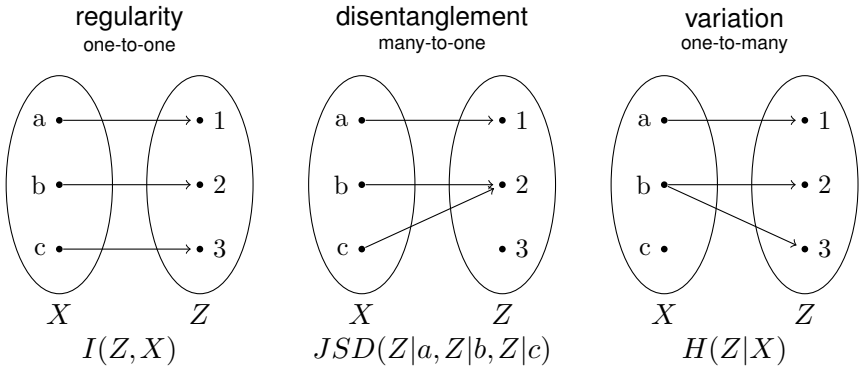


Figure 5.1: Three basic kinds of mapping structure we consider here, labelled with their linguistic analog, and the information theoretic quantity we introduce to measure them in section 5.4. Note that we show part of the mapping ($a \to 1$) as regular in all cases because the mappings we consider exhibit a combination of all 3 structures. As such we assess the *degree* of each structure, not whether or not it exists. Variation (one-to-many) is possible despite the fact that the networks mappings we examine are deterministic, because our X contains instances of the same token in different sentences, meaning $b \to 2$ and $b \to 3$ reflects b in different contexts.

we quantify structure in the mappings learned by large language models while drawing parallels to a reference mapping about which we have strong intuitions for what structure looks like - unlike high-dimensional vector spaces - and which is related to the domain in which our models are trained: natural language.

At its core language is a mapping - relating concepts, and complex propositions, to words, constructions, and phrases which refer to them (de Saussure, 1916). While many natural communication systems fit this bill, language is unique amongst them (Hockett, 1960). It is learned from a finite sample, generalises readily to novel concepts and contexts, with *system level* structures that provide us a system simple enough to be learned by children, but expressive enough to describe the universe. This parallels our desiderata for mappings in deep-learning models which need to be learned from finite data, able to generalise, and expressive enough to describe the world from which their training data is drawn. We look at whether *system level* structures emerge in representation spaces learned by large language models; first introducing basic kinds of structure in a mapping, relating them to their analogs in linguistics, before quantifying each of them information-theoretically.

We build on the framework for interpretability introduced in Conklin and

Smith (2024), redefining some of their measures, and extending it to large language models. To do this we also introduce a novel method for highly-parallelisable entropy estimation in vector space - **soft entropy**. This approach is similar to discretisation based methods used to analyse deep-learning (Goldfeld et al., 2018; Shwartz-Ziv & Tishby, 2017), but fully differentiable, less affected by hyper-parameter settings, and dramatically more memory and compute efficient. Additionally the estimator can easily be applied at different levels of abstraction like model, layer, and subspace - this broken-down estimate enables direct comparison between different model sizes. We use soft entropy to quantify structure in language models ranging from 14 million to 12 billion parameters, looking at when system-level structure emerges during training, how scaling affects representation structure, and what kinds of structure drive generalisation. Our analysis is able to predict downstream performance on GLUE benchmarks based only on a models' representations at the end of pre-training (before 2 million steps of fine-tuning). To summarise our core contributions, this paper:

- Frames structure in large language models in terms of related notions of structure from linguistics and information theory
- Introduces a novel method for entropy estimation of continuous spaces, that is fast, efficient and differentiable
- Shows how scaling a model's hidden dimension, or number of layers, affect representational structure
- Correlates representation structure at the end of pre-training with performance downstream after fine-tuning

5.2 Related Work

Our work is related to a long history of research in NLP which tries to identify correspondences between linguistic structures in training data and representations or behaviours (Belinkov et al., 2017; Blevins et al., 2018; Dziri et al., 2024; Marvin & Linzen, 2018; Shi et al., 2016). It is particularly closely related to probing (Hupkes et al., 2018; Pimentel et al., 2020) which trains a classifier to predict labels from a larger model's representations. MDL probing (Voita & Titov, 2020) also includes a notion of regularity in terms of the complexity of the probe required

to recover the labels. Given that we quantify structure in the mapping between labels and representations directly, our work represents a non-parametric approach to probing. The analysis here is also related to work in language emergence which looks at the languages that emerge between models in a multi-agent setting. A variety of quantifications of linguistic structure have been proposed for that domain that leverage similar intuitions to the ones used here (Brighton et al., 2005; Chaabouni et al., 2020; Conklin & Smith, 2022; Lazaridou et al., 2018; Resnick et al., 2020)

There is also existing work that tries to characterise training dynamics information theoretically (Goldfeld et al., 2018; Saxe et al., 2019; Shwartz-Ziv & Tishby, 2017; Tishby & Zaslavsky, 2015), however these are largely theoretical works and/or applied to feed-forward networks on tasks like digit classification. Conklin and Smith, 2024 applies information theoretic methods to transformers trained on a single task - but uses dimension-wise discretisation, which is difficult to scale. Our approach to estimating entropy is similar to the limiting density of discrete points (Jaynes, 1957) and is related to kernel density estimation (Parzen, 1962) in the way it relates discrete points to a continuous function.

5.3 Identifying Structure in Mappings

We consider 3 basic structures in a mapping between two spaces: *one-to-one*, *many-to-one*, and *one-to-many* (see Figure 5.1). These are related to linguistic concepts of regularity, disentanglement, and variation respectively. In a model we quantify these properties between labels for a model's input and the corresponding representations. Any labels for an input sentence can be used, experiments here use ones that come for free with any text data: token, bigram, and trigram. This enables analysis of lexical and contextual information in the model and shows the generality of the approach. Data labeled with parts of speech could show how syntactic information is represented — given any set of labels for the input our analysis quantifies structure in representation space with respect to them.

To formalise this in terms of transformer language models, consider mappings at the token level. Given a model f that maps a set of sentences X to representational space Y. For each sentence $x^k \in X$, the model takes as input a sequence of tokens $t^k_a, t^k_b, t^k_c ... \in x^k$ and returns a sequence of vectors $y^k_a, y^k_b, y^k_c ... \in Y^k$ where y^k_a is the vector corresponding to token a when it occurs in sentence k. While each sequence

Y^k is of variable length, the individual vectors are the same size. We can create a list Y of all token representations from all sentences in the dataset, or a list of all tokens corresponding to a given label $Y|label$.

$$Y = [y_a^k : \forall y_a^k \in f(x^k) : \forall x^k \in X] \tag{5.1a}$$

$$Y|\text{label} = [y_a^k \ if \ a = \text{label} : \forall y_a^k \in Y] \tag{5.1b}$$

We can apply the same approach to look at bigram or trigram information, where we label the representation y_a^k with either bigram (a, b) or trigram (a, b, c). The next section explains how we estimate entropy in vector space; first we walk through the kinds of structure we measure. The estimation procedure gives us a categorical distribution which describes vector space $P(Y)$ used below.

A Note on the Relationship with the Preceding Chapter

The formalisations presented here build on those introduced in the previous chapter, and for clarity we can look at where they overlap or are distinct. Measures of regularity and variation are the same in both. Regularity is the label-level mutual information aggregated across the entire set; Variation is label-level conditional entropy aggregated. In the previous chapter disentanglement used the Jensen-Shannon divergence between a label's distribution, and a distribution of all other labels in a set. This requires a computation of the divergence at the label level that then gets aggregated across all labels in a set. In this chapter it is instantiated as a multi-variate Jensen-Shannon divergence - the divergence between each individual label distribution and their mixture. This assesses a similar quantity as the previous disentanglement measure, but does so in a single Jensen-Shannon divergence rather than needing to aggregate across individual labels. The information proportion measure is new for this chapter, and represents a kind of normalised mutual information that reflects how regularity with respect to different sets of labels fit together in the model.

Variation describes how much representations for a label vary in representation space. In the token case this reflects whether a model learns a single context independent representation of the token or a different representation for every sentence it occurs in. We can quantify this in terms of the conditional entropy of space given a label. The resulting quantity is related to intrinsic dimensionality (Levina & Bickel, 2004), reflecting how much of representational space is used to represent a given feature in the input, but faster to compute given it requires no pairwise comparisons. In addition to the formalisation below we bound this and the regularity measure to lie between 0 and 1 to aid interpretation[1].

$$\textsf{variation}(Y, \text{set}) = \frac{1}{|\text{set}|} \sum_{\text{label}}^{\text{set}} H(Y|\text{label}) \tag{5.2}$$

Regularity reflects the amount of variation in representation space we can explain by knowing a label. It is bounded mutual information, and reflects the difference between overall variation in the space H(Y) and the variance in representations for a given label H(Y|label). It reflects how monotonically aligned representation space is with that label. In language regularity is often measured similarly (Ferdinand et al., 2019; Smith & Wonnacott, 2010) and is used to quantify how syntactically structured a system is.

$$\textsf{regularity}(Y, \text{set}) = \frac{1}{|\text{set}|} \sum_{\text{label}}^{\text{set}} H(Y) - H(Y|\text{label}) \tag{5.3}$$

Disentanglement measures whether clusters corresponding to labels within a set are separable — e.g. whether different tokens or bigrams are represented in different parts of space. We estimate this with a multi-variate Jensen-Shannon divergence. This requires a mixture distribution M computed by taking a mean of individual label distributions weighted by the probability of the label $M \propto \sum_{\text{label}}^{\text{set}} P(\text{label})P(Y|\text{label})$. The divergence then looks to see if the entropy of the mixture $\mathcal{H}(M)$ can be explained in terms of the individual label distributions. The result is the mutual information between the mixture M and the weights used to create it $P(\text{label})$ and so is bounded by the entropy of the weight distribution $\mathcal{H}(\text{label})$. We use this to normalise the measure so that as values approach 1 labels are maximally separable in space, and as it approaches 0 all labels in a set

[1] Bounded by dividing by the entropy of a uniform distribution, converting entropy to efficiency.

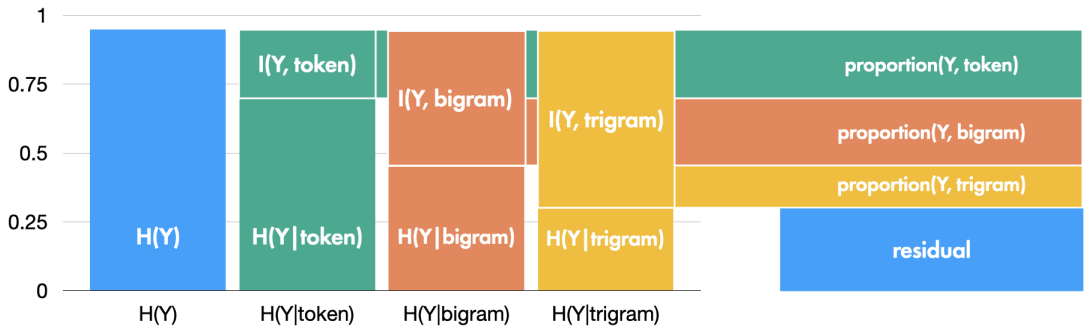


Figure 5.2: An exemplar showing how different information theoretic quantities relate to each other. x axis shows different sets of labels, y axis shows efficiency (entropy bounded to between 0 and 1). The lower portion of each bar is conditional entropy with respect to each label set. The top is mutual information with a given set $H(Y) - H(Y|\text{set})$. Information proportion is the difference between the mutual information of a set and its super set — if it has one— with residual information being information left in $H(Y)$ that cannot be explained in terms of any set of labels. Here the super set for bigrams are tokens (given we analyse token representations occurring within a specific bigram), and the difference between their mutual information reflects the amount of information in representation space attributable to bigrams alone. Tokens have no super set, so their information proportion is equivalent to their mutual information (regularity) normalised by the entropy of the space $H(Y)$.

overlap. This is related to previous measures of entanglement (Chen et al., 2018; Conklin & Smith, 2024) but is faster to compute, and allows labels to contribute proportionally to the estimate based on their probability.

$$\text{disentanglement}(Y, \text{set}) = H(M) - \sum_{\text{label}}^{\text{set}} P(\text{label}) H(Y|\text{label}) \tag{5.4}$$

Proportion reflects the proportion of information in the model a set of labels can account for. Any information that can't be explained in terms of a label is considered 'residual' - and is computed as a residual entropy (see Resnick et al., 2020, for some discussion). Recall that regularity describes how much variation in representations we can explain by knowing a label, or how much information in the model can be explained in terms of that label. To compute an information proportion we compute how much of the model's total information (H) is accounted for by regularity with respect to a specific label set $\text{regularity}(Y, \text{set})$. In experiments here though, the labels used nest inside each other: representations corresponding

to a trigram label, are also part of a bigram label, which are also part of a token label. Which means regularity with respect to trigram information, includes regularity with respect to both bigram and token information. We separate this out by subtracting the regularity of the superseding label set $\textsf{regularity}(Y, \text{super set})$ if there is one and normalising by the entropy of the entire space $H(Y)$. If there is not a superseding set (as in the case of token level labels) then we simply normalise regularity by the entropy of the space. Relationships between variation (conditional entropy), regularity (mutual information) and the information proportion are shown in figure 5.2.

$$\textsf{proportion}(Y, \text{set}) = \frac{\textsf{regularity}(Y, \text{set}) - \textsf{regularity}(Y, \text{super set})}{H(Y)} \tag{5.5}$$

The residual, or remaining information in the model that cannot be explained in terms of a label set, is estimated by taking the label set with the highest regularity, and subtracting it from the entropy of the space. This leaves over the entropy that cannot be explained in terms of even the most regular set of labels (or any of the superseding ones). Normalising this by the entropy of the space gives us a proportion.

$$\textsf{residual}(Y, \text{set}) = \frac{\textsf{regularity}(Y, \text{smallest set})}{H(Y)} \tag{5.6}$$

5.4 Soft Entropy Estimation

There are few approaches to entropy estimation that are sufficiently fast and memory efficient to be applied to large language models. This is frustrating given information theoretic tools are well suited to quantifying complex structures in distributed systems. With soft entropy we introduce an approach that prioritises efficiency, while performing comparably to existing methods. It's worth noting that we focus on estimating the *discrete* entropy rather than differential entropy. We draw inspiration from Jaynes (1957), who notes differential entropy is not the true continuous analog of discrete entropy and proposes the limiting density of discrete points as an alternative. This takes entropy to be the divergence between a distribution and an invariant measure (usually a uniform distribution over the same support); it reflects how 'non-uniform' a distribution is. Our method follows this intuition, sampling points uniformly across space, and comparing them with samples from the model.

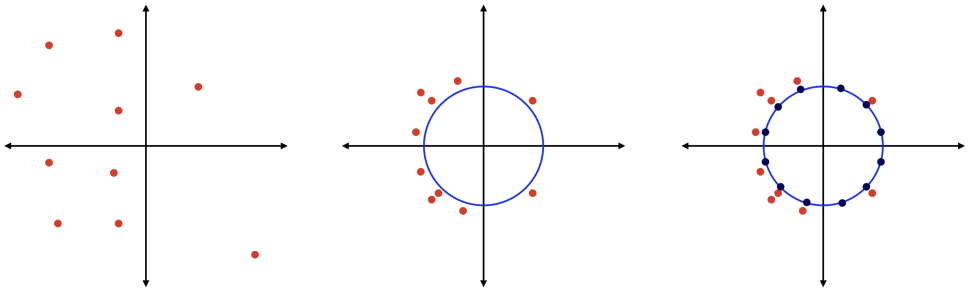
We define a mapping between real-valued space and information space, creating a categorical distribution that describes a model's representation space. Our estimator returns the entropy of the descriptor distribution, a quantity we call *soft entropy* - distinct from the differential entropy of the space. This process is akin to 'plug-in' estimation (see Beirlant et al., 1997, for review), where you first fit a distribution then estimate its entropy - except here the distribution we 'fit' is categorical. Existing approaches to estimating entropy of vector space often rely on discretisation with clustering (Sajjadi et al., 2018), or binning (Shwartz-Ziv & Tishby, 2017), the approach described here can be seen as a differentiable relaxation of these methods.

5.4.1 Formalisation

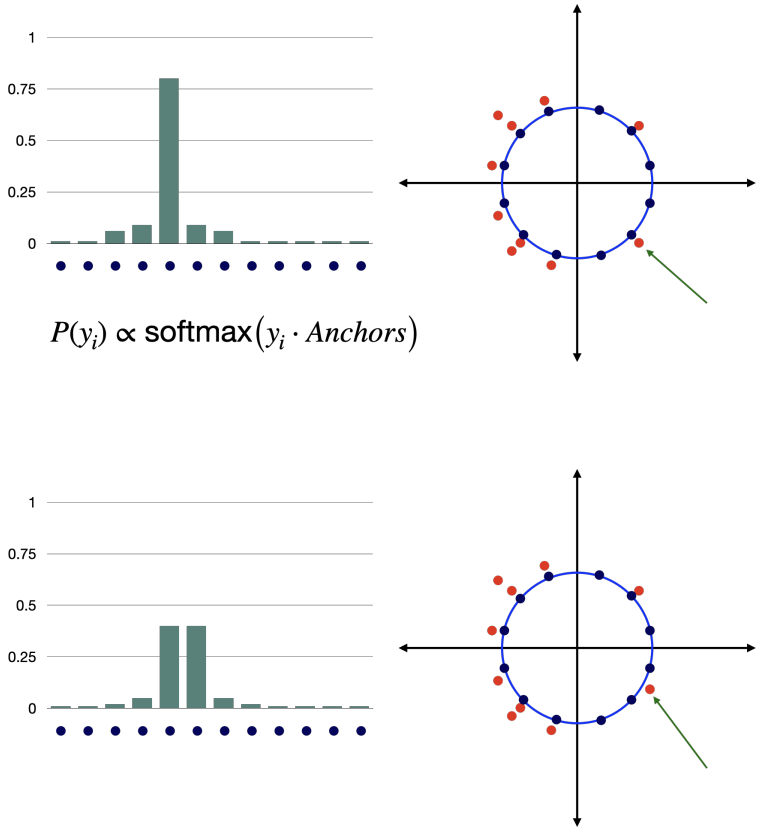
Given a set of representations Y with dimensions batch size bs by hidden size h we take the euclidean norm, so they lie on the unit sphere (note the entire estimation process is depicted visually in figure 5.3). We then sample points uniformly from the surface of the unit sphere, by drawing n samples from a standard normal and taking their euclidean norm. The resulting points S have dimensions $h \times n$ where n is a hyperparameter controlling the number of points. To assess how close each representation is to each point we take the dot product between Y and S. The result is a cosine similarity, which we pass through a softmax to get a distribution over points for each representation with dimensions $bs \times n$. By summing over the batch dimension and re-normalising we get a single categorical distribution that describes the space $P(Y)$ with dimensionality $1 \times n$. To get a binning based estimate we could treat each point as the center of a bin, and assign representations to the point they're closest to, rather than normalising distances.

$$\underset{1\times n}{P(Y)} \propto \sum \text{softmax}\left(\underset{bs\times h}{\frac{Y}{|Y|}} \cdot \underset{h\times n}{\frac{S}{|S|}}\right) \tag{5.7}$$

Because this gives us a categorical distribution, estimation of information-theoretic quantities is straightforward. Soft Entropy of the space follows the equation for shannon entropy: $H(Y) = -P(Y)\log P(Y)$. We can also quantify entropy in subspaces, as opposed to the entire space by applying the estimator in a multi-headed arrangement. We reshape the representations from $bs \times hidden$ to $bs \times head \times \frac{hidden}{heads}$ and the points to $\frac{hidden}{heads} \times heads \times bins$. This allows us to estimate entropy per-head and mean across them.



(a) Points in a 2D Coordinate Space shown in red (left) are normalised to lie on the unit sphere (centre) - here where multiple points are in the same location on the surface they appear as stacked. Anchor points, shown in blue, are then uniformly sampled from the surface of the sphere (right)



(b) Anchor points are used as events in a categorical distribution. To get a distribution over anchors for each red point we take its distance from each anchor - in terms of a dot product - and pass them through a softmax. Shown at top is the distribution over anchors for the point marked with the green arrow. Below is the same distribution when the point is equidistant between anchors.

Figure 5.3: A visual depiction of the soft entropy estimation process (continued on next page).

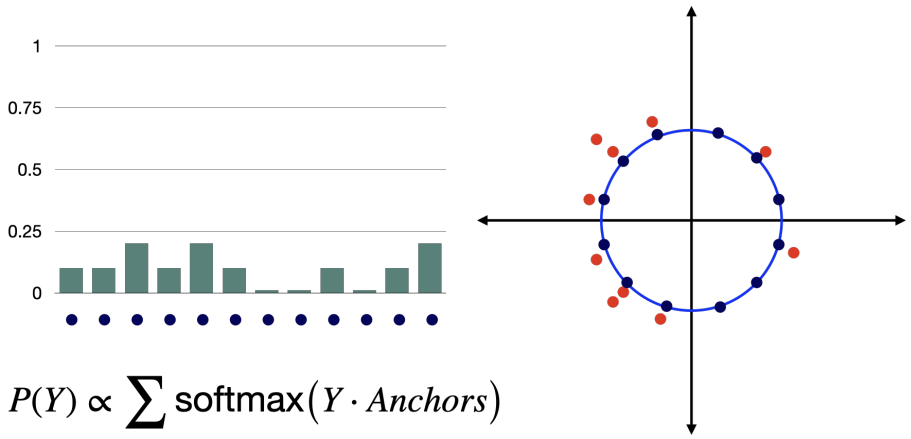


Figure 5.3: (continued) (c) By taking a summation over the distributions for each individual red point and normalising we get a distribution that describes the entire space. Note how the distribution over events in the bar plot at left, is reflective of where red dots are distributed over the surface of the sphere on the right.

H(Y) reflects how uniformly distributed representations are across angles with respect to the origin. It is maximised when representations are uniformly distributed across all 360 degrees, and approaches 0 as representations cluster across an increasingly small subset of angles. This quantity is related to anisiotropy, where representations lie in a narrow cone relative to the origin, but is dramatically faster to compute than taking pairwise cosine similarities between all representations. We draw a parallel between this measure and clustering based estimates of entropy, where representations are first clustered, then discretised (Sajjadi et al., 2018). Here when we project points to the unit sphere we make representations with high cosine similarity, close to each other. To get a clustering estimate we could replace the events in the categorical distribution $P(Y)$ with clusters on the unit sphere rather than uniformly sampled points. In practice sampling points is substantially faster than performing clustering.

5.4.2 Parameters & Computational Efficiency

In the same way that discretisation methods are sensitive to the number of bins used, soft entropy is sensitive to number of 'points' although less so than the discrete case: if two representations are close to each other they can't be split into

separate 'bins,' given we get a distribution over points for each representation rather than assigning it to a single point. This means that increasing the number of points in S doesn't necessarily have a detrimental effect on mutual information and divergences but can still inflate the estimate. In the experiments presented here we use 50 points unless otherwise noted. Additionally a softmax is not invariant to linear transformations and the distances from the dot product are bounded between -1 and 1, this can mean the default estimate is relatively high. After testing on reference distributions we opt to rescale the distances to lie between -100 and 100. This scaling factor is a parameter, like the bandwith parameter in kernel density estimation (Parzen, 1962), controlling the spread with respect to each point.

Our methods map representational space to a categorical distribution using a single dot product, softmax, and summation. These operations are differentiable, memory efficient, fast, and parallelisable. This process is non-parametric, requires no clustering, and is substantially more memory efficient than binning based approaches to entropy estimation which usually requires a step where representations are $bs \times seq \times hidden \times bins$ - using 100 bins on a model with 4096 dimensional spaces proves problematic.

Measure Visualisations

Figure 5.4 visualises each of the information structure measures using soft entropy. These show the measures here applied to the same example visuals as in the previous chapter (figure 4.2) but leveraging soft entropy rather than dimension-wise discretisation. Each visualisation contains 2 to 4 different distributions indicated by colour - each corresponding to a 'label' in the analysis. For each distribution either a uniform, or multivariate normal distribution is selected at random, then randomly parameterised. 100 samples are drawn from each distribution, and the 4 measures introduced above are applied to these samples. To enable straight-forward visualisation each distribution is 2 dimensional. These visuals help us to link each measure to properties of clusters in real-valued space. Broadly the values here are similar to the dimension-wise approach, but can better account for clusters that overlap on each dimension but are separable in 2D space.

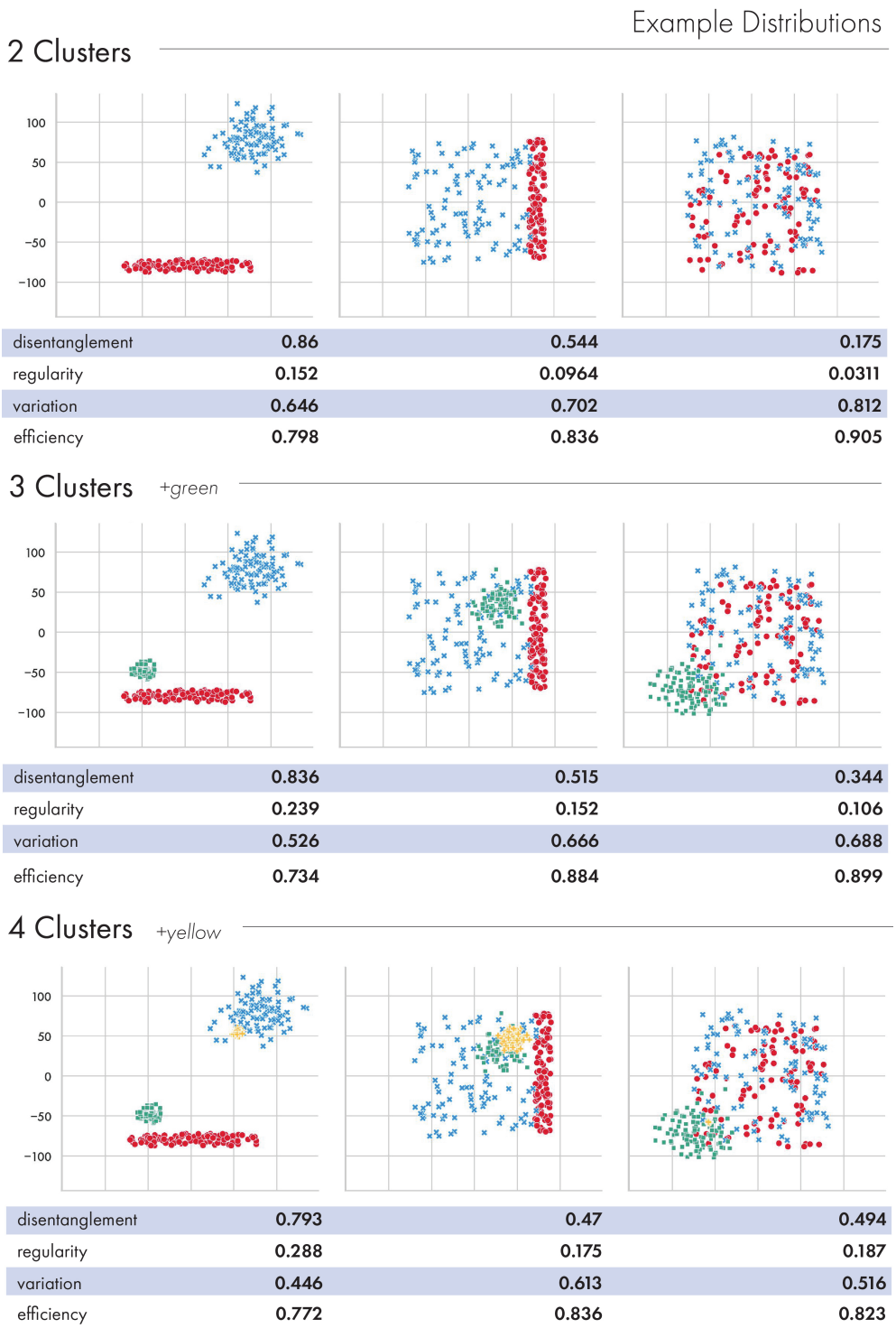


Figure 5.4: The soft entropy estimator applied to exemplar distributions. Examples shown for 2 to 4 clusters with each column being additive: each facet includes the distributions from above in the column. Disentanglement, regularity and variation scores for each facet are reported beneath it. In the top row disentanglement decreases with each facet from left to right, with the leftmost example being fully separable, and the rightmost fully entangled. Looking down the right column, adding the green cluster increases regularity as it is largely separable from the red and blue. The addition of green also decreases variation, as the average cluster size has decreased.

5.5 Validation & Comparison With Existing Methods

As this represents a novel approach to entropy estimation for vector spaces it is important to relate it to existing approaches to this problem. Doing so gives a sense of how precise the estimator is, and the degree to which the quantity it measures is related to existing notions of entropy. This in and of itself proves somewhat challenging - other approaches to estimating the shannon entropy of continuous spaces are also *estimators*, and so do not provide a ground truth value with which we can compare. In an effort to provide an indication of both how the soft entropy estimate relates to existing estimators, and ground truth estimates, we relate our entropy estimator to differential entropy; differential entropy is the usual continuous analog of shannon entropy. As a reminder the equation for shannon entropy is:

$$\mathcal{H}(x) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{5.8}$$

Differential entropy replaces the summation in the equation above with an integral. For a given density function f differential entropy $\mathcal{D}$ can be expressed as:

$$\mathcal{D}(f) = -\int_{x} f(x) \log f(x) dx \tag{5.9}$$

for certain density functions, like a gaussian, this has an analytic solution.

Given that computation of differential entropy requires commitment to a particular density function, and computing an integral, it is common to instead discretise representations and compute an entropy estimate using equation 5.8. This is the approach utilised in the preceding chapter, and the general intuition underpinning the soft-entropy estimator introduced here. The histogram entropy estimator (described in (Paninski, 2003)) allows conversion between discrete and differential entropy estimates. To do so a histogram estimator takes the equation for discrete entropy and inside of the logarithm divides the probability of bin i, $p(x_i)$ by the width of the bin $w(x_i)$.

$$\mathcal{D}_h(x) = -\sum_{i=1}^{n} p(x_i) \log \left(\frac{p(x_i)}{w(x_i)} \right) \tag{5.10}$$

This converts the estimate from shannon entropy, to differential entropy, via a method that can be applied to any "binning based" entropy estimate.

With this in mind we return to the problem at hand - benchmarking the soft entropy estimator against a ground truth. We select a gaussian distribution with a closed-form differential entropy, drawing samples from it, before applying the soft entropy estimator to those samples, and converting the resulting estimate to differential entropy. This gives us a ground-truth entropy value (differential entropy of the underlying gaussian), and an estimate of that value (soft entropy estimate, converted to differential entropy via the histogram method). With these two quantities we can get a sense of the relationship between the estimator introduced here and existing notions of entropy in continuous spaces. I will note that this multi-step procedure is imperfect for at least two reasons worth highlighting. First, the accuracy of the resulting estimate is a product of all steps in the process - the histogram conversion to differential entropy may introduce error separate from any error in the soft entropy estimator itself. Second, if we want to have a ground-truth with which to compare we are constrained to benchmarking against distributions with a closed form differential entropy. This is a surprisingly short list of density functions, and we only look at gaussians here. While gaussian mixtures would be more interesting - and more representative of the kinds of distributions found in models' hidden states- they offer no analytic solution to their differential entropy and so would require us to also select an estimation procedure for the 'ground-truth' against which we are benchmarking. With these caveats in mind, the results that follow still offer a substantive sanity check that the estimation procedure introduced here measures a quantity with clear relation to existing notions of entropy, and existing estimators.

5.5.1 Comparing Soft Entropy to Existing Entropy Estimators

In addition to relating our soft-entropy estimator with differential entropy, we also compare against two other discretisation-based estimators: fully discretising (akin to the preceding chapter), and k-means clustering (as described in Sajjadi et al. (2018)), results are shown in figure 5.5. In each plot the x axis shows the number of samples used to compute the estimate, the y axis shows the difference between the closed form entropy of the distribution samples are drawn from and the estimator. Positive values indicate the estimator has over-estimated the entropy of the space, while negative values reflect under-estimation. The x-axis gives a notion of sample efficiency, showing how the estimates change as more samples are provided from

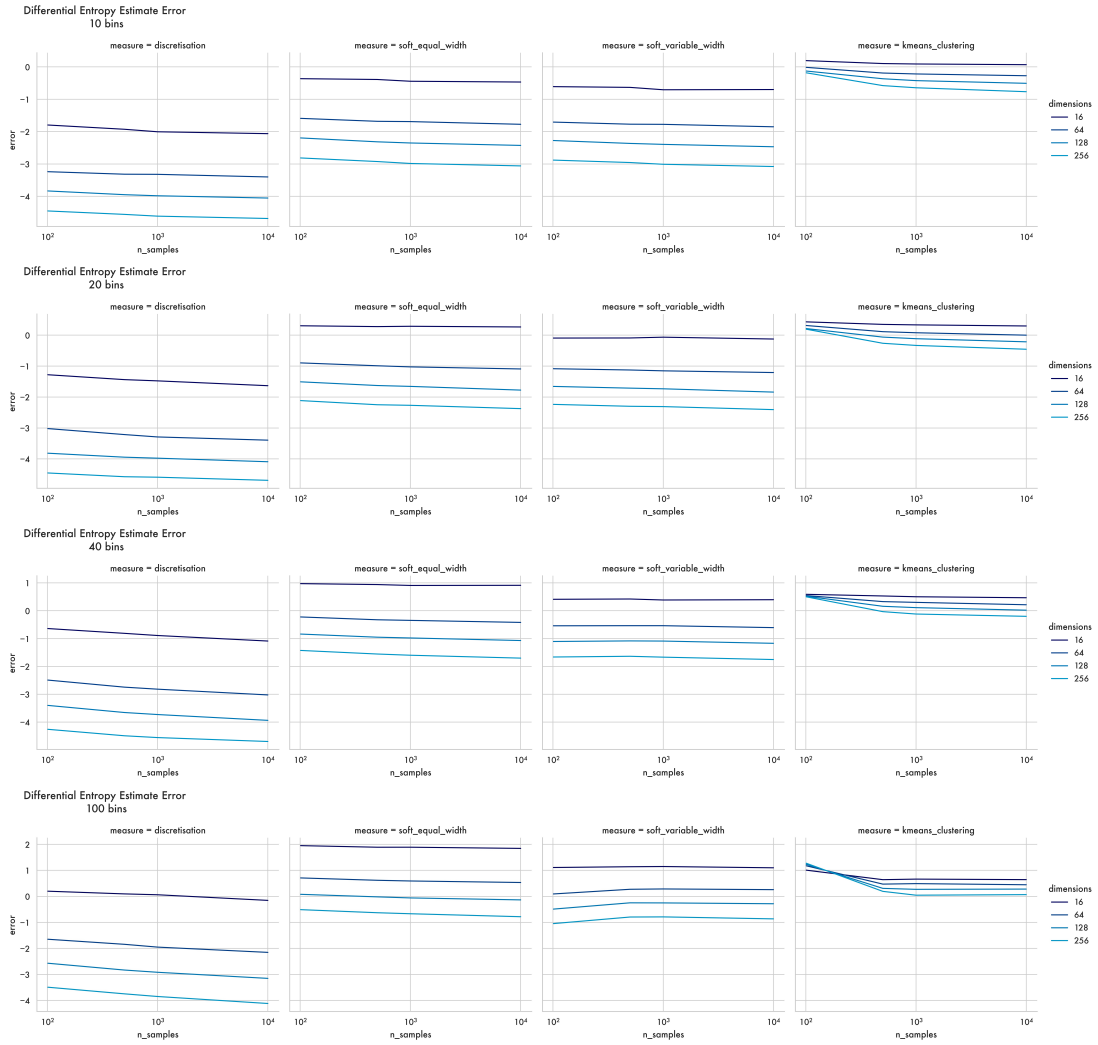


Figure 5.5: Comparison of Entropy Estimation Methods | In each facet the x-axis reflects number of samples used to compute the estimate, y-axis shows error of the estimator relative to the closed-form differential entropy of the underlying distribution. Each line is the mean error of the estimator applied to samples from 1000 different random multivariate normal distributions. Different lines indicate different dimensionalities for the distribution, ranging from 16 to 256. Columns are different entropy estimation methods, from left to right: full discretisation, soft entropy using equal width bins in the histogram estimator's conversion to differential entropy, soft entropy using variable width bins, k-means clustering. Rows reflect different numbers of bins, or clusters, ranging from 10 to 100.

the underlying distribution. Each column is a different entropy estimation method, from left to right these methods are full discretisation, soft entropy estimation with equal width bins used in the histogram conversion, soft entropy with variable width bins used in the conversion, then on the right k-means clustering. Each row in the visual uses a different number of bins, ranging from 10 to 100 to give an idea of how number of bins affects each estimator but also how sample efficient different estimators are. Each line is the mean of 1000 runs of the simulation each using a different randomly generated normal distribution. Additionally the different lines on each plot reflect different dimensionalities - meaning we perform this benchmarking on representations ranging from 16 dimensions to 256 to affirm that our estimator performs well in a variety of different spaces.

First of all, results show that the soft entropy estimator behaves similarly to the two existing approaches - discretisation and clustering. This is reflected by soft entropy consistently having a small error (the y axis in figure 5.5 is close to 0), meaning the estimator consistently gets close to the ground-truth entropy of the distribution from which samples were drawn. This acts as a confirmation that the quantity we assess with this estimator has clear links to existing formalisations of entropy. Being directly relatable to the closed form differential entropy of a distribution suggests that soft entropy is best seen as a more scalable, performant estimator of existing quantities rather than introducing some new quantity unrelated to existing work.

Additionally the soft entropy estimator appears to be more sample efficient than either discretisation or clustering, with lower error than existing approaches when using only 100 samples from the distribution. In the general case discretisation seems to consistently under-estimate the entropy of the space, especially in higher dimensions. By contrast clustering tends to over-estimate, with soft-entropy ending up in between. Consistently providing more accurate estimates than full discretisation, but slightly less accurate than clustering particularly in higher-dimensional spaces. It is worth noting that the slight increased precision of clustering relative to soft entropy comes at dramatically higher computational complexity: clustering requires the convergence of a clustering algorithm for each estimate, while soft entropy requires only a dot product. Further benchmarking showing the effects of computing subspace entropies on the sample efficiency of the soft entropy estimate can be found in appendix B.1.

5.6 Experiments

We use our measures of structure in a mapping, and soft entropy estimation to analyse properties of large language models in three ways. First we look at the how structure develops over the course of training in an encoder-only transformer, analysing 5 different initialisation of BERT over 2 million training steps in section 5.6.2. In section 5.6.3 we look at how model size affects representational space in both encoder and decoder only models, comparing structures inside decoder-only models ranging from 14 million parameters to 12 billion from the Pythia collection of models (Biderman et al., 2023). We also look at different sizes of BERT released in Turc et al., 2019, which allows us to make more precise comparisons varying number of layers, or hidden size independently, rather than just overall parameter count. Finally in section 5.6.4 we look at the relationship between representation structure and downstream task performance. We use the Multiberts (Sellam et al., 2022), 25 BERT base models that differ only in their initialisation, correlating their representation structure at the end of pretraining with performance 2 million steps of fine-tuning later.

5.6.1 Estimating Entropy To Enable Model Comparisons

Making fair comparisons between different models is often challenging given differences in number of layers and dimensionalities. Previous information theoretic analyses in deep-learning often report estimates for each layer separately (e.g. Shwartz-Ziv & Tishby, 2017; Voita et al., 2021), which can make overall interpretation and comparison difficult. Instead we look at a model's hidden state as a single random variable distributed across layers. In practice though larger hidden states will have more information, what we want in order to make fair comparisons is a relative entropy estimate, reflecting how much information a representation space encodes proportional to its size.

To this end we report two different quantities, *layer entropy* and *subspace entropy*. For layer entropy we compute an estimate at each layer, then mean across them. This lets us directly compare models of the same dimensionalities but differing depths. For subspace entropy we apply the soft entropy estimator in a multi-headed arrangement as described in section 5.4.1. This lets us break representation spaces into lower dimensional subspaces; in the results here subspace entropy is computed over 32-dimensional spaces across every layer in the model

then aggregated. This lets us compare entropies over the same sizes subspace for models with different overall dimensionalities. While breaking a vector into subspaces may break some cross-dimensional dependencies we believe that this effect is relatively small - results testing this on sample distributions are included with other entropy estimate benchmarking in the appendix B.1.

5.6.2 When Structure Emerges During Training

We look at 5 different initialisations of BERT over the course of 2 million training steps (model checkpoints also released as part of Sellam et al. (2022)). At each checkpoint we compute our 3 structure measures with respect to token, bigram and trigram labels from the wikipedia data. We choose to use these labels because they are known for virtually every text dataset that's fed into a model.

Main findings are shown in figure 5.6. Overall trajectories for each measure are remarkably similar to the phases of training described in Conklin and Smith (2024), which applied a similar analysis to 3 layer encode-decoder transformers trained on a single semantic-parsing task - suggesting some generality to this characterisation of training dynamics in deep-learning. At the start of training ($< 100,000$ steps) representations quickly align with token-level information, with distinct tokens becoming represented in distinct, disentangled parts of space. Past this point the dynamic shifts as representations begin to contextualise. Token disentanglement drops significantly, while bigram and trigram disentanglement increase. These likely contrast because in order to better represent lower-level information like bigrams, separate tokens need to spread out (variation increases) and overlap (disentanglement decreases). This process of contextualisation is the defining dynamic of the majority of training. Unlike findings in Conklin and Smith (2024), later stages of training are not characterised by overall compression of the space (overall entropy decreasing), this may be a difference between single task models and LLMs or may reflect that BERT was substantially undertrained, as noted in Liu (2019).

Decoder-Only Model Timecourse

To compare how structure develops over time in encoder only models (like BERT) and decoder-only models, we also look at the trajectory of 5 different initialisations of the 410 million parameter Pythia model. We select this parameterisation because

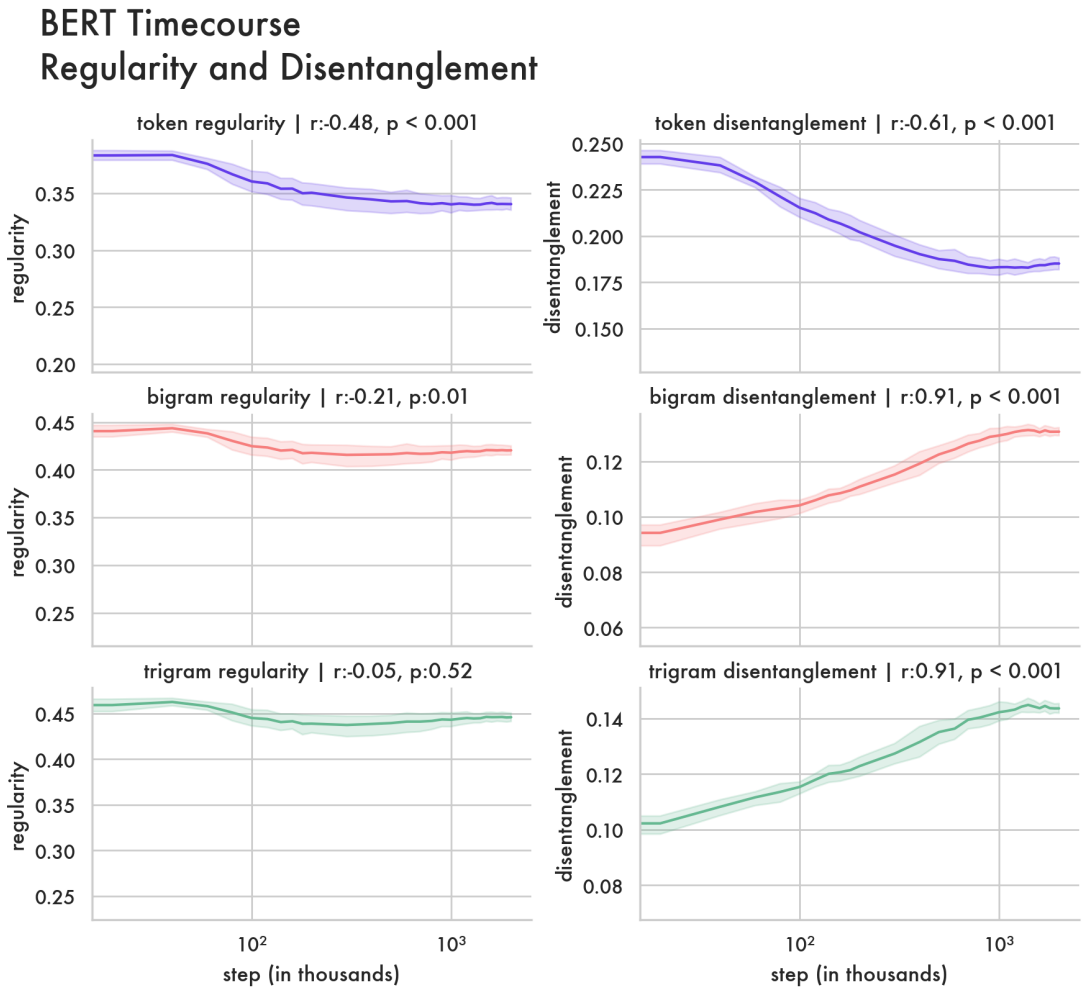


Figure 5.6: Information Structure with respect to 256,000 sentences from wikipedia over 2 million steps of training. Each line represents the mean of 5 different initialisations of BERT with shading representing 95% confidence intervals. Also included above each facet is a spearman correlation between x and y. Estimates here use layer entropy, given there's no need to compare different dimensionalities. Shown here are regularity and disentanglement, with variation and information proportion on the following page. Note that checkpoints released by Sellam et al. (2022) are only every 20,000 steps of training. Timecourses for decoder-only models visualised in figure 5.7 provide visualisations of information structure in early training. (continued on next page)

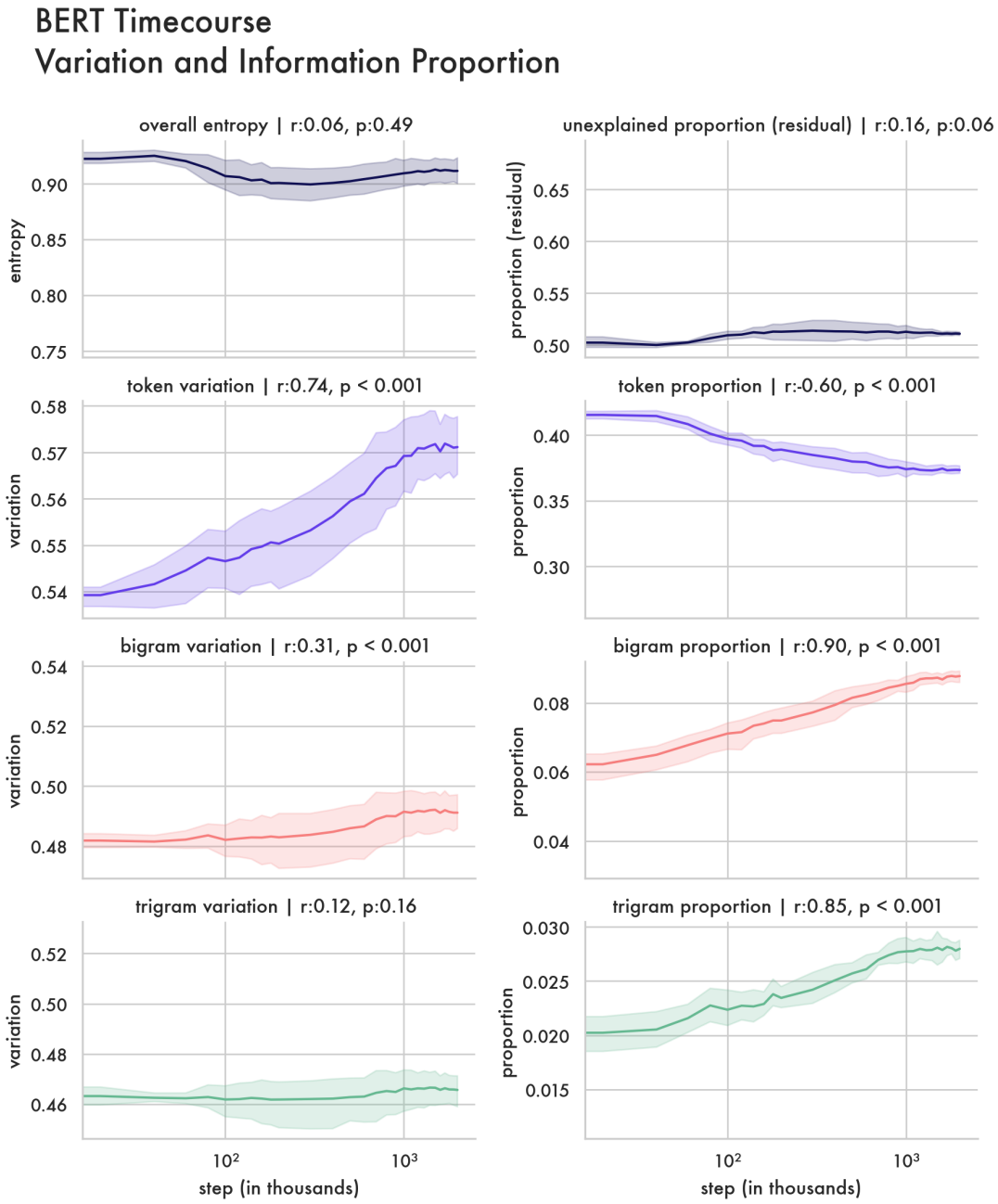


Figure 5.6: (continued): Shown here are the variation and information proportion timecourses for the same 5 BERT models over training. Also shown in the top two facets are the overall entropy of the space and the unexplained (residual) information proportion over training. Overall entropy exhibits minor fluctuations but no overall pattern of compression - which is present in the decoder-only model timecourse shown in the next figure.

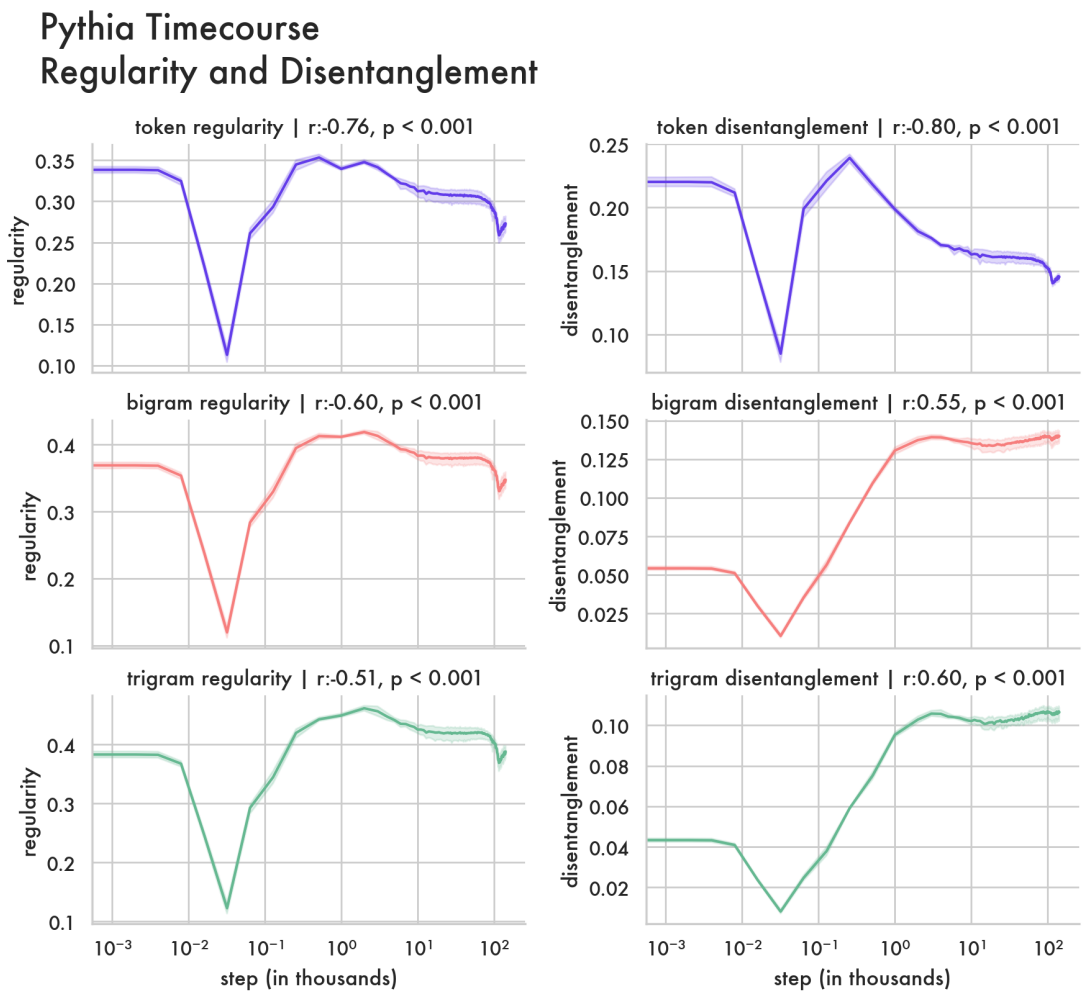


Figure 5.7: Information Structure with respect to 256,000 sentences from wikipedia over 2 million steps of training. Each line represents the mean of 5 different initialisations of the Pythia 410m parameter model with shading representing 95% confidence intervals. Also included above each facet is a spearman correlation between x and y. Estimates here use layer entropy. Note that unlike the BERT model timecourse shown in the previous plots these include log-spaced model checkpoints between step 0 and step 20,000. As a result the dramatic spike across all measures at 10e-2.5 steps does not appear on the BERT timecourses - it may still take place, but we lack the checkpoints to verify. (continued on next page)

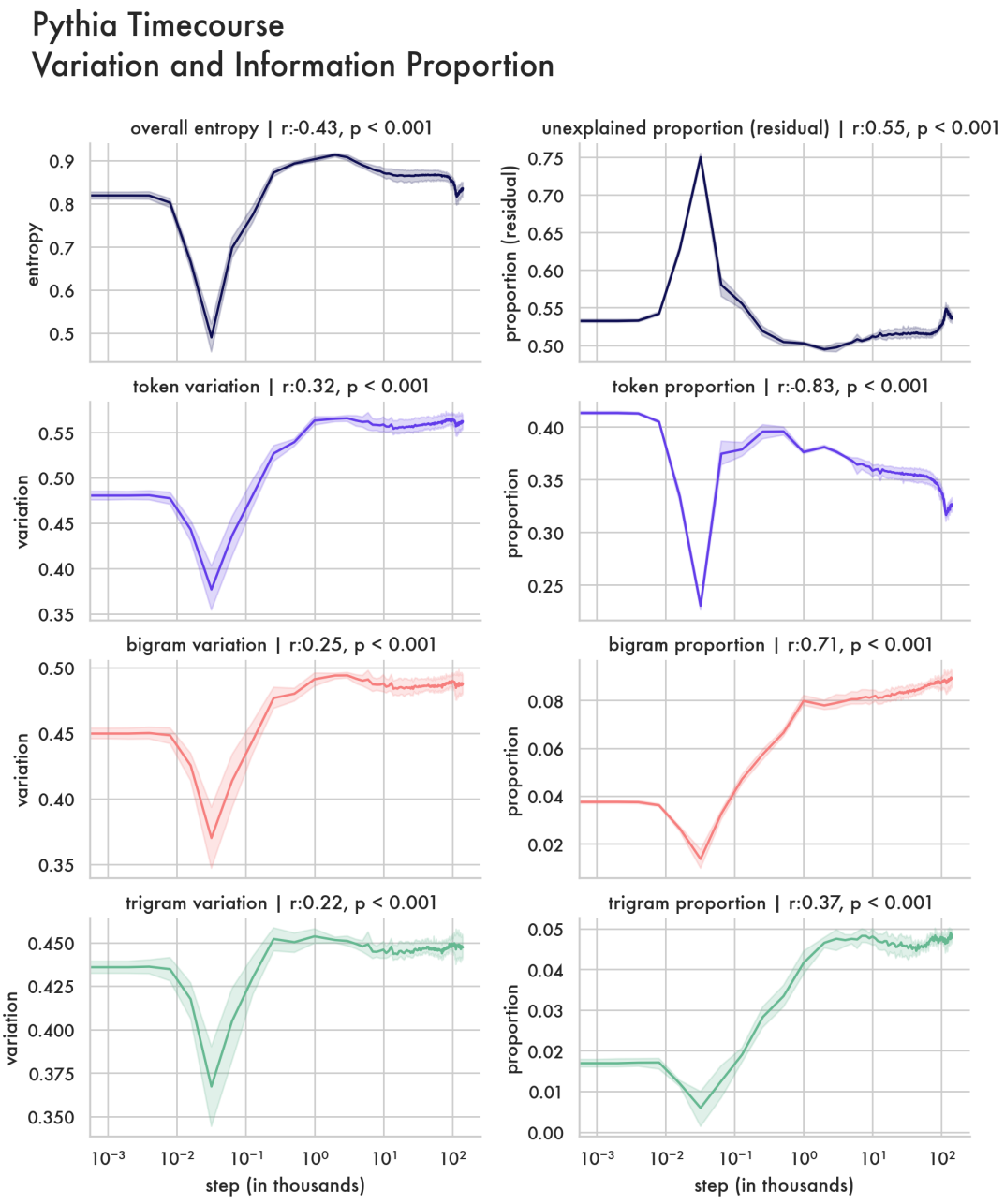


Figure 5.7: (continued): Shown here are the variation and information proportion timecourses for 5 Pythia 410 million parameter models models over training. Also shown in the top two facets are the overall entropy of the space and the unexplained (residual) information proportion over training. Overall entropy exhibits a significant trend of compression later in training. Note that during the spike early in training, the space compresses and the unexplained proportion accounts for 75% of the space before the token proportion starts to steadily increase.

it is comparable in scale to BERT - differing only in embedding matrix parameters - and uses the same number of layers and attention heads. Main Findings are shown in figure 5.7, it is important to note though that the the BERT models released by Sellam et al. (2022) (visualised previously) only provide checkpoints every 20,000 steps early in training. By contrast the Pythia models include log-spaced checkpoints early on, allowing us to look at what happens during the first 1000 steps. This early phase in particular closely resembles the results from Conklin and Smith (2024) for models trained on a single task. With the space quickly compressing, then expanding as representations align with token-level information, before a long contextualisation phase. The pattern is also broadly similar to the timecourse of the encoder-only model, although here the decoder model compresses its overall representation space significantly in the latter phase of training. This may reflect architectural differences between the models, or differences in their objectives - with encoder-only models predicting a masked word given the surrounding context, and decoder-only models predicting the next word based on preceding context.

Overall the timecourse results are striking given their similarity across different kinds of large-language models, and to models from previous work trained on a single task. This suggests some real generality to the two-phase framing of Deep-Learning training trajectories adopted in Conklin and Smith (2024).

5.6.3 Model Size Conditions Representational Structure

How does scale affect representational structure? We look at this in both decoder-only and encoder only models, again performing structure estimates using 256,000 sentences from english wikipedia, and labels for token, bigram, and trigram information. Figure 5.8 shows results for the decoder-only models, with both layer and subspace entropy reported. Both are reported for reference, and to give an intuition to how they relate - but as discussed above layer entropy does not allow a like-for-like comparison between different dimensionalities. As you would expect larger models have higher layer entropy - each layer of the 12b model has 5120 dimensions compared with 128 in the smallest, it would be surprising if they contained the same amount of information. Subspace entropy - which provides a more directly comparable estimate between model sizes - reveals a different pattern with the largest models beginning to compress their representations more, with the

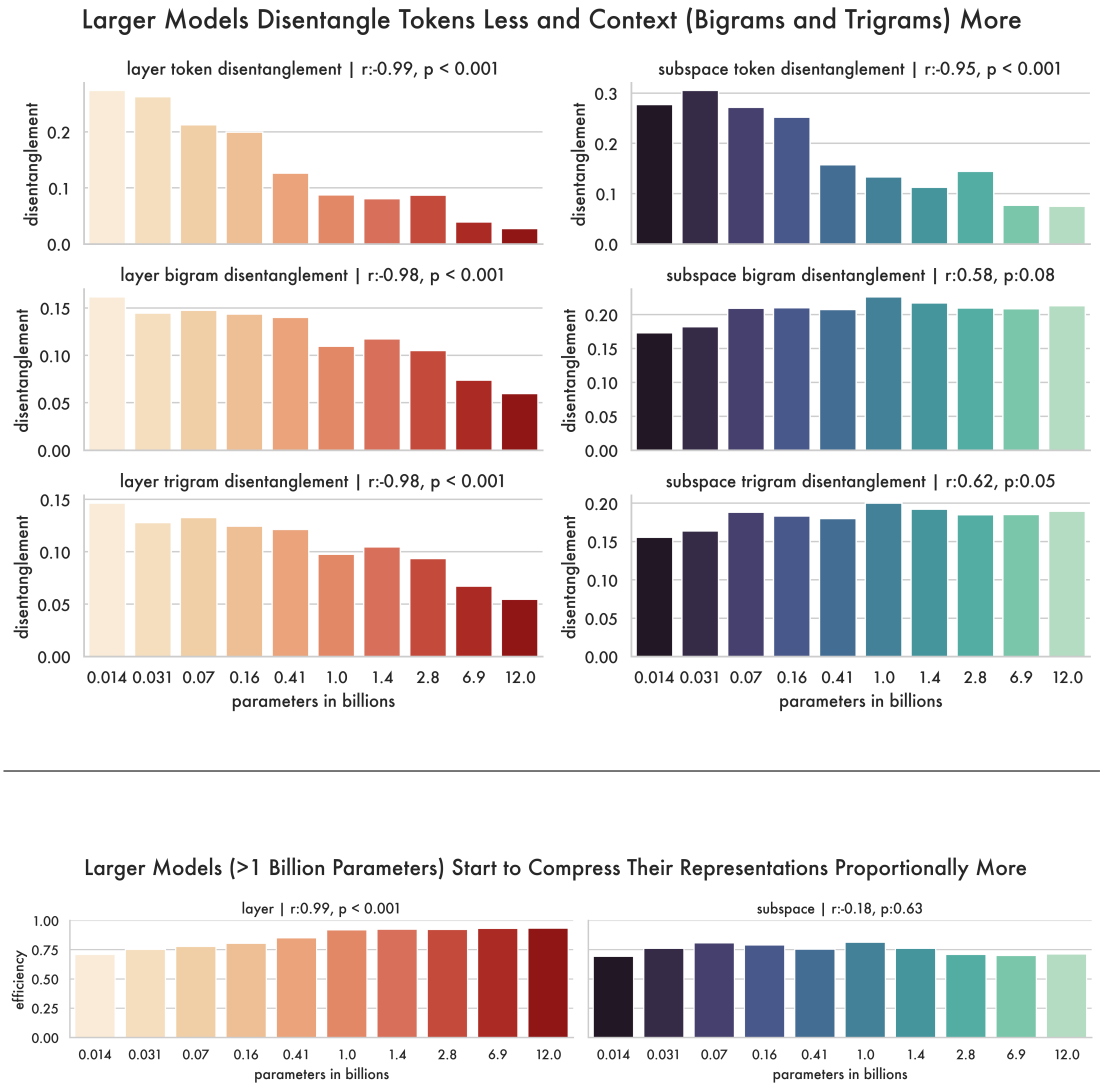


Figure 5.8: Analyses computed on pythia decoder-only models ranging from 14m parameters to 12 billion. Red/Orange bars show mean layer entropy, blue/green bars show mean subspace entropy. Above each plot is a spearman correlation between x and y. **Top:** y-axis shows disentanglement for different model sizes. Subspace entropy shows contextual information (bigrams and trigrams) are more disentangled in larger models, with size significantly correlating with disentanglement **Bottom:** y-axis shows overall entropy of each model size. While layer entropy increases monotonically with size as expected - subspace entropy begins to compress in larger models.

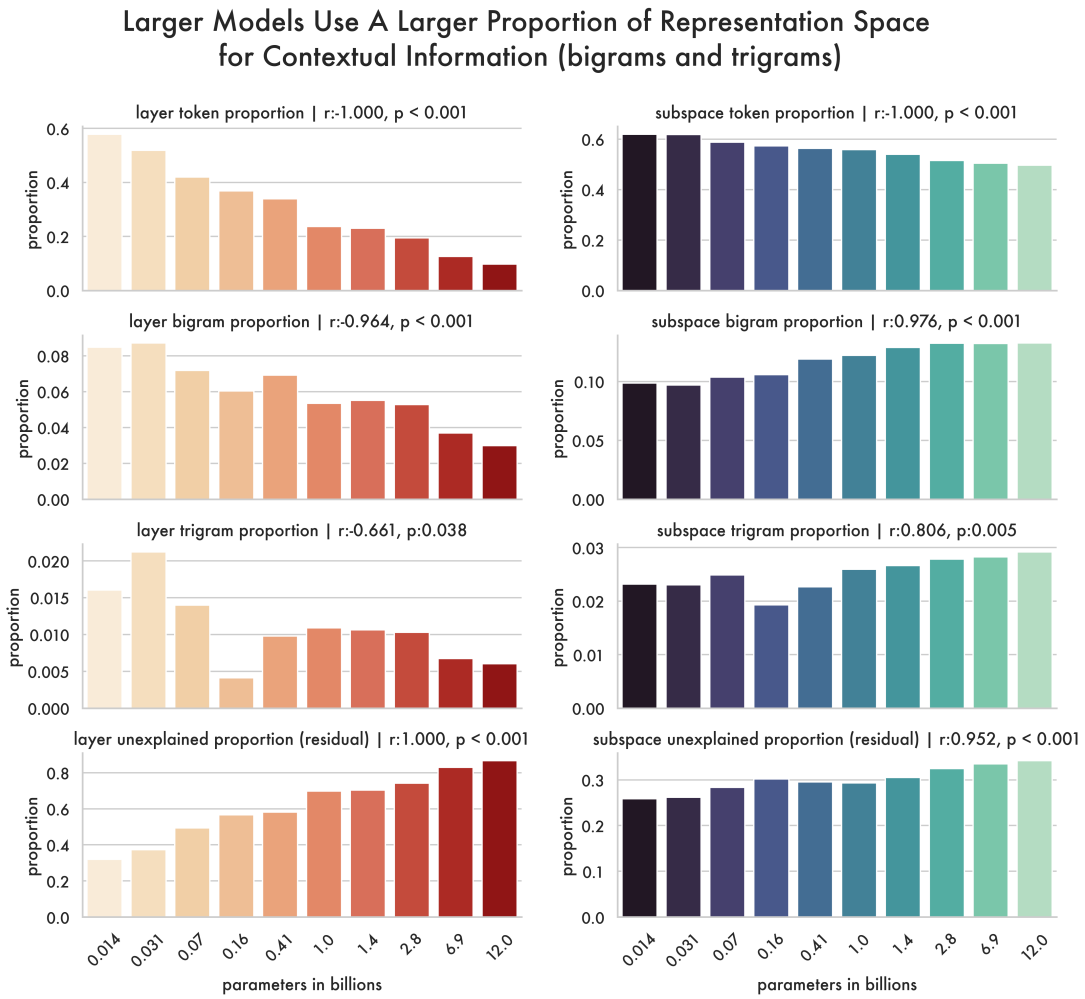


Figure 5.9: Analyses computed on pythia decoder-only models ranging from 14m parameters to 12 billion. Red/Orange bars show mean layer entropy, blue/green bars show mean subspace entropy. Above each facet is a spearman correlation between x and y. y-axis shows the proportions of representation space that encode token/bigram/trigram information for each model size(on the x-axis). Subspace entropy shows larger models use proportionally more space for token and bigram information.

12b version almost matching the subspace entropy of the smallest model. Because the representation space is larger information can be more distributed across space, meaning each subspace can compress more on a relative basis. We draw an analogy to Shannon's source coding model (Shannon, 1948) where meanings are mapped to signals; signal space has two key parameters - signal length and alphabet size. A smaller alphabet has less uncertainty, exemplified by morse code's binary alphabet where operators only need to tell the difference between a dot and dash. Smaller alphabets require a longer signal - sentences in morse code are far longer than in english - this is the tradeoff for a more robust encoding. In our subspace entropy analysis larger models have more subspaces, analogous to a longer signal. This can enable compression of each subspace like shrinking the alphabet at each character in a signal, which may help explain their improved performance.

Figure 5.9 plots the proportion of representation space that encodes token, bigram and trigram information and the information we can't explain in terms of any of the labels - the residual. This is estimated by comparing the regularity for each set of labels with the information left over which isn't regular with respect to any label. Looking at the subspace analysis larger models devote more of their representational space to contextual information, and less to token information. They also have more information we can't explain in terms of these labels. That could reflect other information from the training data not explainable in terms of lexical/contextual labels, or it could reflect artefacts not explainable in terms of any label. The middle plot shows disentanglement across model sizes, with larger models subspaces disentangling contextual information more.

An issue with the pythia suite of models is that while they differ in size, that difference is driven by changes in both depth and dimensionality[2]. In an effort to isolate the effects of these different kinds of scaling we use sets of BERT models released by Turc et al. (2019). Figure 5.10 shows effects on representational structure for models with a dimensionality of 768, but layers ranging from 2 to 12, and models with 12 layers but dimensionalities from 128 to 768. Overall both kinds of scaling have a similar effect, namely increasing contextual information, with dimensionality's effect being much stronger than depth. Although increasing model depth does significantly increase the unexplained (residual) proportion.

[2]It's also worth noting models also differ in the dimensionality of attention heads. this may have an effect on structure but we lack controlled comparisons to draw conclusions.

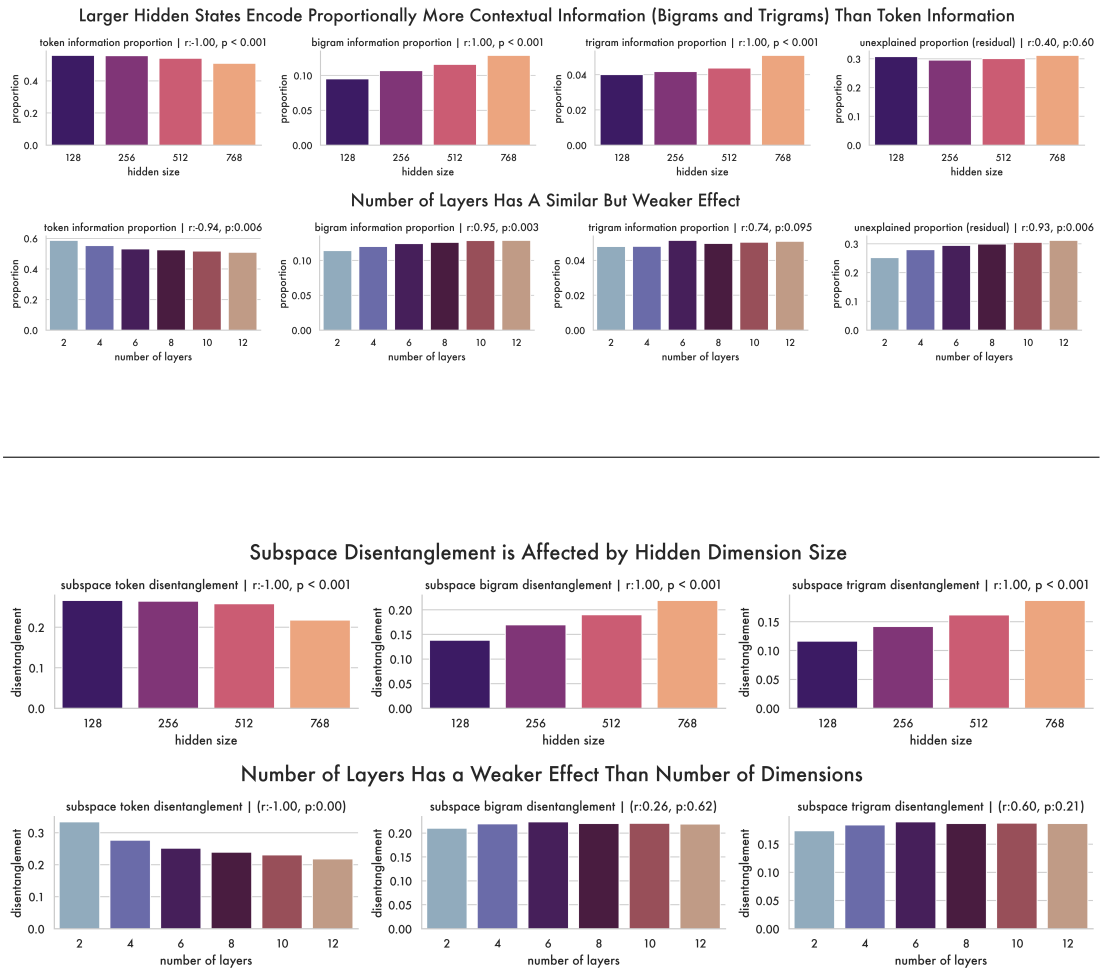


Figure 5.10: Scaling comparison of depth vs. dimensionalities on BERT models. All plots use subspace entropy, color reflects depth or dimension (both shown on the x axis) – atop plots is a spearman correlation between x and y **Top:** y-axis proportion of representation space that encodes token/bigram/trigram information. Larger models devote a larger proportion of their representation space to contextual information - here captured by bigram and trigram labels - while reducing token-level information. This effect is stronger for increasing size through hidden size, rather than number of layers. However, note that the increased number of layers does much more significantly increase the unexplained (residual) proportion. This could reflect models with greater depth representing more abstract syntactic and semantic information that cannot be captured by the labels used here. **Bottom:** disentanglement of label information (y axis) across different sizes. Here increasing model scale reduces token disentanglement. For bigram and trigram disentanglement, it depends on the scaling method used. Increasing hidden size significantly increases disentanglement of bigrams and trigrams, while increasing the number of layers has little effect.

This could reflect models with greater depth representing more abstract syntactic and semantic information that cannot be captured by the labels used here. By contrast, both scaling methods decrease token disentanglement significantly, but only increasing the size of the hidden dimension affects bigram and trigram disentanglement, significantly increasing both — increasing model depth has no clear effect on bigram and trigram disentanglement

5.6.4 Predicting Downstream Performance

We look at spearman correlation coefficients between structural properties of representation space and downstream task performance. In order to isolate as many variables as possible we use the Multibert models (Sellam et al., 2022) which is 25 different initialisations of BERT (Devlin et al., 2019). By comparing performance between models that differ only in terms of the random seed used to initialise them we can have some confidence that effects we measure between representational structure and downstream performance are likely driven by structure rather than model size, training data, or training objective. The Multiberts provide checkpoints at the end of pre-training, and evaluations for fine-tuned versions of each of these across the GLUE benchmarks (A. Wang, 2018)[3]. We take 10 million sentences sampled randomly from the C4 dataset (Raffel et al., 2020) and compute our structure measures with respect to token, bigram, and trigram labels. The C4 dataset contains data from a general crawl of the internet - of which we use the english subset. This is a diverse collection of text sources which enables us to get a general structure estimate for each model. We correlate representational structure with respect to C4 at the end of pre-training with performance on GLUE tasks after fine-tuning. It is important to note that this means we are able to predict which of the models will do better on a downstream task before the models are fine-tuned for 2 million steps on data from that task. As far as we're aware this is the first analysis able to predict downstream performance from pre-training. Additionally the structure measures we use in this correlation are not estimated using data from those benchmarks. Despite the estimate using non-task data, on models 2 million steps of fine-tuning removed from evaluation we still find a number of significant correlations. This suggests that representational structure,

[3]For each seed used during pre-training, the multiberts train 5 different seeds during fine tuning. We compute a structure estimate with respect to the 25 different models at the end of pre-training, and correlate this with the performance of all 5 fine-tuned versions of each model.

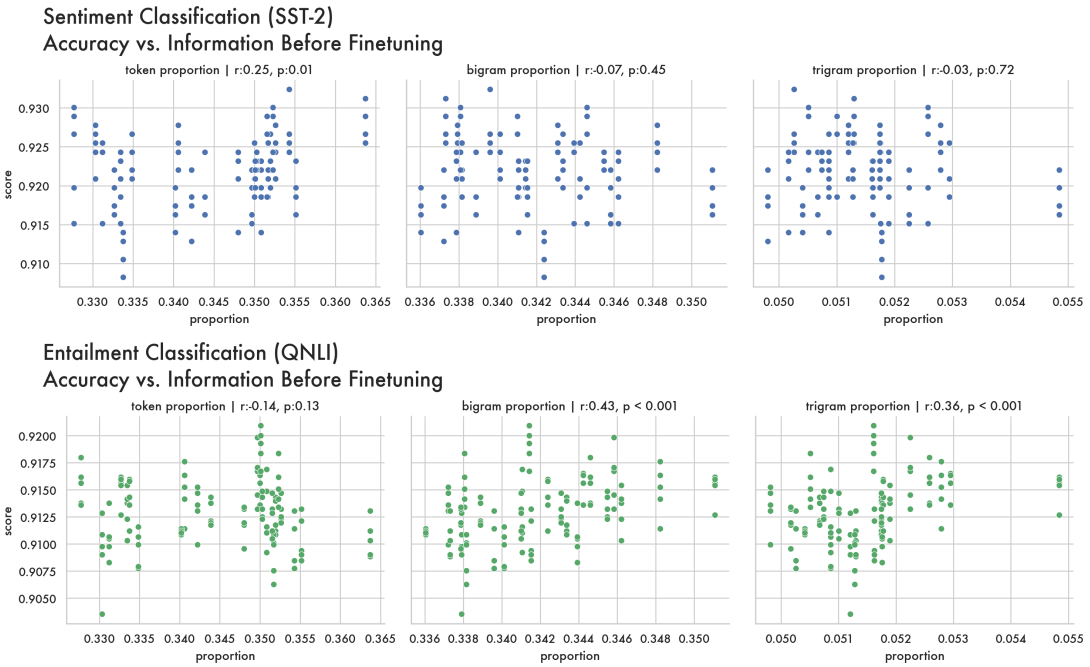


Figure 5.11: Scatterplots showing model performance on two GLUE benchmarks, sentiment classification (SST-2) and Entailment (QNLI) (y-axis) vs. information proportion at the token bigram and trigram level. Above each facet is a spearman correlation between the x and y axes. Sentiment classification just needs to decide if a sentence is positive or negative, and can rely on token-level information to do so. Entailment classification by contrast requires a model to determine if two sentences have the same meaning. Sentiment classification accuracy correlates with the model having a higher proportion of token information, while Entailment classification correlates with a higher proportion of bigram and trigram information.

as captured by our measures, has an effect on generalisation performance.

For clarity we focus on two of the glue tasks in particular, a sentiment classification task (SST-2), and an entailment task (QNLI). For sentiment classification, a model only needs to determine if the general sentiment of a sentence is positive or negative. Prior to the advent of large language models, this task was often approached using bag-of-words, or non-contextual word embeddings like word2vec (Barry, 2017; Mikolov, 2013). By contrast entailment tasks like QNLI require a model to determine if two sentences entail one another. This requires determining if the meaning of two sentences overlap, which is harder to do with only non-contextual word level information. Figure 5.11 shows task performance against information proportion at the token, bigram and trigram level, along with

		$\mathcal{H}$	proportion		disentanglement		variation	
		overall	token	context	token	context	token	context
QNLI	r	0.039	-0.138	0.410	-0.016	0.161	0.136	-0.020
	p	0.663	0.125	< 0.001	0.855	0.072	0.131	0.827
SST-2	r	-0.152	0.247	-0.048	-0.044	-0.196	-0.242	-0.197
	p	0.090	0.005	0.598	0.628	0.028	0.007	0.027

Table 5.1: Spearman Correlations between representational structure across 25 different initialisations of BERT at the end of pre-training (before fine-tuning) and downstream task performance on two GLUE benchmarks (after 2M steps of fine-tuning). The same benchmarks are visualised in 5.11. For readability bigram and trigram information scores are averaged before performing the correlation to give a general 'context' estimate. Positive correlations indicate a higher score on the measure correlates with higher task performance.

spearman correlations between each measure and task performance. Additionally table 5.1 shows spearman correlations between task accuracy and representational disentanglement and variation. Sentiment classification accuracy correlates positively with a higher proportion of representation space being dedicated to token-level information - but does not correlate significantly with bigram or trigram information. Entailment classification inverts this pattern showing strong significant correlations with bigram and trigram information, but no significant correlation with token level information.

5.7 Conclusion

We have introduced a set of measures for thinking about and describing structure in large language models information theoretically. This approach can show how representations become structured over the course of training, how that structure is influenced by model scale, and what structural properties correlate with downstream performance. It's backboned by a new scalable, parallelisable, and differentiable approach to entropy estimation, that can be applied at the subspace level to enable like-to-like comparisons between models of different sizes. We related the structural properties found here to structures in linguistics, and Shannon's model of communication in an effort to contextualise these structures in terms of other areas of science. We think that continued work mapping large

language models to spaces and measures about which we have stronger intuitions than vector space is crucial in helping us understand, interpret, and improve models going forward.

Chapter 6

Biasing Representational Structure with Meta Learning

Controlling Information Structure

If I when [redacted]
the sun is a flame-white disc [redacted] if I [redacted]
dance [redacted] waving my shirt round my head
and singing softly to myself [redacted]

-William Carlos Williams

Given that the last two chapters identified particular structures that predict generalisation performance, can we directly intervene during training to select for those structures? One way of doing this could be to reduce a model's capacity. Chapter 3 showed how limiting the capacity of a model in a multi-agent setting can have a regularising effect on on the discrete signals it produces, with models with smaller hidden dimensions converging to languages with less variation. By contrast in chapter 4, we looked at structure inside a model, where models with a larger hidden dimension compressed their representation space more and became more regular with respect to contextual information than their smaller counterparts. At the end of chapter 4 we observed that this may be because, when studying model-internal representations, limiting hidden size can have an effect analogous

to limiting channel capacity in the discrete case (i.e. restricting signal length). This point was made clearer in the last chapter, where varying a large language model's hidden size has a much stronger effect on representation structure than varying its depth, despite the fact that both affect a model's capacity. In general modifying a model's parameter count can have effects on performance unrelated to capacity; the lottery ticket hypothesis (Frankle & Carbin, 2018) suggests larger models have a better chance of getting a good initialisation by virtue of having more parameters to be randomly initialised. Larger models could perform better because they luck into a better initialisation rather than for any reason related to their capacity during learning.

In an ideal case we would be able to isolate capacity as an independent variable without altering other factors that can affect model performance. This chapter uses meta-learning to introduce an inductive bias to the model through the objective it optimises, allowing us to look at the effect of different biases on the same underlying model architecture. The experiments here endeavour to introduce a bias that limits the model's ability to memorise examples in its training data by encouraging update steps on an example that improves performance on similar examples. Results show this approach improves generalisation performance on two different architectures across two different datasets.

6.0.1 Relating Information Structure to Compositionality, Memorisation & Generalisation

As something of a procedural note, the paper this chapter is based on is the first project I worked on during my PhD — as a result it was written before I adopted the approach to representational structure used throughout this thesis. This chapter talks instead about models' behavioural properties - like generalising or memorising - rather than what those may look like on a representational level. Given that, it is worth discussing how the two approaches to terminology relate - what compositionality, memorisation and generalisation mean in terms of regularity, variation, and disentanglement. As discussed in chapter 1 regularity reflects how predictable a mapping between spaces is, and compositionality enables predictable mapping by reusing parts across the system which can be composed together. However just because a mapping is compositional doesn't entail its being maximally regular - as discussed at length in chapter 3 natural languages manage

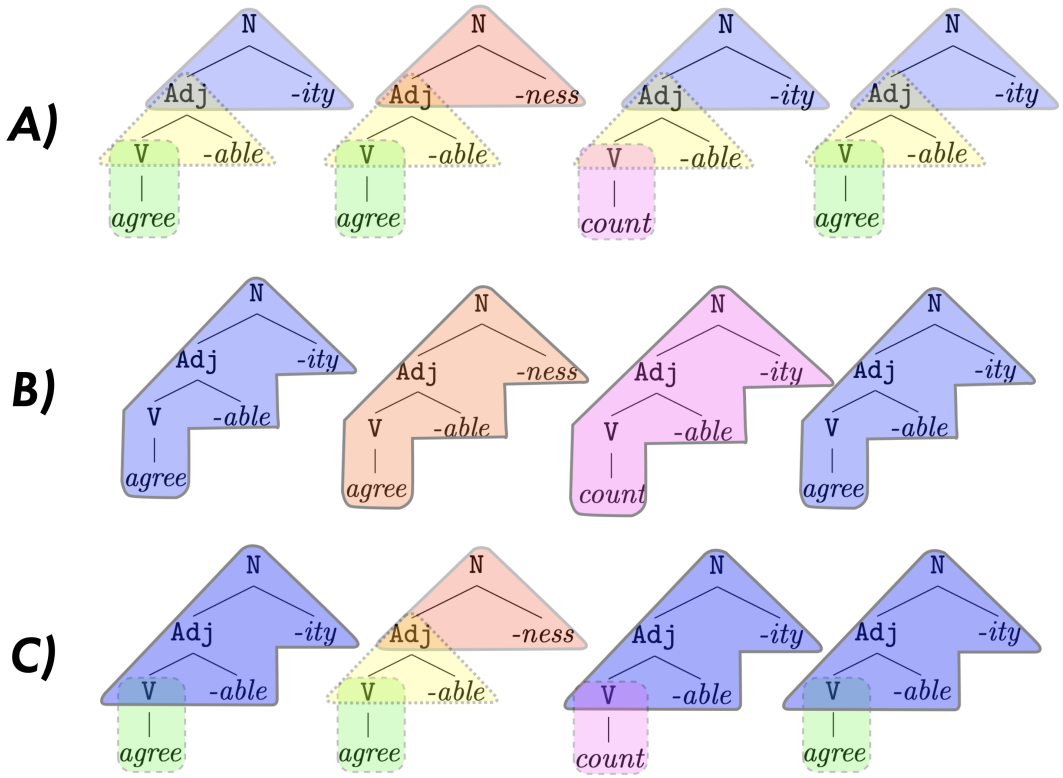


Figure 6.1: Figures reproduced from O'Donnell (2015) show three different approaches to decomposing a word. Colour in each case indicates lexicon entries. **A)** decomposes words fully into each individual affix, **B)** stores the entire word in memory without decomposition and **C)** splits the difference, partially decomposing each entry and memoizing frequent subunits

to be compositional while supporting extensive variation. Given that the models considered in this chapter can generalise to thousands of in-distribution examples it is unlikely that they are incapable of representing information compositionally (Brighton, 2002). O'Donnell (2015) provides a way of thinking about gradations of regularity within a compositional system. Looking at morphology O'Donnell (2015) considers a context free grammar with a lexicon that stores the lexical items a grammar can combine, figure 6.1 shows 3 different approaches to decomposing a word into lexical entries. Colour indicates what portions of each word are stored in the lexicon A) fully decomposes a word into each root and affix, while B) stores the entire word as a single item, C) splits the difference by decomposing the full word, but memoizing frequent subunits. We can look at A) as being compositional and fully regular, B) as being non-compositional with maximal variation, and C) as being compositional with both regularity and variation. In O'Donnell (2015)

an approach based on C) ends up providing the best fit to a corpus of natural language data in English.

Implicitly when we talk about memorising vs. generalising, or more compositional vs. less we talk about how much is stored in the lexicon vs. generated by the grammar. In reality, figure 6.1 gives an example of how this distinction need not be binary, with natural language reflecting gradations of both. A point made further by usage-based approaches to language (e.g. Croft, 2001; Goldberg, 1995; Tomasello, 2005), which erode the binary distinction between grammar and lexicon in favour of constructions that unify meaning and form and are learned probabilistically on the basis of experience (Goldberg, 2003). In light of this, the approach taken throughout this thesis conceptualises structure probabilistically by using information theory, and quantifies properties of a system that are naturally graded (regularity, variation, and disentanglement) rather than trying to describe a system in terms of binary distinctions. As Russell (1912) notes, learning from data — even when learning rules — is ultimately about probability:

> The man who has fed the chicken every day throughout its life at last wrings its neck instead [...] It must be conceded, to begin with, that the fact that two things have been found often together and never apart does not, by itself, suffice to prove demonstratively that they will be found together in the next case we examine. The most we can hope is that the oftener things are found together, the more probable it becomes that they will be found together another time, and that, if they have been found together often enough, the probability will amount almost to certainty. It can never quite reach certainty, because we know that in spite of frequent repetitions there sometimes is a failure at the last as in the case of the chicken whose neck is wrung. Thus probability is all we ought to seek.
>
> Russell (1912) | p. 30

This is not to say talking about memorisation or generalisation is incorrect (the remainder of this chapter does at some length), but that in models, as in natural language, it is nearly always both — not so much a question of either/or but a question of degree. To me, questions of degree are best seen as questions of probability.

The remainder of this chapter is a paper **Meta-Learning to Compositionally Generalise** that appeared at the International Meeting of the Assosciation of Computational Linguists in 2021. Authors are myself, Bailin Wang, Kenny Smith and Ivan Titov - I

wrote the majority of the paper including the theoretical framing and motivation. I contributed to code for the experiments but majority of code was written by Bailin based on a codebase from a previous meta-learning project - Kenny and Ivan supervised the project and gave writing feedback prior to submission to the conference. The paper is presented here minimally changed from the conference version that underwent peer-review. Changes are largely related to formatting to make the content more readable outside of the original conference paper template.

6.1 Meta-Learning to Compositionally Generalise

Compositionality is the property of human language that allows for the meaning of a sentence to be constructed from the meaning of its parts and the way in which they are combined (Cann, 1993). By decomposing phrases into known parts we can generalize to novel sentences despite never having encountered them before. In practice this allows us to produce and interpret a functionally limitless number of sentences given finite means (Chomsky, 1965).

Whether or not neural networks can generalize in this way remains unanswered. Prior work asserts that there exist fundamental differences between cognitive and connectionist architectures that makes compositional generalization by the latter unlikely (Fodor & Pylyshyn, 1988). However, recent work has shown these models' capacity for learning some syntactic properties. Hupkes et al. (2018) show how some architectures can handle hierarchy in an algebraic context and generalize in a limited way to unseen depths and lengths. Work looking at the latent representations learned by deep machine translation systems show how these models seem to extract constituency and syntactic class information from data (Belinkov et al., 2018; Blevins et al., 2018). These results, and the more general fact that neural models perform a variety of NLP tasks with high fidelity (eg. Dong & Lapata, 2016; Vaswani et al., 2017), suggest these models have some sensitivity to syntactic structure and by extension may be able to learn to generalize compositionally.

Recently there have been a number of datasets designed to more formally assess connectionist models' aptitude for compositional generalization (Hupkes

et al., 2019; Kim & Linzen, 2020; Lake & Baroni, 2018). These datasets frame the problem of compositional generalization as one of out-of-distribution generalization: the model is trained on one distribution and tested on another which differs in ways that would be trivial for a compositional strategy to resolve. A variety of neural network architectures have shown mixed performance across these tasks, failing to show conclusively that connectionist models are reliably capable of generalizing compositionally Keysers et al., 2020; Lake and Baroni, 2018. Natural language requires a mixture of memorization and generalization (Jiang et al., 2020), memorizing exceptions and atomic concepts with which to generalize. Previous work looking at compositional generalization has suggested that models may memorize large spans of sentences multiple words in length (Hupkes et al., 2019; Keysers et al., 2020). This practice may not harm in-domain performance, but if at test time the model encounters a sequence of words it has not encountered before it will be unable to interpret it having not learned the atoms (words) that comprise it. Griffiths (2020b) looks at the role of limitations in the development of human cognitive mechanisms. Humans' finite computational ability and limited memory may be central to the emergence of robust generalization strategies like compositionality. A hard upper-bound on the amount we can memorize may be in part what forces us to generalize as we do. Without the same restriction models may prefer a strategy that memorizes large sections of the input potentially inhibiting their ability to compositionally generalize.

In a way the difficulty of these models to generalize out of distribution is unsurprising: supervised learning assumes that training and testing data are drawn from the same distribution, and therefore does not necessarily favour strategies that are robust out of distribution. Data necessarily under-specifies for the generalizations that produced it. Accordingly for a given dataset there may be a large number of generalization strategies that are compatible with the data, only some of which will perform well outside of training (D'Amour et al., 2020). It seems connectionist models do not reliably extract the strategies from their training data that generalize well outside of the training distribution. Here we focus on an approach that tries to to introduce a bias during training such that the model arrives at a more robust strategy.

To do this we implement a variant of the model agnostic meta-learning algorithm (MAML, Finn et al., 2017). The approach used here follows B. Wang et al. (2020) which implements an objective function that explicitly optimizes for

out-of-distribution generalization in line with D. Li et al. (2018). B. Wang et al. (2020) creates pairs of tasks for each batch (which here we call meta-train and meta-test) by sub-sampling the existing training data. Each meta-train, meta-test task pair is designed to simulate the divergence between training and testing: meta-train is designed to resemble the training distribution, and meta-test to resemble the test distribution. The training objective then requires that update steps taken on meta-train are also beneficial for meta-test. This serves as a kind of regularizer, inhibiting the model from taking update steps that only benefit meta-train. By manipulating the composition of meta-test we can control the nature of the regularization applied. Unlike other meta-learning methods this is not used for few or zero-shot performance. Instead it acts as a kind of meta-augmented supervised learning, that helps the model to generalize robustly outside of its training distribution.

The approach taken by B. Wang et al. (2020) relies on the knowledge of the test setting. While it does not assume access to the test distribution, it assumes access to the family of test distributions, from which the actual test distribution will be drawn. While substantially less restrictive than the standard iid setting, it still poses a problem if we do not know the test distribution, or if the model is evaluated in a way that does not lend itself to being represented by discrete pairs of tasks (i.e. if test and train differ in a variety of distinct ways). Here we propose a more general approach that aims to generate meta-train, meta-test pairs which are populated with similar (rather than divergent) examples in an effort to inhibit the model from memorizing its input. Similarity is determined by a string or tree kernel so that for each meta-train task a corresponding meta-test task is created from examples deemed similar.

By selecting for similar examples we design the meta-test task to include examples with many of the same words as meta-train, but in novel combinations. As our training objective encourages gradient steps that are beneficial for both tasks we expect the model to be less likely to memorize large chunks which are unlikely to occur in both tasks, and therefore generalize more compositionally. This generalizes the approach from B. Wang et al. (2020), by using the meta-test task to apply a bias not-strictly related to the test distribution: the design of the meta-test task allows us to design the bias which it applies. It is worth noting that other recent approaches to this problem have leveraged data augmentation to make the training distribution more representative of the test distribution

(Andreas, 2020). We believe this line of work is orthogonal to ours as it does not focus on getting a model to generalize compositionally, but rather making the task simple enough that compositional generalization is not needed. Our method is model agnostic, and does not require prior knowledge of the target distribution.

We summarise our contributions as follows:

- We approach the problem of compositional generalization with a meta-learning objective that tries to explicitly reduce input memorization using similarity-driven virtual tasks.
- We perform experiments on two text-to-semantic compositional datasets: COGS and SCAN. Our new training objectives lead to significant improvements in accuracy over a baseline parser trained with conventional supervised learning.

6.2 Methods

We introduce the meta-learning augmented approach to supervised learning from D. Li et al. (2018) and B. Wang et al. (2020) that explicitly optimizes for out-of-distribution generalization. Central to this approach is the generation of tasks for meta-learning by sub-sampling training data. We introduce three kinds of similarity metrics used to guide the construction of these tasks.

6.2.1 Problem Definition

Compositional Generalization Kim and Linzen (eg. 2020) and Lake and Baroni (2018) introduce datasets designed to assess compositional generalization. These datasets are created by generating synthetic data with different distributions for testing and training. The differences between the distributions are trivially resolved by a compositional strategy. At their core these tasks tend to assess three key components of compositional ability: systematicity, productivity, and primitive application. Systematicity allows for the use of known parts in novel combinations as in (a). Productivity enables generalization to longer sequences than those seen in training as in (b). Primitive application allows for a word only seen in isolation during training to be applied compositionally at test time as in (c).

(a) The cat gives the dog a gift → The dog gives the cat a gift

(b) The cat gives the dog a gift → The cat gives the dog a gift and the bird a gift

(c) made → The cat made the dog a gift

A compositional grammar like the one that generated the data would be able to resolve these three kinds of generalization easily, and therefore performance on these tasks is taken as an indication of a model's compositional ability.

Conventional Supervised Learning The compositional generalization datasets we look at are semantic parsing tasks, mapping between natural language and a formal representation. A usual supervised learning objective for semantic parsing is to minimize the negative log-likelihood of the correct formal representation given a natural language input sentence, i.e. minimising

$$\mathcal{L}_{\mathcal{B}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log p_{\boldsymbol{\theta}}(y|x) \tag{6.1}$$

where N is the size of batch $\mathcal{B}$, y is a formal representation and x is a natural language sentence. This approach assumes that the training and testing data are independent and identically distributed.

Task Distributions Following from B. Wang et al. (2020), we utilize a learning algorithm that can enable a parser to benefit from a distribution of virtual tasks, denoted by $p(\tau)$, where τ refers to an instance of a virtual compositional generalization task that has its own training and test examples.

6.2.2 MAML Training

Once we have constructed our pairs of virtual tasks we need a training algorithm that encourages compositional generalization in each. Like B. Wang et al. (2020), we turn to optimization-based meta-learning algorithms Finn et al., 2017; D. Li et al., 2018 and apply DG-MAML (Domain Generalization with Model-Agnostic Meta-Learning), a variant of MAML Finn et al., 2017. Intuitively, DG-MAML encourages optimization on meta-training examples to have a positive effect on the meta-test examples as well.

During each learning episode of MAML training we randomly sample a task τ which consists of a training batch $\mathcal{B}_t$ and a generalization batch $\mathcal{B}_g$ and conduct optimization in two steps, namely *meta-train* and *meta-test*.

Algorithm 1 MAML Training Algorithm

Require: Original training set $\mathcal{T}$

Require: Learning rate α, Batch size N

for step $\leftarrow 1$ **to** T **do**
  Sample a random batch from $\mathcal{T}$ as a virtual training set $\mathcal{B}_t$
  Initialize an empty generalization set $\mathcal{B}_g$
  for $i \leftarrow 1$ **to** N **do**
    Sample an example from $\tilde{p}(\cdot \mid \mathcal{B}_t[i])$
    Add it to $\mathcal{B}_g$
  end for
  Construct a virtual task $\tau := (\mathcal{B}_t, \mathcal{B}_g)$
  Meta-train update:
    $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{B}_t}(\boldsymbol{\theta})$
  Compute meta-test objective:
    $\mathcal{L}_\tau(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{B}_t}(\boldsymbol{\theta}) + \mathcal{L}_{\mathcal{B}_g}(\boldsymbol{\theta}')$
  Final Update:
    $\boldsymbol{\theta} \leftarrow \text{Update}(\boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} \mathcal{L}_\tau(\boldsymbol{\theta}))$
end for

Meta-Train The meta-train task is sampled at random from the training data. The model performs one stochastic gradient descent step on this batch

$$\theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{B}_t}(\theta) \tag{6.2}$$

where α is the meta-train learning rate.

Meta-Test The fine-tuned parameters θ' are evaluated on the accompanying generalization task, meta-test, by computing their loss on it denoted as $\mathcal{L}_{\mathcal{B}_g}(\theta')$. The final objective for a task τ is then to jointly optimize the following:

$$\begin{aligned} \mathcal{L}_\tau(\theta) &= \mathcal{L}_{\mathcal{B}_t}(\theta) + \mathcal{L}_{\mathcal{B}_g}(\theta') \\ &= \mathcal{L}_{\mathcal{B}_t}(\theta) + \mathcal{L}_{\mathcal{B}_g}(\theta - \alpha \nabla_\theta \mathcal{L}_\beta(\theta)) \end{aligned} \tag{6.3}$$

The objective now becomes to reduce the joint loss of both the meta-train and meta-test tasks. Optimizing in this way ensures that updates on meta-train are also beneficial to meta-test. The loss on meta-test acts as a constraint on the loss from meta-train. This is unlike traditional supervised learning ($\mathcal{L}_\tau(\theta) = \mathcal{L}_{\mathcal{B}_t}(\theta) + \mathcal{L}_{\mathcal{B}_g}(\theta)$) where the loss on one batch does not constrain the loss on another.

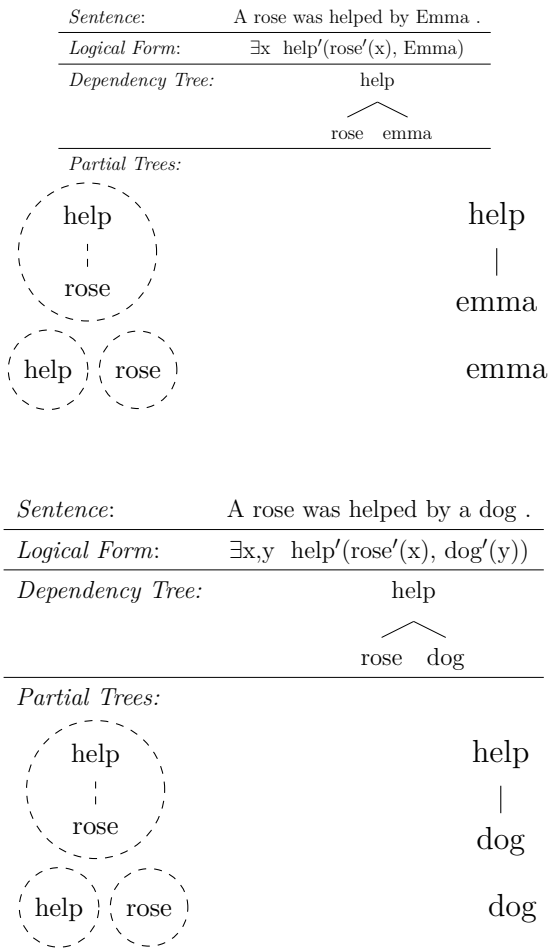


Figure 6.2: The dependency-tree forms for the logical forms of two sentences. Shown below each tree are its partial trees. As there are three partial trees shared by the examples their un-normalized tree kernel score is 3.

With a random $\mathcal{B}_t$ and $\mathcal{B}_g$, the joint loss function can be seen as a kind of generic regularizer, ensuring that update steps are not overly beneficial to meta-train alone. By constructing $\mathcal{B}_t$ and $\mathcal{B}_g$ in ways which we expect to be relevant to compositionality, we aim to allow the MAML algorithm to apply specialized regularization during training. Here we design meta-test to be similar to the meta-train task because we believe this highlights the systematicity generalization that is key to compositional ability: selecting for examples comprised of the same atoms but in different arrangements. In constraining each update step with respect to meta-train by performance on similar examples in meta-test we expect the model to dis-prefer a strategy that does not also work for meta-test like memorization of whole phrases or large sections of the input.

Source Example: The girl changed a sandwich beside the table .

Neighbours using Tree Kernel	Similarity
A sandwich changed .	0.55
The girl changed .	0.55
The block was changed by the girl .	0.39
The girl changed the cake .	0.39
change	0.32
Neighbours using String Kernel	
The girl rolled a drink beside the table .	0.35
The girl liked a dealer beside the table .	0.35
The girl cleaned a teacher beside the table .	0.35
The girl froze a bear beside the table .	0.35
The girl grew a pencil beside the table .	0.35
Neighbours using LevDistance	
The girl rolled a drink beside the table .	-2.00
The girl liked a dealer beside the table .	-2.00
The girl cleaned a teacher beside the table .	-2.00
The girl froze a bear beside the table .	-2.00
The girl grew a pencil beside the table .	-2.00

Table 6.1: Top scoring examples according to the tree kernel, string kernel and Levenshtein distance for the sentence 'The girl changed a sandwich beside the table .' and accompanying scores.

6.2.3 Similarity Metrics

Ideally, the design of virtual tasks should reflect specific generalization cases for each dataset. However, in practice this requires some prior knowledge of the distribution to which the model will be expected to generalize, which is not always available. Instead we aim to naively structure the virtual tasks to resemble each

other. To do this we use a number of similarity measures intended to help select examples which highlight the systematicity of natural language.

Inspired by kernel density estimation Parzen, 1962, we define a relevance distribution for each example:

$$\tilde{p}(x', y'|x, y) \propto \exp\Big(k([x, y], [x', y'])/\eta\Big) \tag{6.4}$$

where k is the similarity function, $[x, y]$ is a training example, η is a temperature that controls the sharpness of the distribution. Based on our extended interpretation of relevance, a high $\tilde{p}$ implies that $[x, y]$ is systematically relevant to $[x', y']$ - containing many of the same atoms but in a novel combination. We look at three similarity metrics to guide subsampling existing training data into meta-test tasks proportional to each example's $\tilde{p}$.

Levenshtein Distance First, we consider Levenshtein distance, a kind of edit distance widely used to measure the dissimilarity between strings. We compute the negative Levenshtein distance at the word-level between natural language sentences of two examples:

$$k([x, y], [x', y']) = -1 * \text{LevDistance}(x, x') \tag{6.5}$$

where LevDistance returns the number of edit operations required to transform x into x'. See Table 6.1 for examples.

Another family of similarity metrics for discrete structures are convolution kernels Haussler, 1999.

String-Kernel Similarity We use the string subsequence kernel Lodhi et al., 2002:

$$k([x, y], [x', y']) = \text{SSK}(x, x') \tag{6.6}$$

where SSK computes the number of common subsequences between natural language sentences at the word-level. See Table 6.1 for examples. [1]

Tree-Kernel Similarity In semantic parsing, the formal representation y usually has a known grammar which can be used to represent it as a tree structure. In light of this we use tree convolution kernels to compute similarity between examples: [2]

$$k([x, y], [x', y']) = \text{TreeKernel}(y, y') \tag{6.7}$$

[1] We use the normalized convolution kernels in this work, i.e., $k'(x_1, x_2) = k(x_1, x_2)/\sqrt{k(x_1, x_1)k(x_2, x_2)}$

[2] Alternatively, we can use tree edit-distance Zhang and Shasha, 1989.

where the TreeKernel function is a convolution kernel Collins and Duffy, 2001 applied to trees. Here we consider a particular case where y is represented as a dependency structure, as shown in Figure 6.2. We use the partial tree kernel (Moschitti, n.d.) which is designed for application to dependency trees. For a given dependency tree partial tree kernels generate a series of all possible partial trees: any set of one or more connected nodes. Given two trees the kernel returns the number of partial trees they have in common, interpreted as a similarity score. Compared with string-based similarity, this kernel prefers sentences that share common syntactic sub-structures, some of which are not assigned high scores in string-based similarity metrics, as shown in Table 6.1.

Though tree-structured formal representations are more informative in obtaining relevance, not all logical forms can be represented as tree structures. In SCAN Lake and Baroni, 2018 y are action sequences without given grammars. As we will show in the experiments, string-based similarity metrics have a broader scope of applications but are less effective than tree kernels in cases where y can be tree-structured.

Sampling for Meta-Test Using our kernels we compute the relevance distribution in Eq 6.4 to construct virtual tasks for MAML training. We show the resulting procedure in Algorithm 1. In order to construct a virtual task τ, a meta-train batch is first sampled at random from the training data (line 2), then the accompanying meta-test batch is created by sampling examples similar to those in meta-train (line 5).

We use *Lev-MAML, Str-MAML and Tree-MAML* to denote the meta-training using Levenshtein distance, string-kernel and tree-kernel similarity, respectively.

6.3 Experiments

6.3.1 Datasets and Splits

We evaluate our methods on the following semantic parsing benchmarks that target compositional generalization.

SCAN contains a set of natural language commands and their corresponding action sequences Lake and Baroni, 2018. We use the Maximum Compound Divergence (MCD) splits Keysers et al., 2020, which are created based on the

principle of maximizing the divergence between the compound (e.g., patterns of 2 or more action sequences) distributions of the training and test tests. We apply Lev-MAML and Str-MAML to SCAN where similarity measures are applied to the natural language commands. Tree-MAML (which uses a tree kernel) is not applied as the action sequences do not have an underlying dependency tree-structure.

COGS contains a diverse set of natural language sentences paired with logical forms based on lambda calculus Kim and Linzen, 2020. Compared with SCAN, it covers various systematic linguistic abstractions (e.g., passive to active) including examples of lexical and structural generalization, and thus better reflects the compositionality of natural language. In addition to the standard splits of Train/Dev/Test, COGS provides a generalization (Gen) set drawn from a different distribution that specifically assesses compositional generalization. We apply Lev-MAML, Str-MAML and Tree-MAML to COGS; Lev-MAML and Str-MAML make use of the natural language sentences while Tree-MAML uses the dependency structures reconstructed from the logical forms.

6.3.2 Baselines

In general, our method is model-agnostic and can be coupled with any semantic parser to improve its compositional generalization. Additionally Lev-MAML, and Str-MAML are dataset agnostic provided the dataset has a natural language input. In this work, we apply our methods on two widely used sequence-to-sequences models. [3]

LSTM-based Seq2Seq has been the backbone of many neural semantic parsers Dong and Lapata, 2016; Jia and Liang, 2016. It utilizes LSTM Hochreiter and Schmidhuber, 1997 and attention Bahdanau et al., 2014 under an encoder-decoder Sutskever et al., 2014 framework.

Transformer-based Seq2Seq also follows the encoder-decoder framework, but it uses Transformers Vaswani et al., 2017 to replace the LSTM for encoding and decoding. It has proved successful in many NLP tasks e.g., machine translation. Recently, it has been adapted for semantic parsing B. Wang et al., 2019 with superior performance.

[3] Details of implementations and hyperparameters can be found in the Appendix.

We try to see whether our MAML training can improve the compositional generalization of contemporary semantic parsers, compared with standard supervised learning. Moreover, we include a meta-baseline, referred to as Uni-MAML, that constructs meta-train and meta-test splits by uniformly sampling training examples. By comparing with this meta-baseline, we show the effect of similarity-driven construction of meta-learning splits. Note that we do not focus on making comparisons with other methods that feature specialized architectures for SCAN datasets (see Section 6.5), as these methods do not generalize well to more complex datasets Furrer et al., 2021.

GECA We additionally apply the good enough compositional augmentation (GECA) method laid out in Andreas (2020) to the SCAN MCD splits. Data augmentation of this kind tries to make the training distribution more representative of the test distribution. This approach is distinct from ours which focuses on the training objective, but the two can be combined with better overall performance as we will show. Specifically, we show the results of GECA applied to the MCD splits as well as GECA combined with our Lev-MAML variant. Note that we elect not to apply GECA to COGS, as the time and space complexity [4] of GECA proves very costly for COGS in our preliminary experiments.

6.3.3 Construction of Virtual Tasks

The similarity-driven sampling distribution $\tilde{p}$ in Eq 6.4 requires computing the similarity between every pair of training examples, which can be very expensive depending on the size of of the dataset. As the sampling distributions are fixed during training, we compute and cache them beforehand. However, they take an excess of disk space to store as essentially we need to store an $N \times N$ matrix where N is the number of training examples. To allow efficient storage and sampling, we use the following approximation. First, we found that usually each example only has a small set of neighbours that are relevant to it. [5] Motivated by this observation, we only store the top 1000 relevant neighbours for each example sorted by similarity, and use it to construct the sampling distribution denoted as $\tilde{p}_{\text{top1000}}$. To allow examples out of top 1000 being sampled, we use a linear

[4] See the original paper for details.

[5] For example, in COGS, each example only retrieves 3.6% of the whole training set as its neighbours (i.e., have non-zero tree-kernel similarity) on average.

Model	MCD1	MCD2	MCD3
LSTM	4.70 ±2.20	7.30 ±2.10	1.80 ±0.70
Transformer	0.40 ±0.40	1.80 ±0.40	0.50 ±0.10
T5-base	26.20 ±1.70	7.90 ±1.60	12.10 ±0.10
T5-11B	7.90	2.40	**16.80**
LSTM	27.40 ±8.20	31.00 ±0.40	9.60 ±3.70
w. Uni-MAML	44.80 ±5.40	31.90 ±3.40	10.00 ±1.40
w. Lev-MAML	**47.60** ±**2.30**	**35.20** ±**3.90**	11.40 ±3.00
w. Str-MAML	42.20 ±2.60	33.60 ±4.30	11.40 ±2.20
Transformer	2.60 ±0.80	3.10 ±1.00	2.30 ±1.30
w. Uni-MAML	2.80 ±0.70	3.20 ±1.00	3.20 ±1.60
w. Lev-MAML	4.70 ±1.80	6.70 ±1.40	6.50 ±1.20
w. Str-MAML	2.80 ±0.60	5.60 ±1.60	6.70 ±1.40
GECA + LSTM	51.50 ±4.40	30.40 ±4.80	12.00 ±6.80
w. Lev-MAML	**58.90** ±**6.40**	**34.50** ±**2.50**	12.30 ±4.90

Table 6.2: Main results on SCAN MCD splits. We show the mean and variance (95% confidence interval) of 10 runs. Results in the top four rows are from from Furrer et al. (2021), the remainder are results obtained in this paper.

interpolation between $\tilde{p}_{\text{top1000}}$ and a uniform distribution. Specifically, we end up using the following sampling distribution:

$$\tilde{p}(x', y'|x, y) = \lambda\ \tilde{p}_{\text{top1000}}(x', y'|x, y) + (1 - \lambda)\frac{1}{N}$$

where $\tilde{p}_{\text{top1000}}$ assigns 0 probability to out-of top 1000 examples, N is the number of training examples, and λ is a hyperparameter for interpolation. In practice, we set λ to 0.5 in all experiments. To sample from this distribution, we first decide whether the sample is in the top 1000 by sampling from a Bernoulli distribution parameterized by λ. If it is, we use $\tilde{p}_{\text{top1000}}$ to do the sampling; otherwise, we uniformly sample an example from the training set.

6.3.4 Development Set

Many tasks that assess out-of-distribution (O.O.D.) generalization (e.g. COGS) do not have an O.O.D. Dev set that is representative of the generalization distribution. This is desirable as a parser in principle should never have knowledge of the Gen set during training. In practice though the lack of an O.O.D. Dev set makes model

Model	Gen Dev	Test	Gen
LSTM	-	99.00	16.00 ± 8.00
Transformer	-	96.00	35.00 ± 6.00
LSTM	30.30 ± 7.30	99.70	34.50 ± 4.50
w. Uni-MAML	36.10 ± 6.70	99.70	36.40 ± 3.60
w. Lev-MAML	35.60 ± 5.30	99.70	36.40 ± 5.20
w. Str-MAML	36.30 ± 4.20	99.70	36.80 ± 3.50
w. Tree-MAML	**41.20** $\pm$**2.80**	99.70	**41.00** $\pm$**4.90**
Transformer	54.70 ± 4.00	99.50	58.60 ± 3.70
w. Uni-MAML	60.90 ± 2.80	99.60	64.40 ± 4.00
w. Lev-MAML	62.70 ± 3.80	99.70	64.90 ± 6.30
w. Str-MAML	62.30 ± 3.00	99.60	64.80 ± 5.50
w. Tree-MAML	**64.10** $\pm$**3.20**	99.60	**66.70** $\pm$**4.40**

Table 6.3: Main results on the COGS dataset. We show the mean and variance (standard deviation) of 10 runs. Results in the top two rows are from Kim and Linzen (2020), the remainder are results obtained in this paper.

selection extremely difficult and not reproducible. [6] In this work, we propose the following strategy to alleviate this issue: 1) we sample a small subset from the Gen set, denoted as 'Gen Dev' for tuning meta-learning hyperparmeters, 2) we use two disjoint sets of random seeds for development and testing respectively, i.e., retraining the selected models from scratch before applying them to the final test set. In this way, we make sure that our tuning is not exploiting the models resulting from specific random seeds: we do not perform random seed tuning. At no point are any of our models trained on the Gen Dev set.

6.3.5 Main Results

On SCAN, as shown in Table 6.2, Lev-MAML substantially helps both base parsers achieve better performance across three different splits constructed according to the MCD principle. [7] Though our models do not utilize pre-training such as T5 Raffel et al., 2019, our best model (Lev-MAML + LSTM) still outperforms

[6] We elaborate on this issue in the Appendix.

[7] Our base parsers also perform much better than previous methods, likely due to the choice of hyperparameters.

T5 based models significantly in MCD1 and MCD2. We show that GECA is also effective for MCD splits (especially in MCD1). More importantly, augmenting GECA with Lev-MAML further boosts the performance substantially in MCD1 and MCD2, signifying that our MAML training is complementary to GECA to some degree.

Table 6.3 shows our results on COGS. Tree-MAML boosts the performance of both LSTM and Transformer base parsers by a large margin: 6.5% and 8.1% respectively in average accuracy. Moreover, Tree-MAML is consistently better than other MAML variants, showing the effectiveness of exploiting tree structures of formal representation to construct virtual tasks. [8]

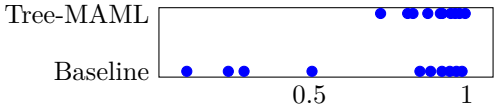
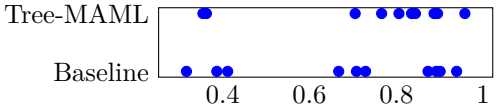
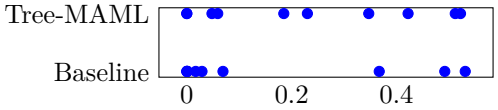
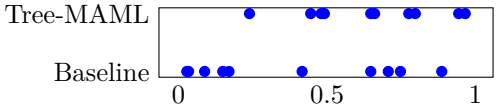
Training	Generalization	Accuracy Distribution
Primitive noun → Subject (common noun)		
shark	A **shark** examined the child.	
Primitive noun → Subject (proper noun)		
Paula	**Paula** sketched William.	
Primitive noun → Object (common noun)		
shark	A chief heard the **shark**.	
Primitive noun → Object (proper noun)		
Paula	The child helped **Paula**.	

Table 6.4: Accuracy on COGS by generalization case. Each dot represents a single run of the model.

[8] The improvement of all of our MAML variants applied to the Transformer are significant ($p < 0.03$) compared to the baseline, of our methods applied to LSTMs, Tree-MAML is significant ($p < 0.01$) compared to the baseline.

6.4 Discussion

6.4.1 SCAN Discussion

The application of our string-similarity driven meta-learning approaches to the SCAN dataset improved the performance of the LSTM baseline parser. Our results are reported on three splits of the dataset generated according to the maximum compound divergence (MCD) principle. We report results on the only MCD tasks for SCAN as these tasks explicitly focus on the systematicity of language. As such they assess a model's ability to extract sufficiently atomic concepts from its input, such that it can still recognize those concepts in a new context (i.e. as part of a different compound). To succeed here a model must learn atoms from the training data and apply them compositionally at test time. The improvement in performance our approach achieves on this task suggests that it does disincentivise the model from memorizing large sections - or entire compounds - from its input.

GECA applied to the SCAN MCD splits does improve performance of the baseline, however not to the same extent as when applied to other SCAN tasks in Andreas (2020). GECA's improvement is comparable to our meta-learning method, despite the fact that our method does not leverage any data augmentation. This means that our method achieves high performance by generalizing robustly outside of its training distribution, rather than by making its training data more representative of the test distribution. The application of our Lev-MAML approach to GECA-augmented data results in further improvements in performance, suggesting that these approaches aid the model in distinct yet complementary ways.

6.4.2 COGS Discussion

All variants of our meta-learning approach improved both the LSTM and Transformer baseline parsers' performance on the COGS dataset. The Tree-MAML method outperforms the Lev-MAML, Str-MAML, and Uni-MAML versions. The only difference between these methods is the similarity metric used, and so differences in performance must be driven by what each metric selects for. For further analysis of the metrics refer to the appendix.

The strong performance of the Uni-MAML variant highlights the usefulness of our approach generally in improving models' generalization performance. Even

without a specially designed meta-test task this approach substantially improves on the baseline Transformer model. We see this as evidence that this kind of meta-augmented supervised learning acts as a robust regularizer particularly for tasks requiring out of distribution generalization.

Although the Uni-MAML, Lev-MAML, and Str-MAML versions perform similarly overall on the COGS dataset they may select for different generalization strategies. The COGS generalization set is comprised of 21 sub-tasks which can be used to better understand the ways in which a model is generalizing (refer to Table 6.4 for examples of subtask performance). Despite having very similar overall performance Uni-MAML and Str-MAML perform distinctly on individual COGS tasks - with their performance appearing to diverge on a number of of them. This would suggest that the design of the meta-test task may have a substantive impact on the kind of generalization strategy that emerges in the model. For further analysis of COGS sub-task performance see the appendix.

Our approaches' strong results on both of these datasets suggest that it aids compositional generalization generally. However it is worth nothing that both datasets shown here are synthetic, and although COGS endeavours to be similar to natural data, the application of our methods outside of synthetic datasets is important future work.

6.5 Related Work

Compositional Generalization A large body of work on compositional generalization provide models with strong compositional bias, such as specialized neural architectures (Gordon et al., 2020; Y. Li et al., 2019; Russin et al., 2019), or grammar-based models that accommodate alignments between natural language utterances and programs (Herzig & Berant, 2020; Shaw et al., 2020). Another line of work utilizes data augmentation via fixed rules (Andreas, 2020) or a learned network (Akyürek & Andreas, n.d.) in an effort to transform the out-of-distribution compositional generalization task into an in-distribution one. Our work follows an orthogonal direction, injecting compositional bias using a specialized training algorithm. A related area of research looks at the emergence of compositional languages, often showing that languages which seem to lack natural-language like compositional structure may still be able to generalize to novel concepts (Chaabouni et al., 2020; Kottur et al., 2017). This may help to explain the ways

in which models can generalize robustly on in-distribution data unseen during training while still struggling on tasks specifically targeting compositionality.

Meta-Learning for NLP Meta-learning methods (Finn et al., 2017; Ravi & Larochelle, 2016; Vinyals et al., 2016) that are widely used for few-shot learning, have been adapted for NLP applications like machine translation (Gu et al., 2018) and relation classification (Obamuyide & Vlachos, 2019). In this work, we extend the conventional MAML (Finn et al., 2017) algorithm, which was initially proposed for few-shot learning, as a tool to inject inductive bias, inspired by D. Li et al. (2018) and B. Wang et al. (2020). For compositional generalization, Lake (2019) proposes a meta-learning procedure to train a memory-augmented neural model. However, its meta-learning algorithm is specialized for the SCAN dataset Lake and Baroni, 2018 and not suitable to more realistic datasets.

6.6 Conclusion

Our work highlights the importance of training objectives that select for robust generalization strategies. The meta-learning augmented approach to supervised learning used here allows for the specification of different constraints on learning through the design of the meta-tasks. Our similarity-driven task design improved on baseline performance on two different compositional generalization datasets, by inhibiting the model's ability to memorize large sections of its input. Importantly though the overall approach used here is model agnostic, with portions of it (Str-MAML, Lev-MAML, and Uni-MAML) proving dataset agnostic as well requiring only that the input be a natural language sentence. Our methods are simple to implement compared with other approaches to improving compositional generalization, and we look forward to their use in combination with other techniques to further improve models' compositional ability.

Chapter 7

In Conclusion

Summary and Future Work

> into the strenuous briefness [redacted]
> of yellow [redacted] coloured twilight i smilingly glide. I into the big vermilion departure swim,sayingly; (Do you think?)the i do,world is probably made of roses & hello: [redacted]
>
> *- e.e. cummings*

This brings us to an end. The past 7 chapters have discussed mappings, and their structure in general terms. Introducing an approach to understanding how they represent, abstract and preserve information, by describing each system in terms of structural primitives. By quantifying the probability of 3 basic kinds of structure we end up being able to better understand how information is structured, how that structure emerges, and what structures drive generalisation in deep learning models. We briefly review how this case was built across chapters before turning to future work.

7.1 In Summation

Chapter 1 observed that we lack sufficient tools to understand how neural networks represent information, learn, and generalise. I made a case that one way to

address this is to look at models as a member of a more general class: mappings. By drawing parallels with other mappings, like natural language, we can think about the kinds of *system-level structures* that drive their performance. Relating representation spaces in these models to other mappings across the cognitive sciences also allows us to build on existing work for what factors — like cognitive capacity — condition or constrain the structures that emerge. The remainder of the thesis builds on these basic ideas working to quantify system-level structure in mappings learned by neural networks, and studying whether model capacity has a similar regularising effect to cognitive capacity seen in related work on humans.

7.1.1 Information Structure

A major goal of this thesis was to introduce flexible, quantitative ways of thinking about structure in a mapping. Chapter 2 makes this concrete by introducing *information structure* which takes a probabilistic approach - describing a system in terms of the probability of 3 structural primitives: one-to-one, one-to-many, and many-to-one. Each of these structures in a mapping between spaces relates intuitively to basic information theoretic quantities: Mutual Information, Conditional Entropy, and the Jensen-Shannon Divergence. The remainder of the thesis puts milage on these ideas, by leveraging them to study how structure develops in discrete mappings (in chapter 3), continuous mappings learned by deep-learning models trained on a single task (chapter 4), and mappings internal to large language models (chapter 5).

Chapter 3 takes initial steps towards the theoretical framework for the thesis as a whole, looking at the discrete $\rightarrow$ discrete mappings that emerge in a multi-agent reinforcement learning model. By quantifying 4 specific kinds of variation found in natural language in terms of conditional entropies, in a model designed to have a high-level analogy to human communication, experiments here build a direct link between information structure and structures in natural language. Chapter 4 formalises the information structure framework for transformer models, defining versions of regularity, variation, and disentanglement for vector space. These experiments show how early in training representations become regular with respect to word-level information, then later in training representations contextualise, with different contextualisations for the same token become more disentangled in space. This later phase takes place after in-distribution performance saturates

and reflects the period of training where generalisation improves. At the end of training, degree of contextual disentanglement predicts which run of the model will generalise best. Additionally we show how a model's loss on the task correlates strongly with measures of structure early in training, showing the objective the model optimises for selects for specific structural properties in representational space.

Given that the approach in this thesis is built on information theory, we need methods for quickly, and accurately estimating entropy. Entropy estimation in vector spaces has long proved challenging, chapter 5 grapples with this problem, discussing limitations of existing approaches and introducing a novel method for estimating the entropy of vector space *soft entropy*. This quantity behaves like discrete entropy but is differentiable, memory efficient, and highly parallelisable - enabling us to apply an information structure analysis to representations of arbitrary size. The remainder of the chapter applies the analysis to large language models ranging in size from 14 million parameters to 12 billion. We show how the timecourse of training larger models looks similar to models trained on a single task, with later steps resulting in disentanglement of different contextualisations of the same token. We also show how larger models use proportionally more of their representation space for contextual information.

7.1.2 Capacity

Across chapters this thesis also studies the effects of capacity on representational structure. In linguistic work cognitive capacity is thought to have a regularising effect, limiting learner's ability to learn lower probability forms (Newport, 1990). More generally, humans' cognitive limitations are thought to drive the robustly generalising strategies we converge to (Griffiths, 2020b). In the multi-agent model in Chapter 3, limiting the capacity of each agent had a regularising effect on the signals they produced. When we look at model-internal representations in Chapter 4, this effect inverted with models that have larger hidden representations compressing their representation spaces more per-dimension and converging to more regular representations with respect to contextual information. The pattern of larger models being more regular with respect to contextual information holds true for the large language models studied in chapter 5.

We make sense of these contrasting results by discussing how modifying the

hidden-size of a model (the number of dimensions available to it) affects capacity, but also key parameters of the representational space. When encoding meanings in discrete signals, two parameters of the signal space affect structural properties of encodings: maximum length and the size of the alphabet used at each position in the signal. A smaller alphabet is more robust to noise but requires a longer signal - illustrated by morse code where operators only need to differentiate between two possibilities at each position, (dot or dash), but where sentences in morse are far longer than their english counterparts. In our analysis increasing the number of hidden dimensions available to the model is akin to increasing the maximum signal length, which can allow the model to compress more per-dimension, arriving at more robust representations. As a result in Chapter 3 where we measure regularity in the discrete signals between models, varying model capacity but not signal capacity shows a regularising effect. Understandably when we modify the number of hidden dimensions while also assessing regularity in hidden representations (chapters 4 and 5) we see a different pattern. The final experimental chapter (6) looks at ways to modify the capacity of models without otherwise altering properties of their representational spaces. Using a meta-learning objective to limit model's ability to memorise training data results in improved generalisation performance.

7.2 Future Work

There are three broad directions for further work that follow from this thesis, optimising for information structure, accounting for information content, and understanding the effects of a machine learning pipeline.

Optimising for Information Structure The clearest extension of the work presented here is to directly optimise for representational structure. Chapters 4 & 5 identify representational structures that correlate with improved generalisation performance. Given the soft entropy estimation method is differentiable, and fast to compute, we could directly optimise a model's representations to exhibit the structural properties we believe are desirable. This would mean adding our information structure estimates to the objective function and directly optimising for models to have higher regularity, variation, or disentanglement. While these may prove difficult to directly optimise for, and multi-term objectives can be

challenging to work with, there is real appeal to being able to provide this kind of high-level structural supervision to hidden states in neural models.

Accounting for Information In the experiments here the conditioning labels used to assess information structure were most often token, bigram, and trigram labels - because these are always available for text data. In future it would be interesting to try and account for all of the information in a model, reducing the 'residual' as much as possible. This would require data with other sets of labels to try and explain other sources of variance in representation space. As an initial direction we could look at multi-linguality, encoding text from a number of different languages, tagged with their language ID. We could then compute what proportion of representation space is devoted to language-specific information. Additionally when conditioning on language ID labels, disentanglement would tell us how separable two languages are in representational space — we could use this to see if languages that are more phylogenetically similar are more entangled in the model. Or if models which group language families together in space ultimately perform better across multi-lingual tasks.

Understanding the Machine Learning Pipeline When training a deep-learning model in 2024 there are a huge number of hyper-parameters that need to be set, largely using specific values identified as optimal by previous work. We have a limited understanding of why certain training regimes are better or worse than others. For example the AdamW optimiser consistently out performs the Adam optimiser which consistently out performs standard stochastic gradient descent. By analysing the information structure that each of these optimisers selects for we could better understand what mechanisms drive robust generalisation in a model. In the general case, identifying the design choices that seem to have the greatest effect on performance, and analysing them from an information structure perspective could give use a representational account of why these choices matter. A clear extension to the preceeding chapter would be to analyse the information structure selected for by the meta-learning objective to give a representational account of how that objective affects model behaviour.

7.3 In Closing

Mappings relate two different spaces, transforming things of one kind into another; they are ubiquitous across the sciences and the world around us. By working to understand them in the general case, we work to unify understanding from a number of different perspectives across disciplines. Doing so lets us ground new, largely inscrutable systems like neural networks, in existing work on representational structure, how it emerges, and how it evolves. Understanding artificial systems in terms of natural ones also forces us to remember that solutions to complex problems are complex in their own right. Natural language is remarkable not just for the parts of it that are simple, and predictable, but for the way it weaves complexity and simplicity, variation and regularity together.

Bibliography

Akyürek, E., & Andreas, J. (n.d.). Lexicon Learning for Few-Shot Neural Sequence Modeling, 13.

Akyürek, E., & Andreas, J. (2022). Compositionality as Lexical Symmetry [arXiv: 2201.12926]. *arXiv:2201.12926 [cs]*. Retrieved April 28, 2022, from http://arxiv.org/abs/2201.12926

Anderson, S. R. (1992). *A-Morphous Morphology*. Cambridge University Press. https://doi.org/10.1017/CBO9780511586262

Andreas, J. (2019). Measuring Compositionality in Representation Learning [arXiv: 1902.07181]. *arXiv:1902.07181 [cs, stat]*. Retrieved December 18, 2020, from http://arxiv.org/abs/1902.07181

Andreas, J. (2020). Good-Enough Compositional Data Augmentation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7556–7566. https://doi.org/10.18653/v1/2020.acl-main.676

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barry, J. (2017). Sentiment analysis of online reviews using bag-of-words and lstm approaches. *AICS*, 272–274.

Beckner, C., Blythe, R., Bybee, J., Christiansen, M. H., Croft, W., Ellis, N. C., Holland, J., Ke, J., Larsen-Freeman, D., & Schoenemann, T. (2009). Language Is a Complex Adaptive System: Position Paper. *Language Learning*, *59*(March 2007), 1–26. https://doi.org/10.1111/j.1467-9922.2009.00533.x

Beirlant, J., Dudewicz, E. J., Györfi, L., Van der Meulen, E. C., et al. (1997). Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, *6*(1), 17–39.

Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., & Glass, J. (2017, July). What do neural machine translation models learn about morphology? In R. Barzilay & M.-Y. Kan (Eds.), *Proceedings of the 55th annual meeting*

of the association for computational linguistics (volume 1: Long papers) (pp. 861–872). Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1080

Belinkov, Y., Màrquez, L., Sajjad, H., Durrani, N., Dalvi, F., & Glass, J. (2018). Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks [arXiv: 1801.07772]. *arXiv:1801.07772 [cs]*. Retrieved June 18, 2020, from http://arxiv.org/abs/1801.07772

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Bickerton, D. (1984). The language bioprogram hypothesis. *Behavioral and brain sciences*, *7*(2), 173–188.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., et al. (2023). Pythia: A suite for analyzing large language models across training and scaling. *International Conference on Machine Learning*, 2397–2430.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blevins, T., Levy, O., & Zettlemoyer, L. (2018). Deep RNNs Encode Soft Hierarchical Syntax [arXiv: 1805.04218]. *arXiv:1805.04218 [cs]*. Retrieved June 18, 2020, from http://arxiv.org/abs/1805.04218

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., et al. (2023). Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, *2*.

Brighton, H., Smith, K., & Kirby, S. (2005). Language as an evolutionary system. *Physics of Life Reviews*, *2*(3), 177–226. https://doi.org/10.1016/j.plrev.2005.06.001

Brighton, H. (2002). Compositional Syntax From Cultural Transmission. *Artificial Life*, *8*(1), 25–54. https://doi.org/10.1162/106454602753694756

Brighton, H., & Kirby, S. (2006). Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, *12*(2), 229–242.

Brown, T. B. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Cann, R. (1993). *Formal semantics an introduction* [OCLC: 1120437841]. Cambridge University Press. Retrieved June 23, 2020, from http://0-www.ebooks.cambridge.org.cataleg.uoc.edu/ebook.jsf?bid=CBO9781139166317

Carver, R. (1981). *What we talk about when we talk about love*. Vintage.

Chaabouni, R., Kharitonov, E., Bouchacourt, D., Dupoux, E., & Baroni, M. (2020). Compositionality and Generalization In Emergent Languages, 16.

Chalmers, D. J. (1993). Connectionism and compositionality: Why fodor and pylyshyn were wrong.

Chater, N., Reali, F., & Christiansen, M. H. (2009). Restrictions on biological adaptation in language evolution. *Proceedings of the National Academy of Sciences*, *106*(4), 1015–1020.

Chater, N., & Vitányi, P. (2003). Simplicity: A unifying principle in cognitive science? *Trends in Cognitive Sciences*, *7*(1), 19–22. https://doi.org/10.1016/S1364-6613(02)00005-0

Chen, R. T. Q., Li, X., Grosse, R. B., & Duvenaud, D. K. (2018). Isolating Sources of Disentanglement in Variational Autoencoders. *Advances in Neural Information Processing Systems*, *31*. Retrieved January 31, 2024, from https://proceedings.neurips.cc/paper_files/paper/2018/hash/1ee3dfcd8a0645a25a35977997223d22-Abstract.html

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. *SSST@EMNLP*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Choi, E., Lazaridou, A., & de Freitas, N. (2018). Compositional Obverter Communication Learning From Raw Visual Input [arXiv: 1804.02341]. *arXiv:1804.02341 [cs]*, 18. Retrieved January 17, 2022, from http://arxiv.org/abs/1804.02341

Chomsky, N. (1957). Logical structure in language. *Journal of the American Society for Information Science*, *8*(4), 284.

Chomsky, N. (1965). *Aspects of the theory of syntax* (50th Anniversary Edition). The MIT Press.

Chomsky, N. (1969). Quine's empirical assumptions. In *Words and objections: Essays on the work of wv quine* (pp. 53–68). Springer.

Chomsky, N. (1995). Language and nature. *Mind*, *104*(413), 1–61.

Chomsky, N. (2014). *The minimalist program*. MIT press.

Christiansen, M. H. (1994). Innite languages, finite minds connectionism, learning and linguistic structure. https://api.semanticscholar.org/CorpusID:15139134

Collins, M., & Duffy, N. (2001). Convolution Kernels for Natural Language, 8.

Conklin, H., & Smith, K. (2022). Compositionality with variation reliably emerges in neural networks. *The Eleventh International Conference on Learning Representations*.

Conklin, H., & Smith, K. (2024). Representations as language: An information-theoretic framework for interpretability. *arXiv preprint arXiv:2406.02449*.

Conklin, H., Wang, B., Smith, K., & Titov, I. (2021). Meta-Learning to Compositionally Generalize [arXiv: 2106.04252]. *arXiv:2106.04252 [cs]*. Retrieved May 4, 2022, from http://arxiv.org/abs/2106.04252

Croft, W. (2001). *Radical construction grammar: Syntactic theory in typological perspective*. Oxford University Press, USA.

Csordás, R., Irie, K., & Schmidhuber, J. (2021). The Devil is in the Detail: Simple Tricks Improve Systematic Generalization of Transformers [arXiv: 2108.12284]. *arXiv:2108.12284 [cs]*. Retrieved October 14, 2021, from http://arxiv.org/abs/2108.12284

Culbertson, J., & Kirby, S. (2016). Simplicity and Specificity in Language: Domain-General Biases Have Domain-Specific Effects. *Frontiers in Psychology*, *6*, 1964. https://doi.org/10.3389/fpsyg.2015.01964

Dagaev, N., Roads, B. D., Luo, X., Barry, D. N., Patil, K. R., & Love, B. C. (2021). A Too-Good-to-be-True Prior to Reduce Shortcut Reliance [arXiv: 2102.06406]. *arXiv:2102.06406 [cs]*. Retrieved September 23, 2021, from http://arxiv.org/abs/2102.06406

Dale, R., & Lupyan, G. (2012). UNDERSTANDING THE ORIGINS OF MORPHOLOGICAL DIVERSITY: THE LINGUISTIC NICHE HYPOTHESIS. *Advances in Complex Systems*, *15*(03n04), 1150017. https://doi.org/10.1142/S0219525911500172

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., Hormozdiari, F., Houlsby, N., Hou, S., Jerfel, G., Karthikesalingam, A., Lucic, M., Ma, Y., McLean, C., Mincu, D., ... Sculley, D. (2020). Underspecification Presents Challenges for Credibility in Modern Machine Learning [arXiv:

2011.03395]. *arXiv:2011.03395 [cs, stat]*. Retrieved January 24, 2021, from http://arxiv.org/abs/2011.03395

de Saussure, F. (1916). Course in general linguistics, 264.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [arXiv: 1810.04805]. *arXiv:1810.04805 [cs]*. Retrieved June 19, 2020, from http://arxiv.org/abs/1810.04805

Dong, L., & Lapata, M. (2016, June). Language to Logical Form with Neural Attention [arXiv:1601.01280 [cs]]. Retrieved September 10, 2022, from http://arxiv.org/abs/1601.01280

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jiang, L., Lin, B. Y., Welleck, S., West, P., Bhagavatula, C., Le Bras, R., et al. (2024). Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, *36*.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. (2022). Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*, *1*, 1.

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

Fedorenko, E. (2014). The role of domain-general cognitive control in language comprehension. *Frontiers in psychology*, *5*, 335.

Ferdinand, V., Kirby, S., & Smith, K. (2019). The cognitive roots of regularization in language. *Cognition*, *184*, 53–68. https://doi.org/10.1016/j.cognition.2018.12.002

Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks [arXiv: 1703.03400]. *arXiv:1703.03400 [cs]*, 10. Retrieved May 25, 2020, from http://arxiv.org/abs/1703.03400

Fodor, J. A. (1975). The language of thought.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*(1-2), 3–71. https://doi.org/10.1016/0010-0277(88)90031-5

Frankle, J., & Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Furrer, D., van Zee, M., Scales, N., & Schärli, N. (2021, September). Compositional Generalization in Semantic Parsing: Pre-training vs. Specialized Architectures [arXiv:2007.08970 [cs]]. Retrieved October 23, 2023, from http://arxiv.org/abs/2007.08970

Futrell, R., Wilcox, E., Morita, T., & Levy, R. (2018). Rnns as psycholinguistic subjects: Syntactic state and grammatical dependency. *arXiv preprint arXiv:1809.01329*.

Futrell, R., Wilcox, E., Morita, T., Qian, P., Ballesteros, M., & Levy, R. (2019). Neural language models as psycholinguistic subjects: Representations of syntactic state. *arXiv preprint arXiv:1903.03260*.

Goldberg, A. E. (1995). *Constructions: A construction grammar approach to argument structure*. University of Chicago Press.

Goldberg, A. E. (2003). Constructions: A new theoretical approach to language. *Trends in Cognitive Sciences*, *7*(5), 219–224. https://doi.org/10.1016/S1364-6613(03)00080-9

Goldberg, A. E. (2006). *Constructions at work: The nature of generalization in language* [OCLC: 193697889]. Oxford University Press. Retrieved June 23, 2020, from http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=3052348

Goldfarb-Tarrant, S., Marchant, R., Sanchez, R. M., Pandya, M., & Lopez, A. (2021). Intrinsic Bias Metrics Do Not Correlate with Application Bias [arXiv: 2012.15859]. *arXiv:2012.15859 [cs]*. Retrieved August 5, 2021, from http://arxiv.org/abs/2012.15859

Goldfeld, Z., Berg, E. v. d., Greenewald, K., Melnyk, I., Nguyen, N., Kingsbury, B., & Polyanskiy, Y. (2018). Estimating information flow in deep neural networks. *arXiv preprint arXiv:1810.05728*.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Goodman, N. (1955). *The new riddle of induction*. na.

Gordon, J., Lopez-Paz, D., Baroni, M., & Bouchacourt, D. (2020). PERMUTATION EQUIVARIANT MODELS FOR COMPOSITIONAL GENERALIZATION IN LANGUAGE, 12.

Griffiths, T. L. (2020a). Understanding human intelligence through human limitations. *Trends in Cognitive Sciences*, *24*, 873–883. https://api.semanticscholar.org/CorpusID:221996148

Griffiths, T. L. (2020b). Understanding Human Intelligence through Human Limitations. *Trends in Cognitive Sciences*, *24*(11), 873–883. https://doi.org/10.1016/j.tics.2020.09.001

Gu, J., Wang, Y., Chen, Y., Cho, K., & Li, V. O. (2018). Meta-learning for low-resource neural machine translation. *arXiv preprint arXiv:1808.08437*.

Guo, S., Ren, Y., Mathewson, K., Kirby, S., Albrecht, S. V., & Smith, K. (2021). Expressivity of Emergent Language is a Trade-off between Contextual Complexity and Unpredictability [arXiv: 2106.03982]. *arXiv:2106.03982 [cs]*. Retrieved October 21, 2021, from http://arxiv.org/abs/2106.03982

Hahn, M., Futrell, R., Levy, R., & Gibson, E. (2022). A resource-rational model of human processing of recursive linguistic structure. *Proceedings of the National Academy of Sciences*, *119*(43), e2122602119. https://doi.org/10.1073/pnas.2122602119

Haussler, D. (1999). *Convolution kernels on discrete structures* (tech. rep.). Technical report, Department of Computer Science, University of California ...

Havrylov, S., & Titov, I. (2017). Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols [arXiv: 1705.11192]. *arXiv:1705.11192 [cs]*. Retrieved January 21, 2020, from http://arxiv.org/abs/1705.11192

Herzig, J., & Berant, J. (2020). Span-based semantic parsing for compositional generalization. *arXiv preprint arXiv:2009.06040*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hockett, C. F. (1960). The Origin of Speech. *Scientific American*, *203*(3), 88–97. https://doi.org/10.2307/24940617

Hu, J., Gauthier, J., Qian, P., Wilcox, E., & Levy, R. P. (2020). A Systematic Assessment of Syntactic Generalization in Neural Language Models [arXiv: 2005.03692]. *arXiv:2005.03692 [cs]*. Retrieved June 23, 2020, from http://arxiv.org/abs/2005.03692

Hudson Kam, C. L., & Chang, A. (2009). Investigating the cause of language regularization in adults: Memory constraints or learning effects? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *35*(3), 815–821. https://doi.org/10.1037/a0015097

Hudson Kam, C. L., & Newport, E. L. (2005). Regularizing Unpredictable Variation: The Roles of Adult and Child Learners in Language Formation and Change. *Language Learning and Development*, *1*(2), 151–195. https://doi.org/10.1080/15475441.2005.9684215

Hupkes, D., Dankers, V., Mul, M., & Bruni, E. (2019). The compositionality of neural networks: Integrating symbolism and connectionism [arXiv: 1908.08351]. *arXiv:1908.08351 [cs, stat]*. Retrieved January 13, 2020, from http://arxiv.org/abs/1908.08351

Hupkes, D., Veldhoen, S., & Zuidema, W. (2018). Visualisation and 'Diagnostic Classifiers' Reveal How Recurrent and Recursive Neural Networks Process Hierarchical Structure. *Journal of Artificial Intelligence Research*, *61*, 907–926. https://doi.org/10.1613/jair.1.11196

Hurford, J. R. (2003). Why synonymy is rare: Fitness is in the speaker. *European conference on artificial life*, 442–451.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, *106*, 620–630. https://api.semanticscholar.org/CorpusID:17870175

Jia, R., & Liang, P. (2016). Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*.

Jiang, Z., Zhang, C., Talwar, K., & Mozer, M. C. (2020). Characterizing Structural Regularities of Labeled Data in Overparameterized Models [arXiv: 2002.03206]. *arXiv:2002.03206 [cs, stat]*. Retrieved January 25, 2021, from http://arxiv.org/abs/2002.03206

Kalchbrenner, N., & Blunsom, P. (2013). Recurrent continuous translation models. *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1700–1709.

Kemp, C., Xu, Y., & Regier, T. (2018). Semantic typology and efficient communication. *Annual Review of Linguistics*, *4*(1), 109–128.

Kennedy, B., Tyson, A., & Saks, E. (2023). Public awareness of artificial intelligence in everyday activities.

Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., Momchev, N., Sinopalnikov, D., Stafiniak, L., Tihon, T., Tsarkov, D., Wang,

X., van Zee, M., & Bousquet, O. (2020). Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. *arXiv preprint arXiv:1912.09713*, 38. https://openreview.net/forum?id=SygcCnNKwr

Kharitonov, E., & Baroni, M. (2020). Emergent language generalization and acquisition speed are not tied to compositionality [arXiv: 2004.03420]. *arXiv:2004.03420 [cs]*. Retrieved January 29, 2022, from http://arxiv.org/abs/2004.03420

Kharitonov, E., Chaabouni, R., Bouchacourt, D., & Baroni, M. (2019). EGG: A toolkit for research on Emergence of lanGuage in Games [arXiv: 1907.00852]. *arXiv:1907.00852 [cs]*. Retrieved January 14, 2022, from http://arxiv.org/abs/1907.00852

Kim, N., & Linzen, T. (2020). COGS: A Compositional Generalization Challenge Based on Semantic Interpretation [arXiv: 2010.05465]. *arXiv:2010.05465 [cs]*, 9087–9105. Retrieved January 25, 2021, from http://arxiv.org/abs/2010.05465

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirby, S. (2001). Spontaneous evolution of linguistic structure-an iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, *5*(2), 102–110. https://doi.org/10.1109/4235.918430

Kirby, S., Cornish, H., & Smith, K. (2008). Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences*, *105*(31), 10681–10686. https://doi.org/10.1073/pnas.0707835105

Kirby, S., Griffiths, T., & Smith, K. (2014). Iterated learning and the evolution of language. *Current Opinion in Neurobiology*, *28*, 108–114. https://doi.org/10.1016/j.conb.2014.07.014

Kirby, S., Tamariz, M., Cornish, H., & Smith, K. (2015). Compression and communication in the cultural evolution of linguistic structure. *Cognition*, *141*, 87–102. https://doi.org/10.1016/j.cognition.2015.03.016

Kottur, S., Moura, J. M. F., Lee, S., & Batra, D. (2017). Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog [arXiv: 1706.08502]. *arXiv:1706.08502 [cs]*, 2962–2967. https://doi.org/10.18653/v1/D17-1321

Laka Mugarza, I. (1996). *A brief grammar of euskara, the basque language*. Universidad del País Vasco, Euskal Herriko Unibertsitatea, Euskarazko ...

Lake, B. M. (2019). Compositional generalization through meta sequence-to-sequence learning. *arXiv preprint arXiv:1906.05381*, 12.

Lake, B. M., & Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks [arXiv: 1711.00350]. *arXiv:1711.00350 [cs]*, 10. Retrieved June 19, 2020, from http://arxiv.org/abs/1711.00350

Lashley, K. S. (1951). *The problem of serial order in behavior* (Vol. 21). Bobbs-Merrill Oxford.

Lazaridou, A., & Baroni, M. (2020). Emergent Multi-Agent Communication in the Deep Learning Era [arXiv: 2006.02419]. *arXiv:2006.02419 [cs]*. Retrieved January 17, 2022, from http://arxiv.org/abs/2006.02419

Lazaridou, A., Hermann, K. M., Tuyls, K., & Clark, S. (2018). Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input [arXiv: 1804.03984]. *arXiv:1804.03984 [cs]*. Retrieved January 17, 2022, from http://arxiv.org/abs/1804.03984

Lazaridou, A., Peysakhovich, A., & Baroni, M. (2017). MULTI-AGENT COOPERATION AND THE EMERGENCE OF (NATURAL) LANGUAGE, 11.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541–551.

Levina, E., & Bickel, P. (2004). Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, *17*.

Lewis, D. (1970). *Convention: A philosophical study*. John Wiley & Sons.

Lezon, T. R., Banavar, J. R., Cieplak, M., Maritan, A., & Fedoroff, N. V. (2006). Using the principle of entropy maximization to infer genetic interaction networks from gene expression patterns. *Proceedings of the National Academy of Sciences*, *103*(50), 19033–19038.

Li, B., Donatelli, L., Koller, A., Linzen, T., Yao, Y., & Kim, N. (2023, October). SLOG: A Structural Generalization Benchmark for Semantic Parsing [arXiv:2310.15040 [cs]]. Retrieved January 28, 2024, from http://arxiv.org/abs/2310.15040

Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. (2018). Learning to generalize: Meta-learning for domain generalization. *Proceedings of the AAAI conference on artificial intelligence*, *32*(1).

Li, Y., Zhao, L., Wang, J., & Hestness, J. (2019). Compositional generalization for primitive substitutions. *arXiv preprint arXiv:1910.02612*.

Lieder, F., & Griffiths, T. L. (2020). Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, *43*, e1. https://doi.org/10.1017/S0140525X1900061X

Liu, Y. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., & Bachem, O. (2019). Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations [ISSN: 2640-3498]. *Proceedings of the 36th International Conference on Machine Learning*, 4114–4124. Retrieved January 31, 2024, from https://proceedings.mlr.press/v97/locatello19a.html

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, *2*(Feb), 419–444.

Lu, Y., Liu, Z., Baratin, A., Laroche, R., Courville, A., & Sordoni, A. (2022, June). Expressiveness and Learnability: A Unifying View for Evaluating Self-Supervised Learning [arXiv:2206.01251 [cs]]. Retrieved February 14, 2023, from http://arxiv.org/abs/2206.01251

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Lupyan, G., & Dale, R. (2010). Language structure is partly determined by social structure. *PloS one*, *5*(1), e8559.

MacKay, D. J. C. (2004). Information theory, inference, and learning algorithms. *IEEE Transactions on Information Theory*, *50*, 2544–2545.

Martins, P. T., & Boeckx, C. (2019). Language evolution and complexity considerations: The no half-merge fallacy. *PLoS biology*, *17*(11), e3000389.

Marvin, R., & Linzen, T. (2018). Targeted Syntactic Evaluation of Language Models [arXiv: 1808.09031]. *arXiv:1808.09031 [cs]*, 1192–1202. https://doi.org/10.18653/v1/D18-1151

McCoy, R. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

Merrill, W., Tsilivis, N., & Shukla, A. (2023). A tale of two circuits: Grokking as competition of sparse and dense subnetworks. *arXiv preprint arXiv:2303.11873*.

Mikolov, T. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, G. (1955). Note on the bias of information estimates. *Information theory in psychology: Problems and methods*.

Miller, G. A. (1951). Language and communication.

Mordatch, I., & Abbeel, P. (2018). Emergence of Grounded Compositional Language in Multi-Agent Populations [arXiv: 1703.04908]. *arXiv:1703.04908 [cs]*. https://doi.org/10.1007/BF00341314

Moschitti, A. (n.d.). Making Tree Kernels practical for Natural Language Learning, 8.

Müller-Eberstein, M., Van Der Goot, R., Plank, B., & Titov, I. (2023). Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. *arXiv preprint arXiv:2310.16484*.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., & Sutskever, I. (2021). Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, *2021*(12), 124003.

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science*, *14*(1), 11–28. https://doi.org/https://doi.org/10.1016/0364-0213(90)90024-Q

Nowak, M. A., Plotkin, J. B., & Jansen, V. A. A. (2000). The evolution of syntactic communication. *Nature*, *404*(6777), 495–498. https://doi.org/10.1038/35006635

Obamuyide, A., & Vlachos, A. (2019). Model-agnostic meta-learning for relation classification with limited supervision.

O'Donnell, T. J. (2015). *Productivity and Reuse in Language: A Theory of Linguistic Computation and Storage*. The MIT Press. https://doi.org/10.7551/mitpress/9780262028844.001.0001

Paninski, L. (2003). Estimation of Entropy and Mutual Information. *Neural Computation*, *15*(6), 1191–1253. https://doi.org/10.1162/089976603321780272

Partee, B., et al. (1995). Lexical semantics and compositionality. *An invitation to cognitive science: Language*, *1*, 311–360.

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, *33*(3), 1065–1076.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Piantadosi, S. T., Tily, H., & Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition*, *122*(3), 280–291. https://doi.org/10.1016/j.cognition.2011.10.004

Pimentel, T., Valvoda, J., Maudslay, R. H., Zmigrod, R., Williams, A., & Cotterell, R. (2020). Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., & Misra, V. (2022). Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, *21*(140), 1–67.

Ravi, S., & Larochelle, H. (2016). Optimization as a model for few-shot learning.

Reali, F., & Griffiths, T. L. (2009). The evolution of frequency distributions: Relating regularization to inductive biases through iterated learning. *Cognition*, *111*(3), 317–328. https://doi.org/10.1016/j.cognition.2009.02.012

Resnick, C., Gupta, A., Foerster, J., Dai, A. M., & Cho, K. (2020). Capacity, Bandwidth, and Compositionality in Emergent Language Learning [arXiv: 1910.11424]. *arXiv:1910.11424 [cs, stat]*. Retrieved January 29, 2022, from http://arxiv.org/abs/1910.11424

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Russell, B. (1912). *The problems of philosophy*. OUP Oxford.

Russin, J., Jo, J., O'Reilly, R. C., & Bengio, Y. (2019). Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv preprint arXiv:1904.09708*.

Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., & Gelly, S. (2018). Assessing generative models via precision and recall. *Advances in neural information processing systems*, *31*.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., & Cox, D. D. (2019). On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, *2019*(12), 124020.

Schneider, T. D. (2010). A brief review of molecular information theory. *Nano communication networks*, *1*(3), 173–180.

Sellam, T., Yadlowsky, S., Tenney, I., Wei, J., Saphra, N., D'Amour, A., Linzen, T., Bastings, J., Turc, I., Eisenstein, J., Das, D., & Pavlick, E. (2022). THE MULTIBERTS: BERT REPRODUCTIONS FOR ROBUSTNESS ANALYSIS. *arXiv preprint arXiv:2106.16163*.

Senghas, A., Coppola, M., Newport, E. L., & Supalla, T. (1997). Argument structure in nicaraguan sign language: The emergence of grammatical devices. *Proceedings of the Boston university conference on language development*, *21*(2), 550–61.

Shannon, C. (1948). A mathematical theory of communication. *Bell Syst. Tech. J.*, *27*, 623–656. https://api.semanticscholar.org/CorpusID:55379485

Shannon, C. (1949). Communication in the Presence of Noise. *Proceedings of the IRE*, *37*(1), 10–21. https://doi.org/10.1109/JRPROC.1949.232969

Shaw, P., Chang, M.-W., Pasupat, P., & Toutanova, K. (2020). Compositional Generalization and Natural Language Variation: Can a Semantic Parsing Approach Handle Both? [arXiv: 2010.12725]. *arXiv:2010.12725 [cs]*. Retrieved December 18, 2020, from http://arxiv.org/abs/2010.12725

Shi, X., Padhi, I., & Knight, K. (2016, November). Does string-based neural MT learn source syntax? In J. Su, K. Duh, & X. Carreras (Eds.), *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 1526–1534). Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1159

Shwartz-Ziv, R., & Tishby, N. (2017, April). Opening the Black Box of Deep Neural Networks via Information [arXiv:1703.00810 [cs]]. Retrieved January 29, 2023, from http://arxiv.org/abs/1703.00810

Smith, K. (2011). Learning Bias, Cultural Evolution of Language, and the Biological Evolution of the Language Faculty. *Human Biology*, *83*(2), 261–278. https://doi.org/10.3378/027.083.0207

Smith, K., Brighton, H., & Kirby, S. (2003). Complex systems in language evolution: The cultural emergence of compositional structure. *Advances in complex systems*, *6*(04), 537–558.

Smith, K., & Kirby, S. (2008). Cultural evolution: Implications for understanding the human language faculty and its evolution. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *363*(1509), 3591–3603. https://doi.org/10.1098/rstb.2008.0145

Smith, K., & Wonnacott, E. (2010). Eliminating unpredictable variation through iterated learning. *Cognition*, *116*(3), 444–449. https://doi.org/10.1016/j.cognition.2010.06.004

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*(1-2), 159–216. https://doi.org/10.1016/0004-3702(90)90007-M

Stevenson, A. (2010). *Oxford dictionary of english*. Oxford University Press, USA.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Symons, J., & Calvo, P. (2014). Systematicity: An overview.

Tishby, N., & Zaslavsky, N. (2015, March). Deep Learning and the Information Bottleneck Principle [arXiv:1503.02406 [cs]]. *IEEE*. Retrieved January 29, 2023, from http://arxiv.org/abs/1503.02406

Tomasello, M. (2005). *Constructing a language: A usage-based theory of language acquisition*. Harvard university press.

Tucker, M., Eisape, T., Qian, P., Levy, R., & Shah, J. (2022, April). When Does Syntax Mediate Neural Language Model Performance? Evidence from Dropout Probes [arXiv:2204.09722 [cs]]. Retrieved August 19, 2022, from http://arxiv.org/abs/2204.09722

Turc, I., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in neural information processing systems*, *30*, 11.

Veldhoen, S., Hupkes, D., & Zuidema, W. (2016). Diagnostic classifiers: Revealing how neural networks process hierarchical structure, 10.

Vinga, S. (2014). Information theory applications for biological sequence analysis. *Briefings in bioinformatics*, *15*(3), 376–389.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. *Advances in neural information processing systems*, 3630–3638.

Voita, E., Sennrich, R., & Titov, I. (2021, September). Language Modeling, Lexical Translation, Reordering: The Training Process of NMT through the Lens of Classical SMT [arXiv:2109.01396 [cs]]. Retrieved October 30, 2023, from http://arxiv.org/abs/2109.01396

Voita, E., & Titov, I. (2020, March). Information-Theoretic Probing with Minimum Description Length [arXiv:2003.12298 [cs]]. Retrieved November 21, 2023, from http://arxiv.org/abs/2003.12298

Von Humboldt, W. (1863). *Humboldt:'on language': On the diversity of human language construction and its influence on the mental development of the human species*. Cambridge University Press.

Wang, A. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wang, B., Lapata, M., & Titov, I. (2020). Meta-learning for domain generalization in semantic parsing. *arXiv preprint arXiv:2010.11988*.

Wang, B., Shin, R., Liu, X., Polozov, O., & Richardson, M. (2019). Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. *arXiv preprint arXiv:1911.04942*.

Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., & Bowman, S. R. (2019). BLiMP: A Benchmark of Linguistic Minimal Pairs for English [arXiv: 1912.00582]. *arXiv:1912.00582 [cs]*. Retrieved January 13, 2020, from http://arxiv.org/abs/1912.00582

Wehbe, L., Blank, I. A., Shain, C., Futrell, R., Levy, R., von der Malsburg, T., Smith, N., Gibson, E., & Fedorenko, E. (2021). Incremental language comprehension difficulty predicts activity in the language network but not the multiple demand network. *Cerebral Cortex*, *31*(9), 4006–4023.

Weinreich, U., Labov, W., & Herzog, M. (1968). Empirical foundations for a theory of language change. *University of Texas Press*, 100.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3), 229–256.

Winters, J., Kirby, S., & Smith, K. (2018). Contextual predictability shapes signal autonomy. *Cognition*, *176*, 15–30.

Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, *8*, 58443–58469.

Zaslavsky, N., Kemp, C., Regier, T., & Tishby, N. (2018). Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, *115*(31), 7937–7942. https://doi.org/10.1073/pnas.1800521115

Zech, J. R., Badgeley, M. A., Liu, M., Costa, A. B., Titano, J. J., & Oermann, E. K. (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS medicine*, *15*(11), e1002683.

Zhang, K., & Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, *18*(6), 1245–1262.

Zuidema, W. (2002). How the poverty of the stimulus solves the poverty of the stimulus. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems* (Vol. 15). MIT Press. https://proceedings.neurips.cc/paper_files/paper/2002/file/04ad5632029cbfbed8e136e5f6f7ddfa-Paper.pdf

Appendix A

What We Talk About When We Talk about Compositionality

A.1 Full Formalisations

For readability body text includes simple definitions for each measure, without aggregation to the language level. Below are more detailed equations that include aggreagation steps:

$$Synonymy(\mathcal{L}) = \frac{1}{|R|}\sum_{r=1}^{|R|}\frac{1}{|A_r|}\sum_{i=1}^{|A_r|}\frac{min\Big[\mathcal{H}(char_p|atom_{r,i}) : \forall p \in P\Big]}{\log(n_{chars})} \tag{A.1}$$

$$Homonomy(\mathcal{L}) = \frac{1}{|R|}\sum_{r=1}^{|R|}\frac{1}{|C|}\sum_{j=1}^{|C|}\frac{min\Big[\mathcal{H}(atom_r|char_{p,j}) : \forall p \in P\Big]}{\log(n_{atoms})} \tag{A.2}$$

A.2 Rolling Mixed Effects Model Implementation

We implement a rolling mixed effects model using the python statsmodels package. We fit a separate model for each of our 6 independent variables: synonymy, entanglement, freedom, homonymy, topsim, posids. The dependent variable across all of them is o.o.d. generalization performance. For each model we include two random effects, random intercepts based on the seed used to initialize that run of the model, and random slopes for the epochs of training. This allows the model to account for variation between different models, given that some initializations outperform others.

The model is fit to a window of 100 epochs of training data at a time, at each step it fits a regression to 100 epochs of data for 20 initializations of the model. It then moves forward one epoch at a time (e.g. the first fit of the model is on epochs 0-100, the second on 1-101, the third 2-102, etc.). We plot the resulting regression coefficient (b value) obtained from each mixed effects model fit to each IV, at each window of data.

For reference the corresponding command to run this model with the LMER package in R (a standard method for fitting these kinds of models) is: lmer(ood_acc $\sim$ IV + (1 + Epoch | Seed)) where IV is one of the variation measures.

A.3 O.O.D. Accuracy Vs. Variation Slices

In addition to the regression analysis presented in the results section we show relational plots for two different epochs in training: one from mid way through and one from late in training near convergence. In line with the regression analysis a more linear relationship between o.o.d. performance and variation is visible earlier in training before the language becomes regular enough for the task. Entanglement in particular shows a steep negative relationship in the 100 epoch plot but is totally scattered by epoch 500. Were we to only assess the relationship between generalization and variation at the end of training we would could easily conclude in line with previous work that they were not meaningfully related.

It's worth noting that the pattern here may not appear as salient as it appears in the rolling mixed effects model presented earlier, there are two major reasons for this: first the rolling model considers 100 epochs at a time, rather than a single slice with only 20 data points, providing it with 100 times the data visualized here by which to assess the relationship between variation and generalization. Secondly the rolling model has a random intercept based on the seed used in each run of the model. This is important because in line with other work on o.o.d. generalization we see a substantial effect of initialization on generalization performance, by including it as a random effect the rolling model can look at each seed separately to see if each seed's generalization performance over the run is related to its language's variation. So while in the visualizations below we may see some seeds which appear like outliers, the rolling model accounts for this, fitting a separate intercept for each run.

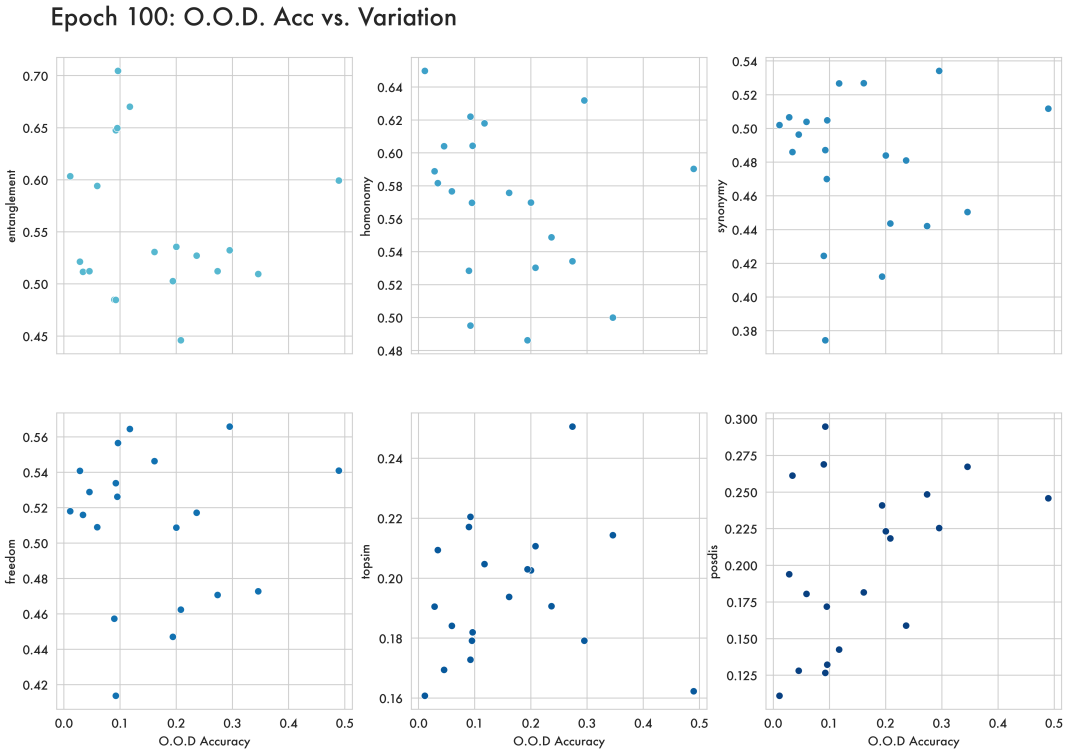


Figure A.1: Plots show each of the variation measures plotted against o.o.d. accuracy at the 100th epoch of training.

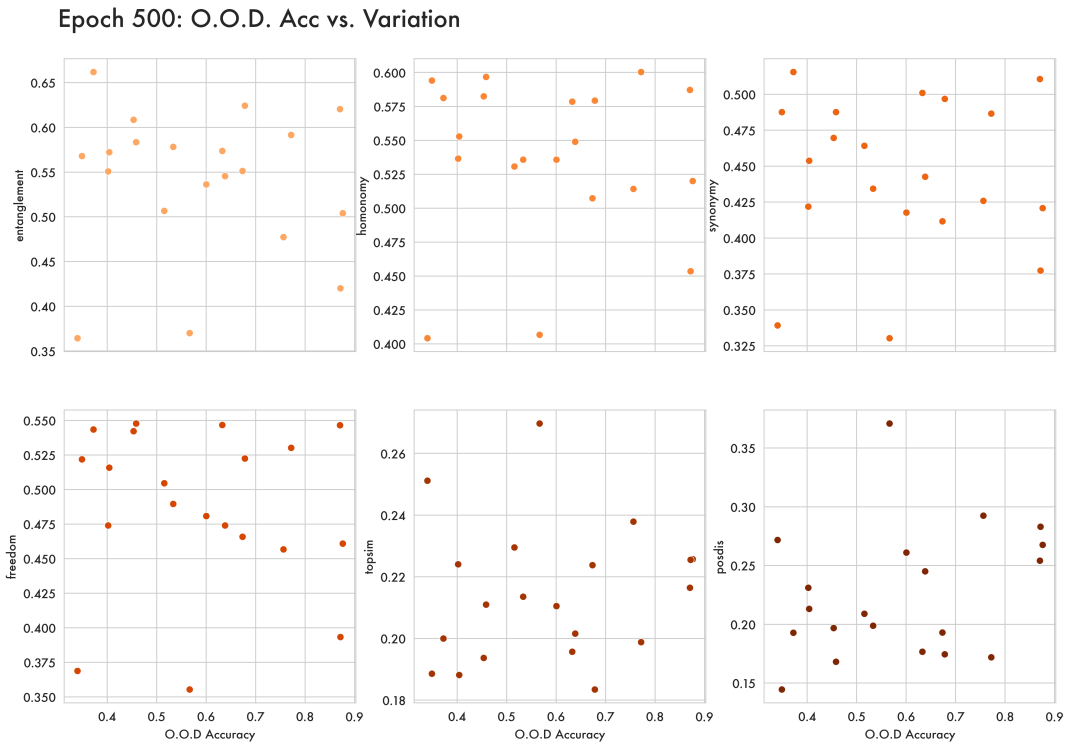


Figure A.2: Plots show each of the variation measures plotted against o.o.d. accuracy at the 500th epoch of training.

A.4 O.O.D. Accuracy Vs. Variation Discussion

As noted in section 3.3 we see a strong relationship between regularity and o.o.d. performance early in training but this effect goes away as the model converges. We attribute this to each run of the model decreasing the degree of variation in its language over time, resulting in a language sufficiently regular to succeed at the task. Where because all languages are sufficiently regular, whether one is slightly more regular than another doesn't necessarily result in better generalization performance. This dovetails with the overarching argument here that as seen in natural language even a high-degree of variation doesn't necessarily undermine a language's ability to generalize.

However it is worth noting that the relationship between regularity and generalization early on could be driven in part by more regular languages being easier for the listener to learn. Chaabouni et al. (2020) observes that higher topsim languages are easier to acquire. By taking emergent languages from the end of training, and separately training a model to map between signals and meanings

using supervised-learning they show higher topsim languages require less training to converge. Here it could be the case that early in training more regular languages are easier for the listener to learn, improving generalization performance early on and explaining in part the early correlation between regularity and generalisation. However, this is an emergent model and languages are not static throughout training meaning what the listener tries to learn is a 'moving-target' changing at each step. For this reason framing emergent results in terms of results that look at the learnability of static languages would potentially seem to draw a false equivalency. It's unclear if the learnability of a language at timestep n matters at $n+1$ when the language has changed. Given this, and the fact that we see all conditions decrease variation in the emergent language over time, we believe the best interpretation of our results is that regularity matters for generalization until the language becomes sufficiently regular for the task. Once sufficient regularity is reached greater regularity doesn't necessarily improve generalization performance so we see no correlation. Although it is possible learnability has some effect - further study of the role learnability plays in an emergent context, where what is learned changes, is needed to understand the full picture.

A.5 Use of Equation 1 for all 4 measures of variation

We use equation 3.1 (copied below in simplified notation) to calculate the conditional probabilities used in all 4 measures of variation:

$$\mathbb{P}(char|position, atom, role) = \frac{count(char, position, atom, role)}{count(position, atom, role)} \tag{A.3}$$

This gives us a distribution over characters for each position, for each atom, in each role. This distribution is intuitively useful for estimating synonymy which can be seen as the entropy over characters in a position.

A.5.1 Freedom and Entanglement

However it's natural to wonder why we use this distribution again when calculating measures of word order freedom and entanglement. Both of these measures refer to how likely it is that a given role is encoded in a position in the signal. Freedom looks at how consistently the atoms in a role are encoded in a position, while

entanglement looks at how consistently any two roles are encoded in the same position. Intuitively we might want to calculate a distribution over positions given roles instead, like:

$$\mathbb{P}(position, char, atom|role) = \frac{count(position, char, atom, role)}{count(role)} \tag{A.4}$$

then marginalize over characters and atoms so that we could directly estimate the probability of a position given a role $\mathbb{P}(position|role)$. The problem with this is that every signal has a character in every position, and every meaning has more than one role (i.e. subject, verb, and object, rather than just having a subject) meaning that the distribution over positions is always uniform. If we were to only marginalize over atoms, to get a distribution over characters and positions $\mathbb{P}(position, char|role)$ this is also nearly uniform, because different atoms are encoded using different characters. So marginalizing over atoms combines distinct distributions for each atom into a near-uniform one. Similarly if we only marginalize over characters to get a distribution over positions and atoms $\mathbb{P}(position, atoms|role)$ because every signal has a character in every position the resulting distribution is also uniform.

Fundamentally the only relevant probability distribution which is consistently non-uniform is the one described in equation 1: $\mathbb{P}(char|position, atom, role)$. Even though every signal has a character in every position every character is not equally likely given a specific $position, atom, role$ combination. As a result this is the distribution we use to calculate measures of word order freedom and entanglement. In order to do so we first observe that in signal positions where an $atom, role$ combination is not encoded $\mathbb{P}(char|position, atom, role)$ is uniform as the distribution is not conditioned by the selected atom and role. Accordingly we take lower conditional entropy $\mathcal{H}(\mathbb{P}(char|position, atom, role))$ (used in equation 3.2) as an indication that an $atom, role$ combination is more likely to be encoded in a position. By taking a mean of this conditional entropy across all atoms in a given role (described in equation 3.4a) we can see if it is consistently low in the same position(s) of the signal for all atoms in that role - indicating adherence to a single word order. Equation 3.4b then aggregates this across all roles.

Entanglement looks to see if there is consistent encoding of multiple roles in a single position. Seeing as we take low conditional entropy as an indication that an $atom, role$ combination is encoded in a given position we compare the mean

from equation 3.4a with means from other roles to see if they are consistently low in the same parts of the signal.

A.5.2 Homonymy

Given that homonymy assesses the probability that a letter in a position encodes each atom in a role, it is possible to look at this by estimating the distribution $\mathbb{P}(atom|char, position, role)$ directly. The same distribution can be calculated by instead taking the $\mathbb{P}(char|position, atom, role)$ distribution and re-normalizing it along the atom axis:

$$\mathbb{H}(char_{p,j}, r) = \frac{\left\{\mathbb{P}(char_{p,j}|atom_{r,i}) : \forall i \in A_r\right\}}{\sum_{i=1}^{|A_r|} \mathbb{P}(char_{p,j}|atom_{r,i})} \tag{A.5}$$

We find empirically that this is equivalent to computing $\mathcal{H}(\mathbb{P}(atom|char, position, role))$ (see results in table A.1) with the only differences between the two resulting from small rounding errors. In table A.1 we report results for both approaches to computing homonomy across model sizes to show their equivalency. Note these results are the means of 6 seeds so differ slightly from figures in the core results. When introducing the measures in section 2.3 of these two approaches we opt for the re-normalization of $\mathbb{P}(char|position, atom, role)$ rather than computing a new probability distribution because we believe this makes the formulation of the homonymy measure more intuitively related to the others, and makes the visualizations in figure 3.1 a direct reflection of how the measures are computed while producing equivalent results.

epoch	ideal	random	small	medium	large
homonymy	0.12	0.99	0.56 ± 0.14	0.62 ± 0.15	0.72 ± 0.05
direct homonymy	0.12	0.99	0.56 ± 0.14	0.62 ± 0.15	0.72 ± 0.05

Table A.1: Homonymy refers to the method of computing homonymy used in the core results and described in equation A.5 while direct homonymy instead directly estimates the distribution $\mathbb{P}(atom|char, position, role)$. Results are the mean of 6 initializations at the best generalizing epoch, so values differ slightly from those in the main results which are the mean of 20.

A.6 Residual Entropy

In addition to topsim and posdis reported in the main results we also report results from one other measure from previous work, residual entropy (Resnick et al., 2020). The results here follow the same pattern as the other measures of variation with larger models arriving and more irregular languages. Additionally all conditions increase the regularity of the emergent language over the course of training. Also shown is the correlation analysis between Residual entropy and O.O.D. performance, showing like other measures of variation residual entropy is a strong predictor early in training but that this effect goes away later on.

epoch	ideal	random	small	medium	large
best	0.0610	0.6250	0.2990±0.16	0.3780±0.12	0.4650±0.08
Δ *o.o.d.*			0.5230±0.05	0.4460±0.08	0.2370±0.20

Table A.2: Residual entropy scores at the best generalizing epoch and the difference between the best generalizing epoch and one drawn from early in training. Results are the mean of 6 initializations.

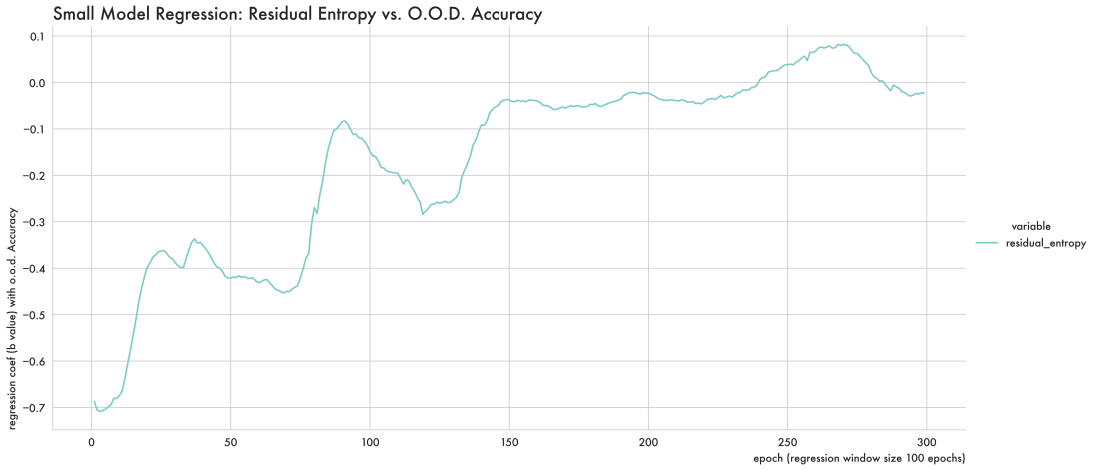


Figure A.3: The rolling mixed effects model coefficients between Residual Entropy and o.o.d. generalization accuracy for the $small$ model for each window.

A.7 I.I.D. Correlation Results

We also include the correlation results between the measures of variation and in-distribution generalization. The results follow a similar pattern with degree of

regularity being a strong predictor of generalization performance early in training but this effect goes away as the emergent language becomes regular enough to generalize well. Interestingly in-distribution and out-of-distribution correlations align almost exactly. This is reassuring in that it shows degree of regularity is important for generalization in general whether it is in or out of distribution.

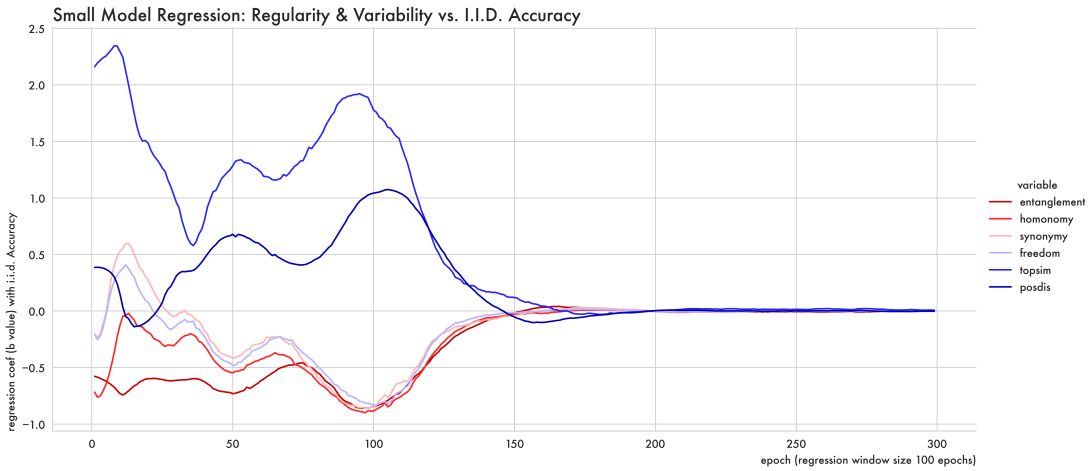


Figure A.4: The model is fit to a window of data from 100 epochs at a time across 20 initializations. The window slides forward one epoch at a time (i.e. epochs 0-100, 1-101 ...) and fits a different model between i.i.d. accuracy and each measure of variation for each window. Shown are the regression coefficients (b values) of our four measures of variation, and two previous measures of regularity (topsim and posdis) with o.o.d. generalization accuracy for the $small$ model for each window.

A.8 Significance Testing for Variation Differences

params	synonymy	entanglement	freedom	homonomy	topsim	posdis
250 vs 500	0.0275	0.1049	0.0574	0.0629	0.2565	0.0993
250 vs 800	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
500 vs 800	0.0001	< 0.0001	0.0002	0.0001	0.0005	0.0001

Table A.3: P Values obtained from a t-test comparing variation measures from different sized initialization. The difference between $large$ and $small$, and $large$ and $medium$ are significant. Of differences between $small$ and $medium$ only synonymy and posdis are significant

A.8.1 Hyperparameters

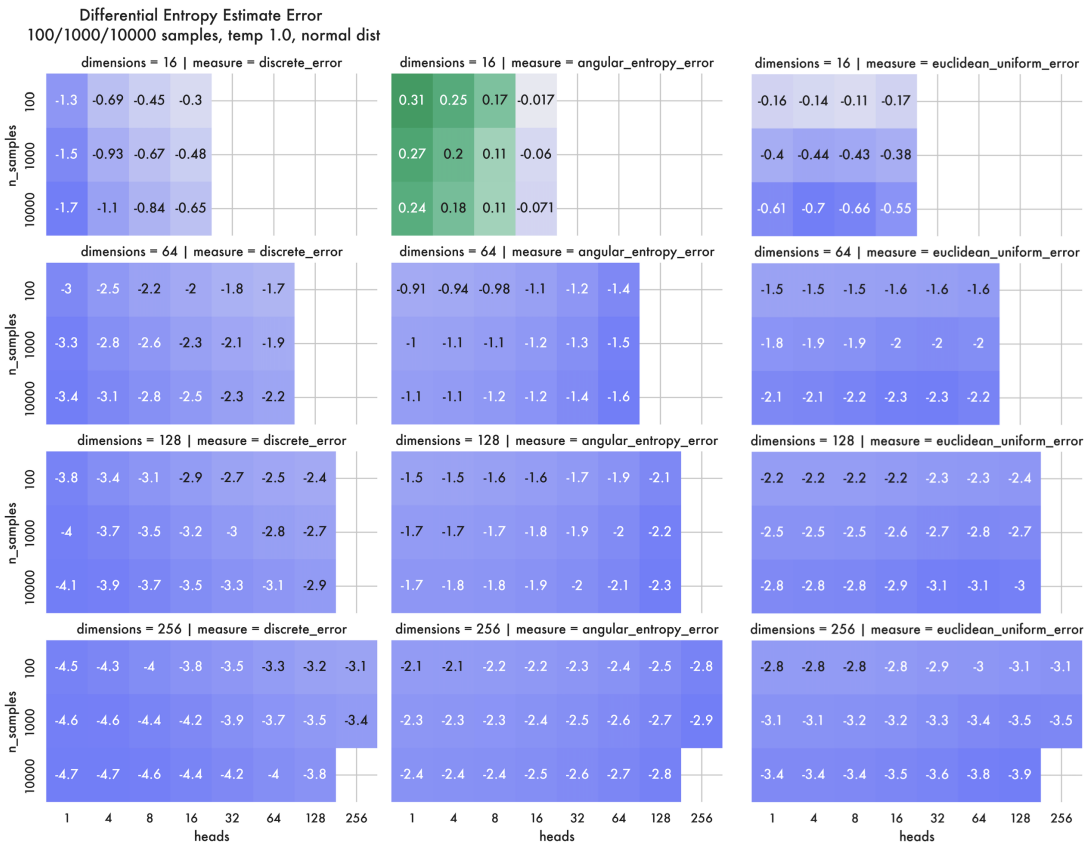
- Recurrent Unit: GRU
- Hidden Size: 250, 500, 800
- Entropy Regularization Coefficient: (sender 0.5, receiver 0.0)
- Batch Size: 5000
- Learning Rate: 1e-3
- Signal Length: 6
- Character Inventory: 26
- Training Epochs: 800
- Embedding Size: 52
- Roles: 3
- Atoms: 25
- Optimizer: (Sender: Reinforce, Receiver: ADAM)

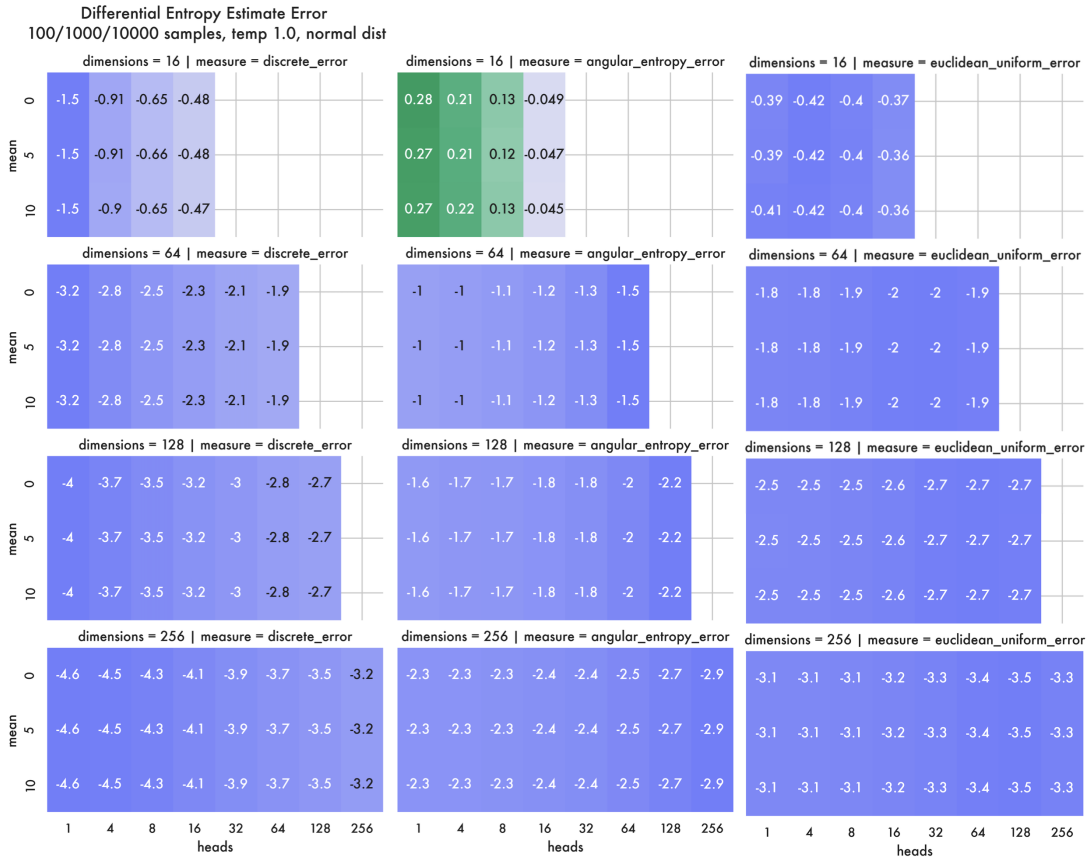
Appendix B

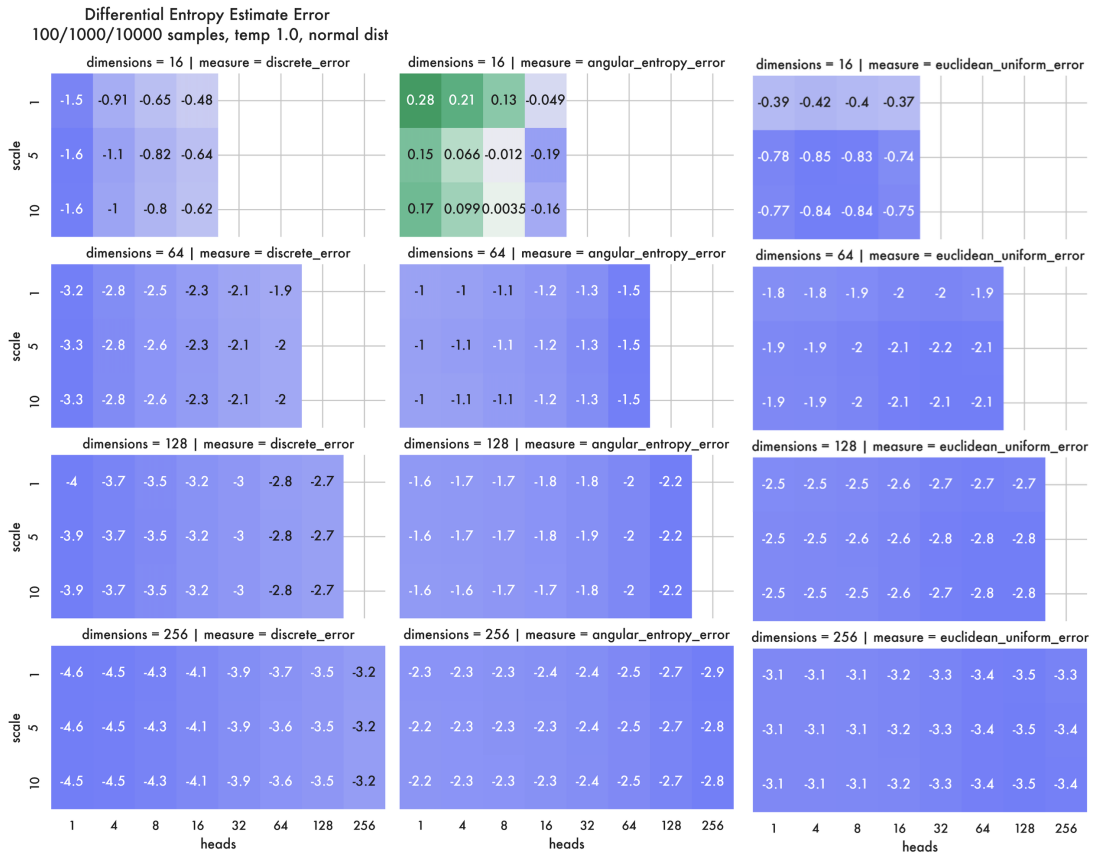
Information Structure in LLMs

B.1 Benchmarking: Effect of Number of Heads, Mean and Scale

We compare versions of our estimator across different levels of subspacing. Angular entropy is the version that appears in the paper, discrete follows the methods of soft entropy estimation but then argmaxes to assign representations to a single point on the sphere. We also include a version that uses euclidean distances instead of the cosine-sphere comparison used in the paper. In principle this is nice because models also represent information topographically, encoding meaning in magnitude as well as angle in representational space. In practice euclidean distances end up being dramatically less memory efficient (and a factor of 4 slower to compute) than cosine similarities when using built-in pytorch methods. This means for scalability reasons we elected to only focus on the cosine case.

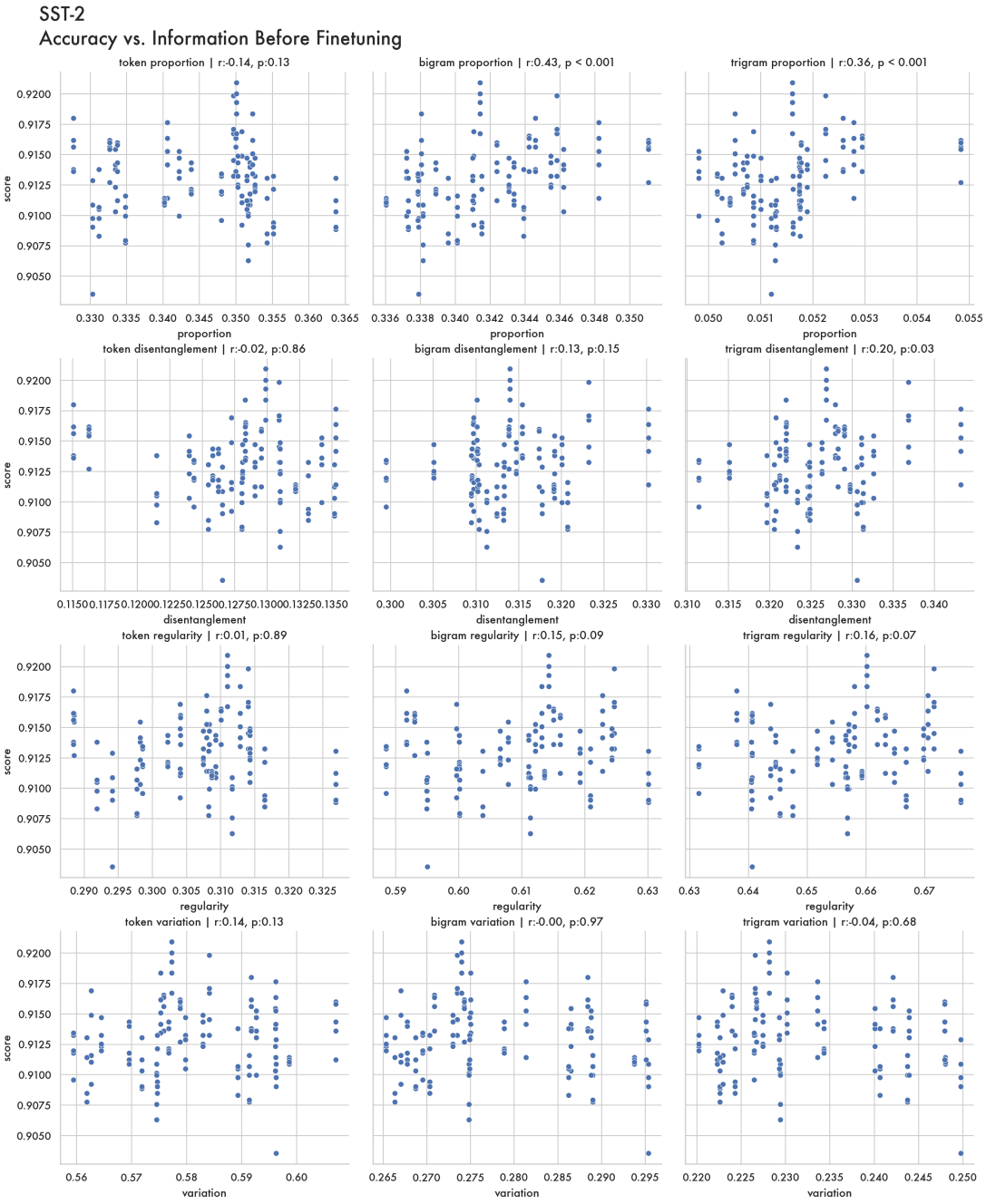

Differential Entropy Estimate Error
100/1000/10000 samples, temp 1.0, normal dist
dimensions = 16 | measure = discrete_error
dimensions = 16 | measure = angular_entropy_error
dimensions = 16 | measure = euclidean_uniform_error
dimensions = 64 | measure = discrete_error
dimensions = 64 | measure = angular_entropy_error
dimensions = 64 | measure = euclidean_uniform_error
dimensions = 128 | measure = discrete_error
dimensions = 128 | measure = angular_entropy_error
dimensions = 128 | measure = euclidean_uniform_error
dimensions = 256 | measure = discrete_error
dimensions = 256 | measure = angular_entropy_error
dimensions = 256 | measure = euclidean_uniform_error
n_samples
100
1000
10000
heads
1 4 8 16 32 64 128 256

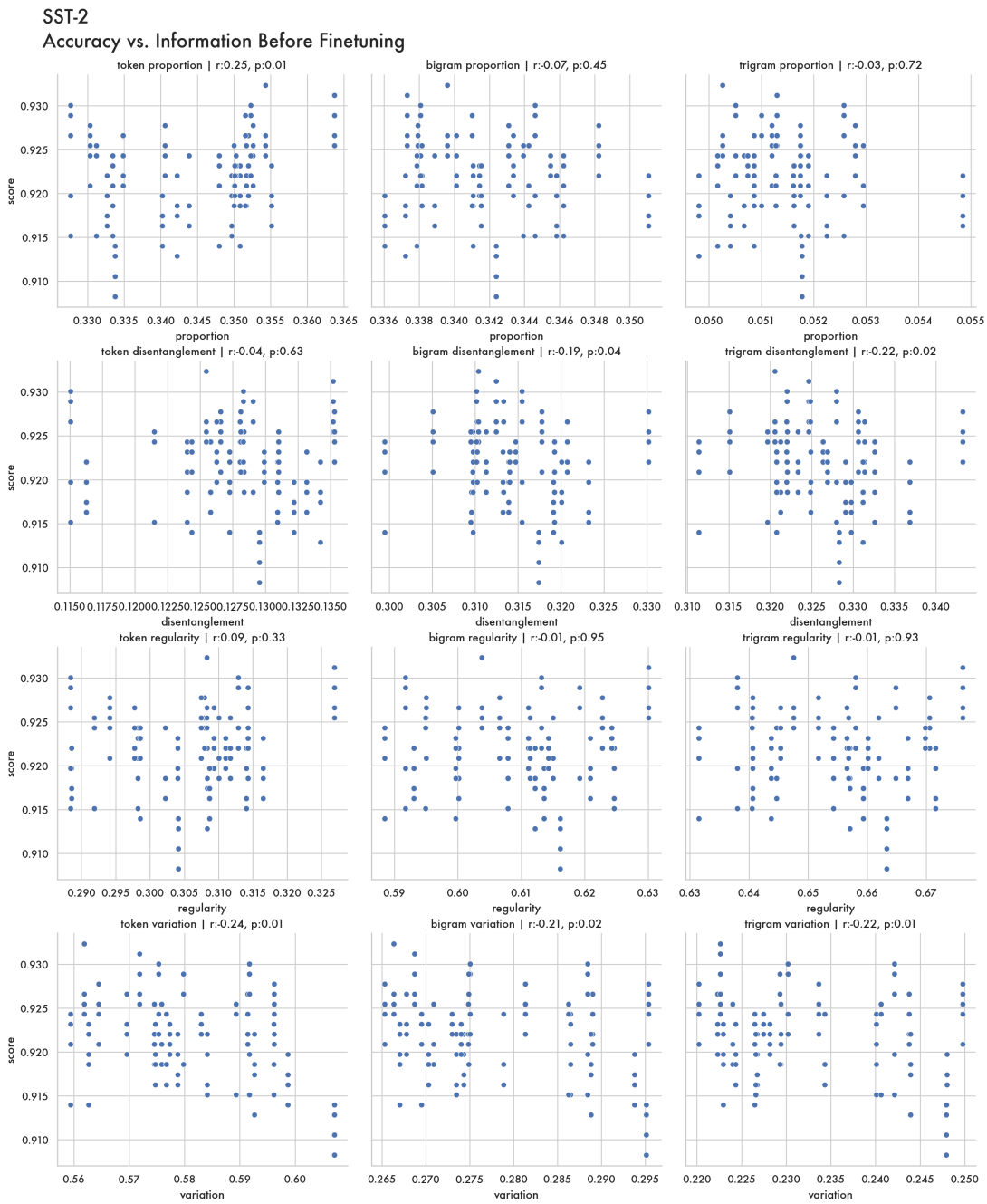

Differential Entropy Estimate Error
100/1000/10000 samples, temp 1.0, normal dist
dimensions = 16 | measure = discrete_error
dimensions = 16 | measure = angular_entropy_error
dimensions = 16 | measure = euclidean_uniform_error
dimensions = 64 | measure = discrete_error
dimensions = 64 | measure = angular_entropy_error
dimensions = 64 | measure = euclidean_uniform_error
dimensions = 128 | measure = discrete_error
dimensions = 128 | measure = angular_entropy_error
dimensions = 128 | measure = euclidean_uniform_error
dimensions = 256 | measure = discrete_error
dimensions = 256 | measure = angular_entropy_error
dimensions = 256 | measure = euclidean_uniform_error
mean
0
5
10
heads
1 4 8 16 32 64 128 256

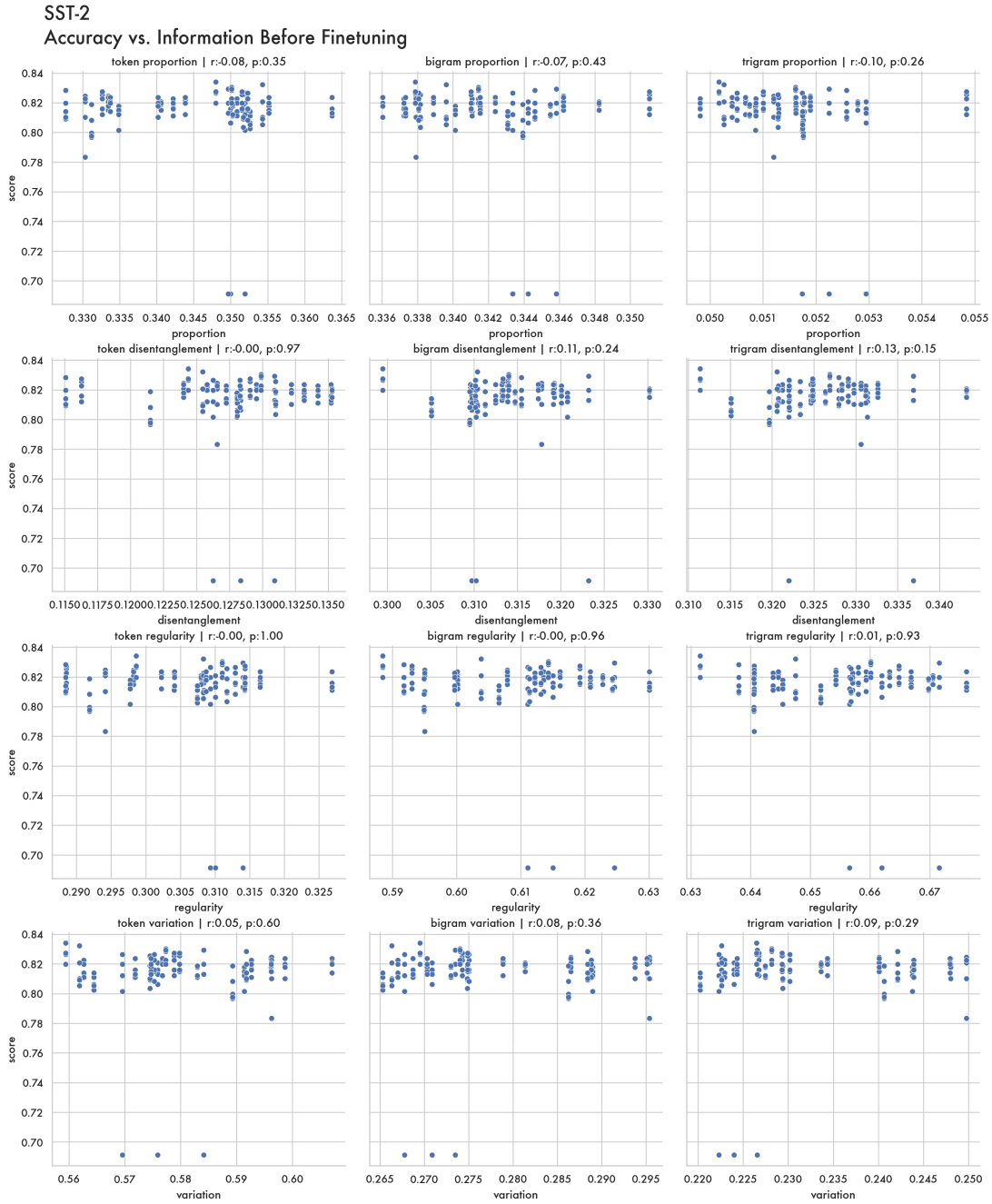


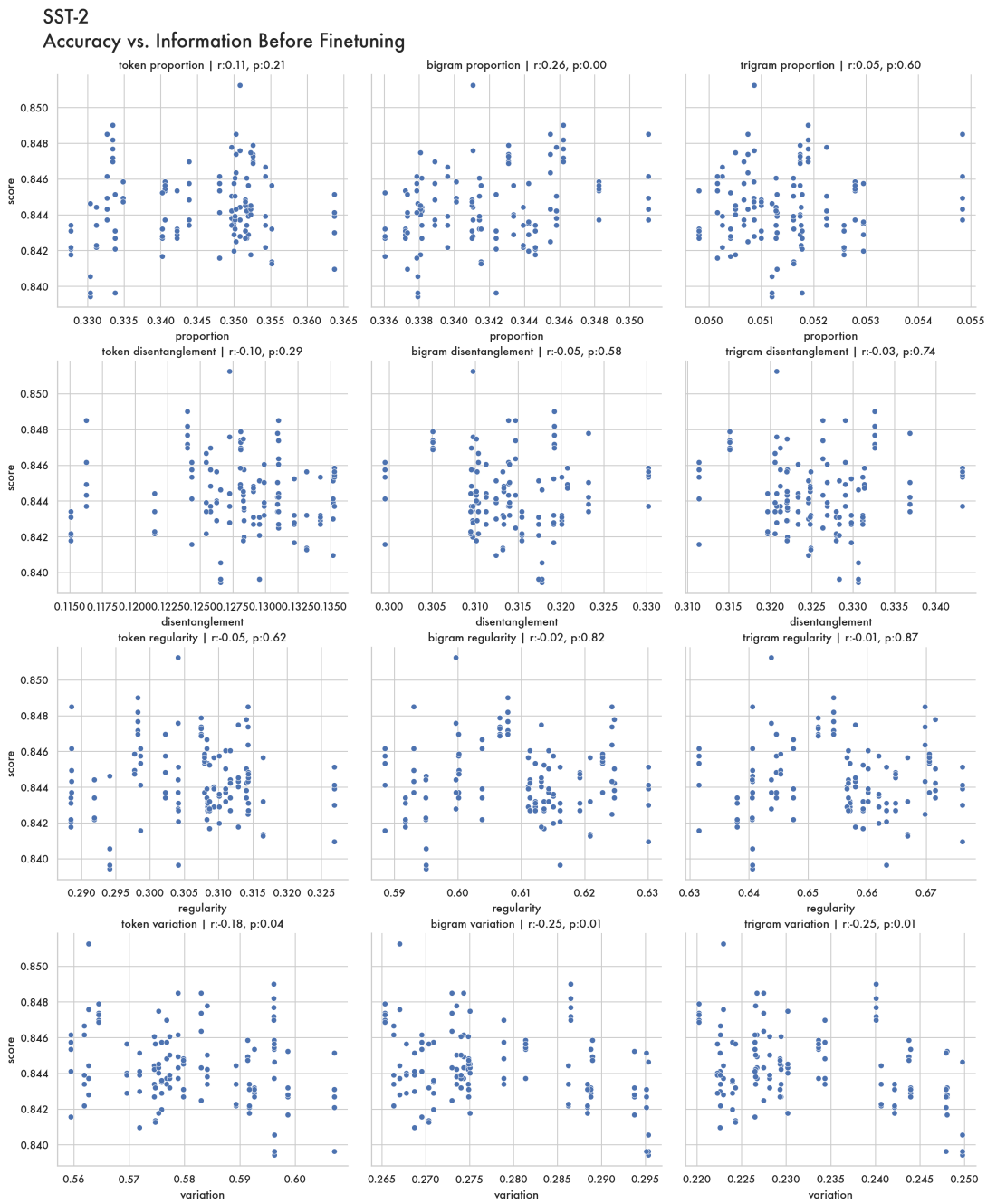
B.2 All GLUE Correlations by Task

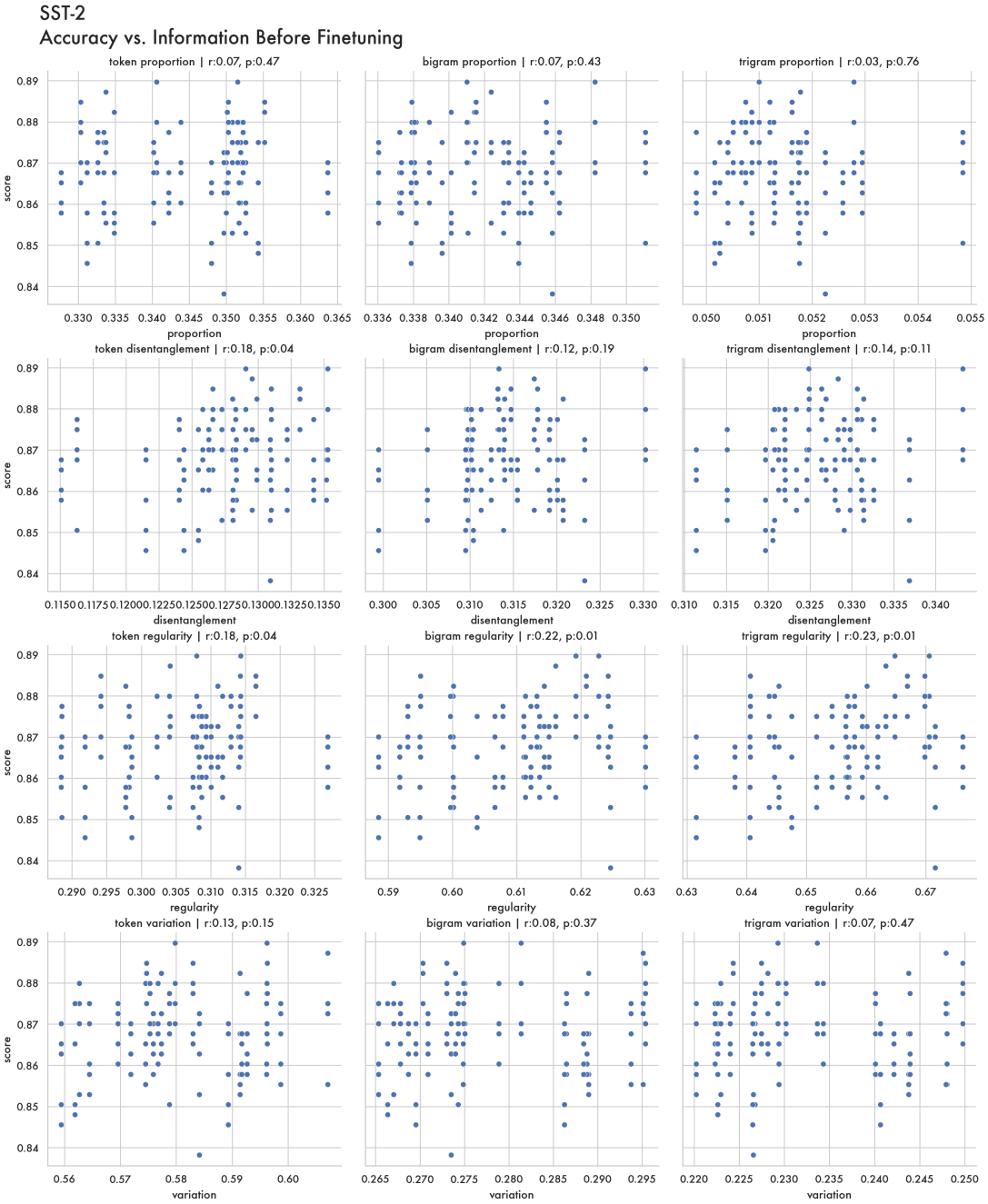
There are a huge volume of correlations for which I apologise

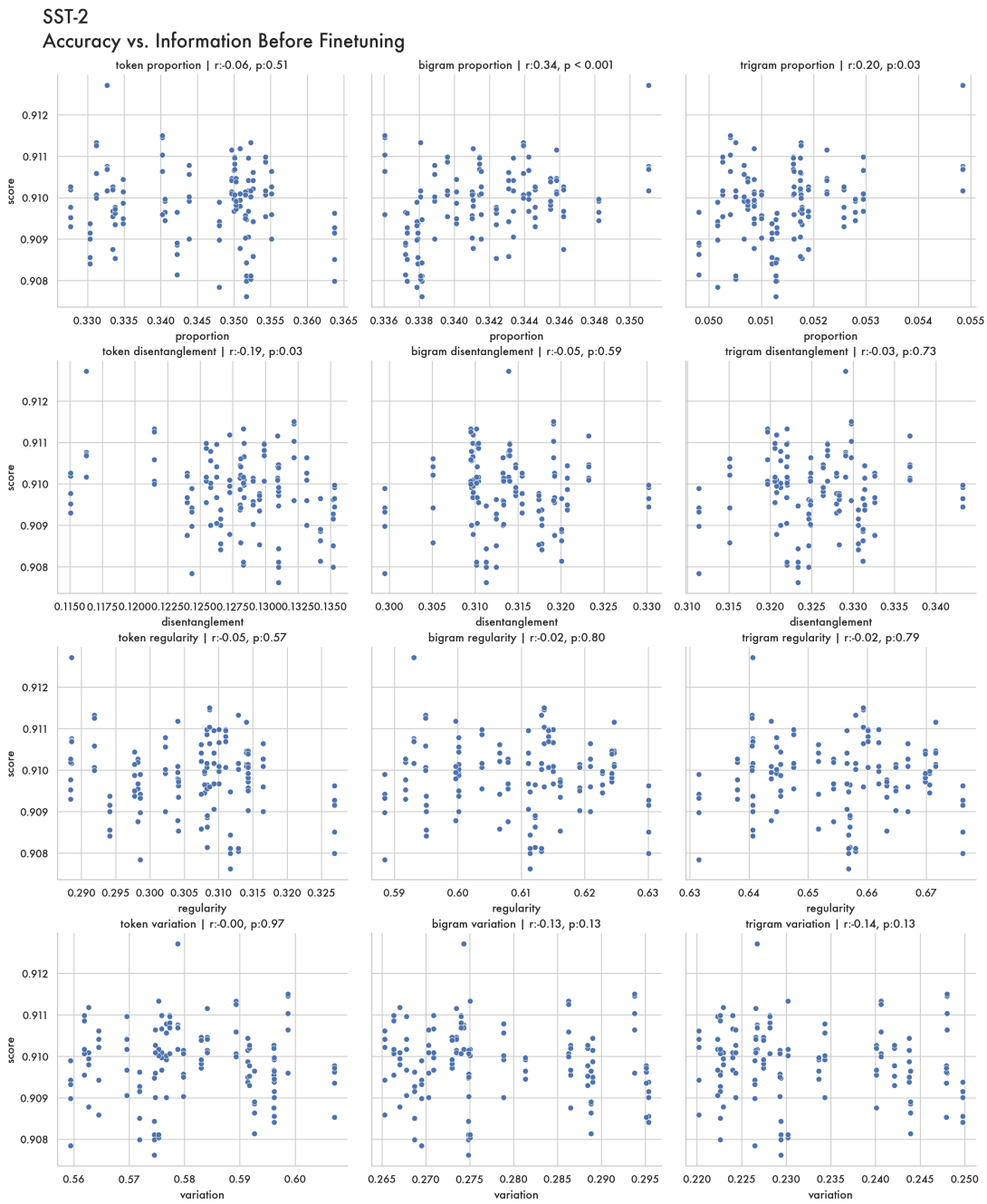

SST-2
Accuracy vs. Information Before Finetuning
token proportion | r:-0.14, p:0.13
bigram proportion | r:0.43, p < 0.001
trigram proportion | r:0.36, p < 0.001
token disentanglement | r:-0.02, p:0.86
bigram disentanglement | r:0.13, p:0.15
trigram disentanglement | r:0.20, p:0.03
token regularity | r:0.01, p:0.89
bigram regularity | r:0.15, p:0.09
trigram regularity | r:0.16, p:0.07
token variation | r:0.14, p:0.13
bigram variation | r:-0.00, p:0.97
trigram variation | r:-0.04, p:0.68
score
proportion
disentanglement
regularity
variation

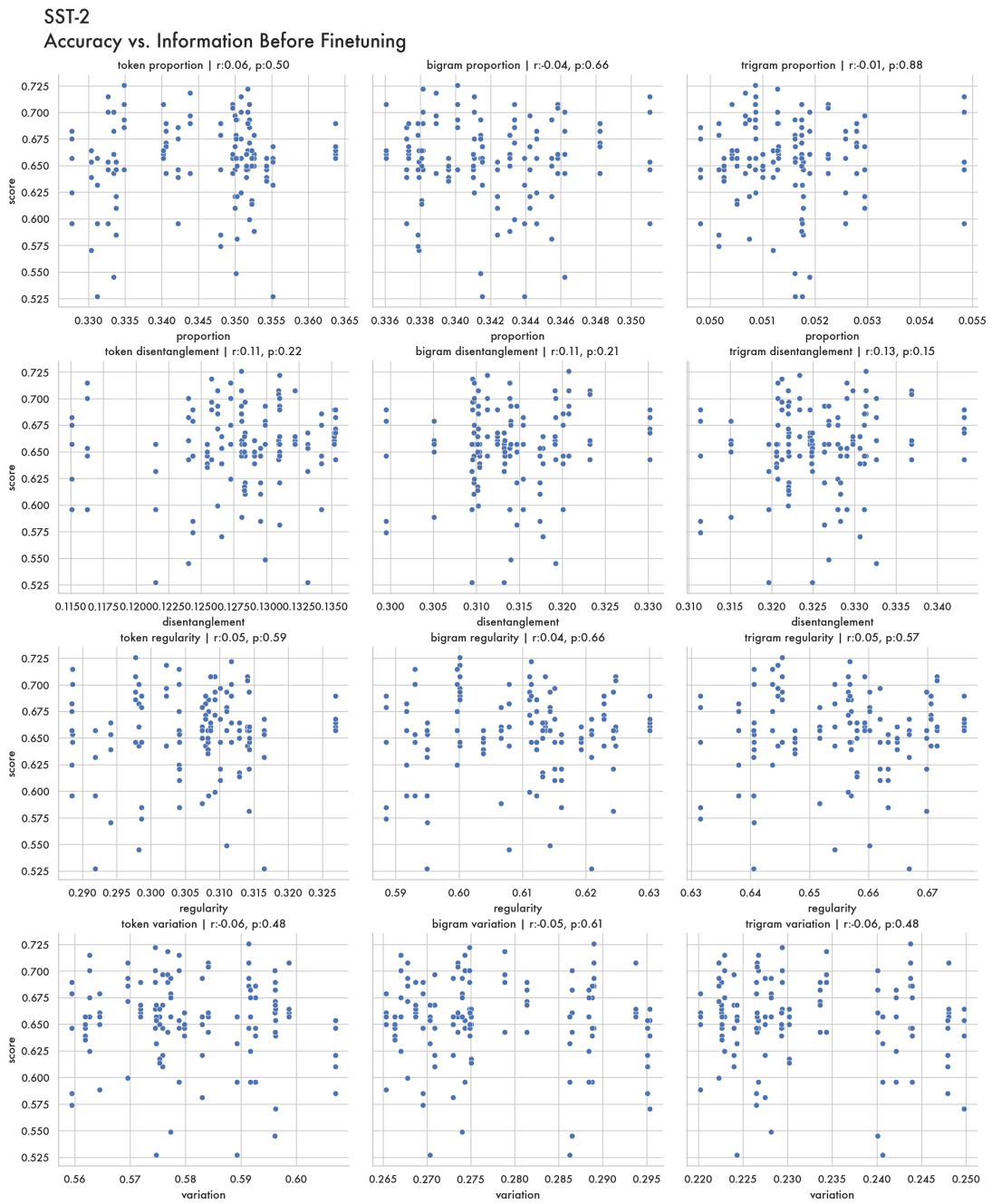

SST-2
Accuracy vs. Information Before Finetuning
token proportion | r:0.25, p:0.01
bigram proportion | r:-0.07, p:0.45
trigram proportion | r:-0.03, p:0.72
token disentanglement | r:-0.04, p:0.63
bigram disentanglement | r:-0.19, p:0.04
trigram disentanglement | r:-0.22, p:0.02
token regularity | r:0.09, p:0.33
bigram regularity | r:-0.01, p:0.95
trigram regularity | r:-0.01, p:0.93
token variation | r:-0.24, p:0.01
bigram variation | r:-0.21, p:0.02
trigram variation | r:-0.22, p:0.01
score
proportion
disentanglement
regularity
variation

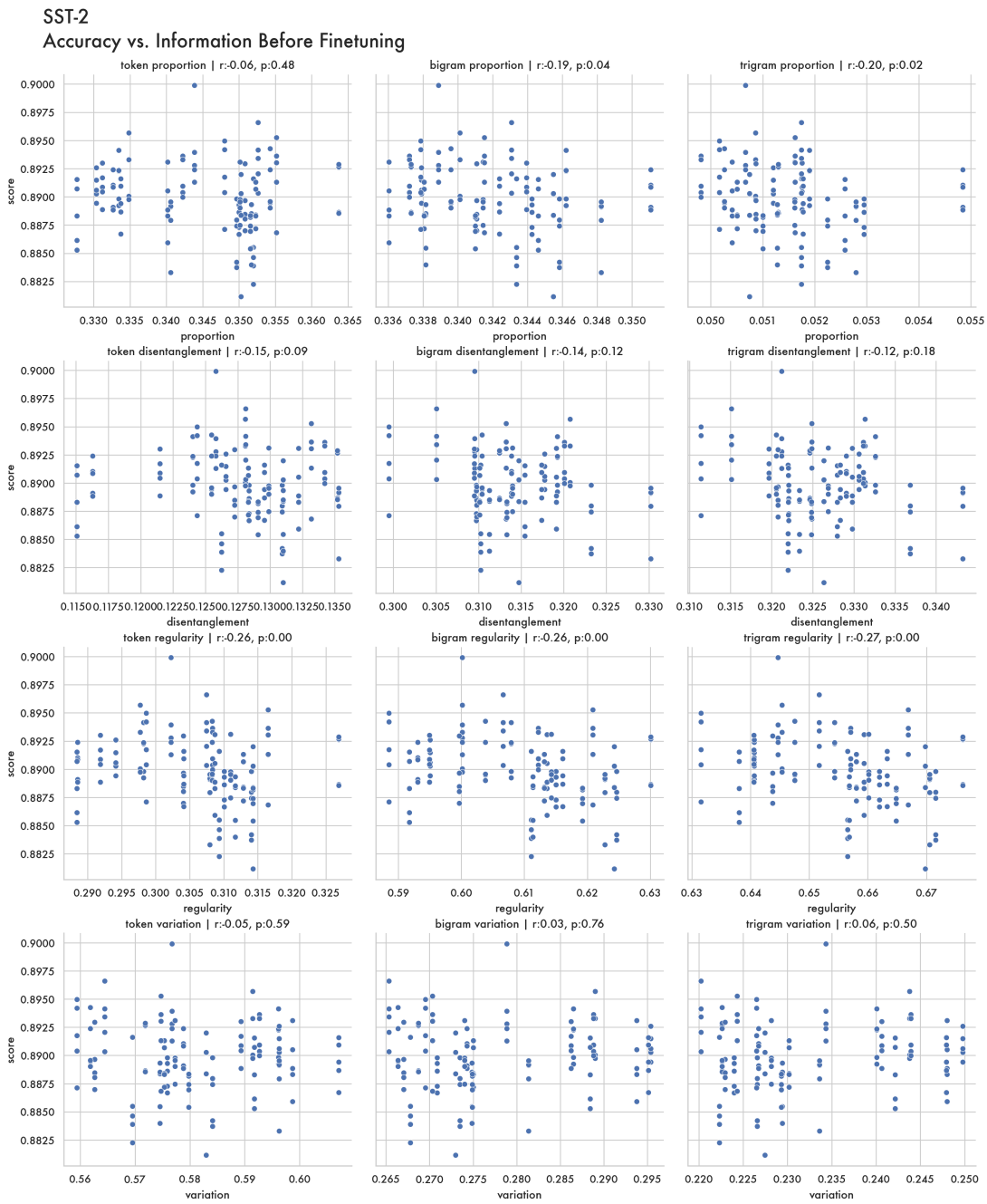

SST-2
Accuracy vs. Information Before Finetuning
token proportion | r:-0.08, p:0.35
bigram proportion | r:-0.07, p:0.43
trigram proportion | r:-0.10, p:0.26
token disentanglement | r:-0.00, p:0.97
bigram disentanglement | r:0.11, p:0.24
trigram disentanglement | r:0.13, p:0.15
token regularity | r:-0.00, p:1.00
bigram regularity | r:-0.00, p:0.96
trigram regularity | r:0.01, p:0.93
token variation | r:0.05, p:0.60
bigram variation | r:0.08, p:0.36
trigram variation | r:0.09, p:0.29
score
proportion
disentanglement
regularity
variation

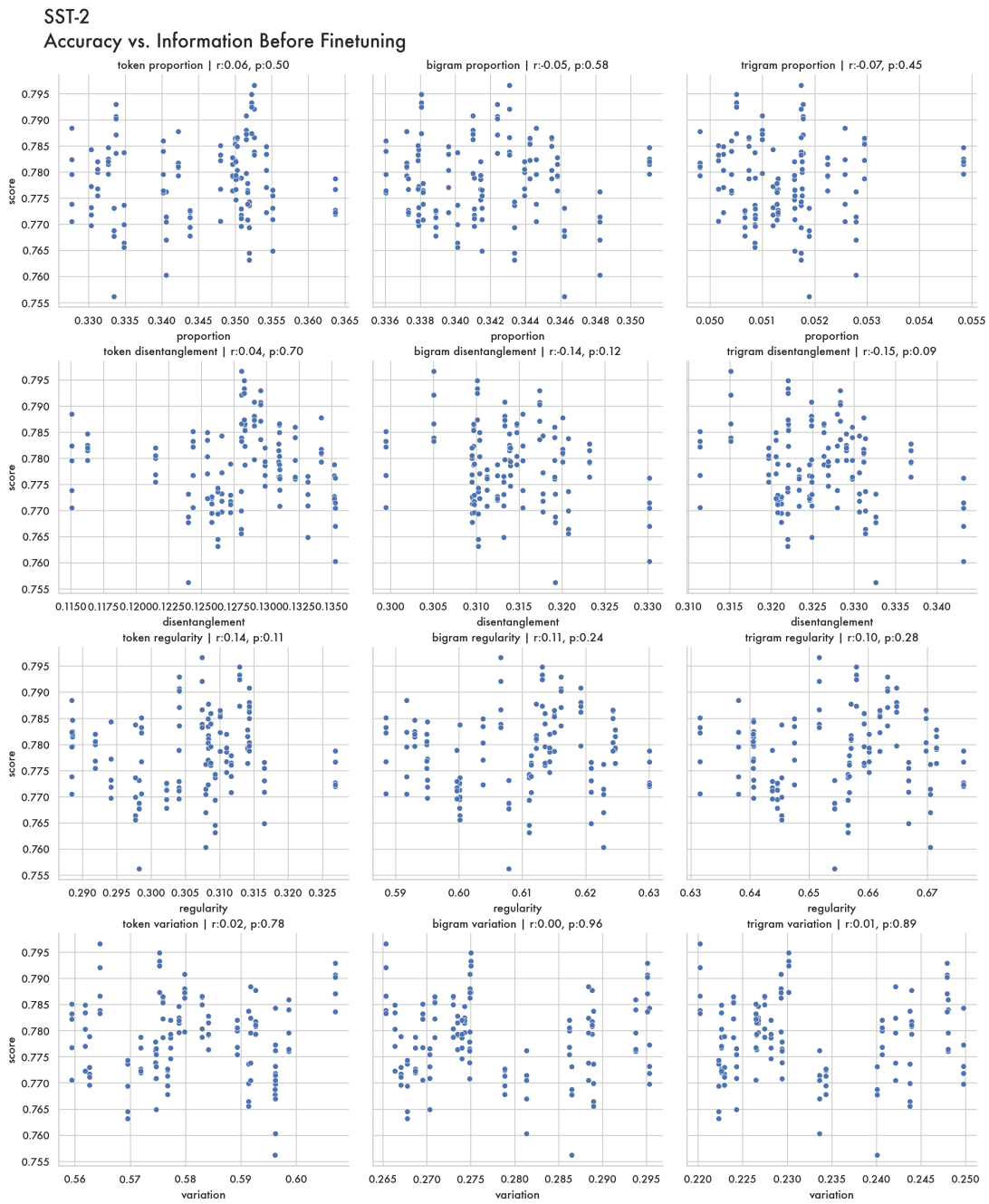

SST-2
Accuracy vs. Information Before Finetuning
token proportion | r:0.11, p:0.21
bigram proportion | r:0.26, p:0.00
trigram proportion | r:0.05, p:0.60
token disentanglement | r:-0.10, p:0.29
bigram disentanglement | r:-0.05, p:0.58
trigram disentanglement | r:-0.03, p:0.74
token regularity | r:-0.05, p:0.62
bigram regularity | r:-0.02, p:0.82
trigram regularity | r:-0.01, p:0.87
token variation | r:-0.18, p:0.04
bigram variation | r:-0.25, p:0.01
trigram variation | r:-0.25, p:0.01
score
proportion
disentanglement
regularity
variation


SST-2
Accuracy vs. Information Before Finetuning
token proportion | r:0.07, p:0.47
bigram proportion | r:0.07, p:0.43
trigram proportion | r:0.03, p:0.76
token disentanglement | r:0.18, p:0.04
bigram disentanglement | r:0.12, p:0.19
trigram disentanglement | r:0.14, p:0.11
token regularity | r:0.18, p:0.04
bigram regularity | r:0.22, p:0.01
trigram regularity | r:0.23, p:0.01
token variation | r:0.13, p:0.15
bigram variation | r:0.08, p:0.37
trigram variation | r:0.07, p:0.47
score
proportion
disentanglement
regularity
variation


SST-2
Accuracy vs. Information Before Finetuning
token proportion | r:-0.06, p:0.51
bigram proportion | r:0.34, p < 0.001
trigram proportion | r:0.20, p:0.03
token disentanglement | r:-0.19, p:0.03
bigram disentanglement | r:-0.05, p:0.59
trigram disentanglement | r:-0.03, p:0.73
token regularity | r:-0.05, p:0.57
bigram regularity | r:-0.02, p:0.80
trigram regularity | r:-0.02, p:0.79
token variation | r:-0.00, p:0.97
bigram variation | r:-0.13, p:0.13
trigram variation | r:-0.14, p:0.13
score
proportion
disentanglement
regularity
variation


SST-2
Accuracy vs. Information Before Finetuning
token proportion | r:0.06, p:0.50
bigram proportion | r:-0.04, p:0.66
trigram proportion | r:-0.01, p:0.88
token disentanglement | r:0.11, p:0.22
bigram disentanglement | r:0.11, p:0.21
trigram disentanglement | r:0.13, p:0.15
token regularity | r:0.05, p:0.59
bigram regularity | r:0.04, p:0.66
trigram regularity | r:0.05, p:0.57
token variation | r:-0.06, p:0.48
bigram variation | r:-0.05, p:0.61
trigram variation | r:-0.06, p:0.48
score
proportion
disentanglement
regularity
variation


SST-2
Accuracy vs. Information Before Finetuning
token proportion | r:-0.06, p:0.48
bigram proportion | r:-0.19, p:0.04
trigram proportion | r:-0.20, p:0.02
token disentanglement | r:-0.15, p:0.09
bigram disentanglement | r:-0.14, p:0.12
trigram disentanglement | r:-0.12, p:0.18
token regularity | r:-0.26, p:0.00
bigram regularity | r:-0.26, p:0.00
trigram regularity | r:-0.27, p:0.00
token variation | r:-0.05, p:0.59
bigram variation | r:0.03, p:0.76
trigram variation | r:0.06, p:0.50
score
proportion
disentanglement
regularity
variation


SST-2
Accuracy vs. Information Before Finetuning
token proportion | r:0.06, p:0.50
bigram proportion | r:-0.05, p:0.58
trigram proportion | r:-0.07, p:0.45
token disentanglement | r:0.04, p:0.70
bigram disentanglement | r:-0.14, p:0.12
trigram disentanglement | r:-0.15, p:0.09
token regularity | r:0.14, p:0.11
bigram regularity | r:0.11, p:0.24
trigram regularity | r:0.10, p:0.28
token variation | r:0.02, p:0.78
bigram variation | r:0.00, p:0.96
trigram variation | r:0.01, p:0.89
score
proportion
disentanglement
regularity
variation

Appendix C

Meta-Learning To Compositionally Generalise

C.1 Details of Base Parsers

We implemented all models with Pytorch Paszke et al., 2019. For the LSTM parsers, we use a two-layer encoder and one-layer decoder with attention Bahdanau et al., 2014 and input-feeding Luong et al., 2015. We only test bidirectional LSTM encoders, as unidirectional LSTM models do not perform very well in our preliminary experiments. For Transformer parsers, we use 2 encoder and decoder layers, 4 attention heads, and a feed-foward dimension of 1024. The hidden size for both LSTM and Transformer models are 256. The hyparameters of base parsers are mostly borrowed from related work and not tuned, as the primary goal of this work is the MAML training algorithm. To experiment with a wide variety of possible Seq2Seq models, we also try a Transformer encoder + LSTM decoder and find that this variant actually performs slightly better than both vanilla Transformer and LSTM models. Further exploration of this combination in pursuit of a better neural architecture for compositional generalization might be interesting for future work.

C.2 Details of Sampling for Meta-Test

The similarity-driven sampling distribution $\tilde{p}$ (Equation 4 of the main paper) requires computing the similarity between every pair of training examples, which can be very expensive depending on the size of of the dataset. As the sampling

distributions are fixed during training, we compute and cache them beforehand. However, they take an excess of disk space to store as essentially we need to store an $N \times N$ matrix where N is the number of training examples. To allow efficient storage and sampling, we use the following approximation. First, we found that usually each example only has a small set of neighbours that are relevant to it. For example, in COGS, each example only retrieves 3.6% of the whole training set as its neighbours (i.e., have non-zero tree-kernel similarity) on average. Motivated by this observation, we only store the top 1000 relevant neighbours for each example sorted by similarity, and use it to construct the sampling distribution denoted as $\tilde{p}_{top1000}$. To allow examples out of top 1000 being sampled, we use a linear interpolation between $\tilde{p}_{top1000}$ and a uniform distribution. Specifically, we end up using the following sampling distribution:

$$\tilde{p}(x', y'|x, y) = \lambda\ \tilde{p}_{top1000}(x', y'|x, y) + (1 - \lambda)\frac{1}{N} \tag{C.1}$$

where $\tilde{p}_{top1000}$ assigns 0 probability to out-of top-1000 examples, N is the number of training examples, and λ is a hyperparameter for interpolation. In practice, we set λ to 0.5 in all experiments.

To sample from this distribution, we first decide whether the sample is in the top 1000 by sampling from a Bernoulli distribution parameterized by λ. If it is, we use $\tilde{p}_{top1000}$ to do the sampling; if not, we just uniformly sample an example from the training set.

C.3 Model Selection Protocol

In our preliminary experiments on COGS, we find almost all the Seq2Seq models achieve $> 99\%$ in accuracy on the original Dev set. However, their performance on the Gen set diverge dramatically, ranging from 10% to 70%. The lack of an informative Dev set makes model selection extremely difficult and difficult to reproduce. This issue might also be one of the factors that results in the large variance of performance reported in previous work. Meanwhile, we found that some random seeds [1] yield consistently better performance than others across different conditions. For example, among the ten random seeds used for Lev-MAML + Transformer on COGS, the best performing seed obtains 73% whereas the lowest performing seed obtains 54%. Thus, it is important to compare different models

[1] Random seeds control the initialization of parameters and the order of training batches.

using the same set of random seeds, and not to tune the random seeds in any model. To alleviate these two concerns, we choose the protocol that is mentioned in the main paper. This protocol helps to make the results reported in our paper reproducible.

C.4 Details of Training and Evaluation

Following Kim and Linzen, 2020, we train all models from scratch using randomly initialized embeddings. For SCAN, models are trained for 1,000 steps with batch size 128. We choose model checkpoints based on their performance on the Dev set. For COGS, models are trained for 6,000 steps with batch size of 128. We choose the meta-train learning rate (α in Equation 2 of the main paper), temperature (η in Equation 4 of the main paper) based on the performance on the Gen Dev set. Finally we use the chosen α, η to train models with new random seeds, and only the last checkpoints (at step 6,000) are used for evaluation on the Test and Gen set.

C.5 Other Splits of SCAN

The SCAN dataset contains many splits, such as Add-Jump, Around Right, and Length split, each assessing a particular case of compositional generalization. We think that MCD splits are more representative of compositional generalization due to the nature of the principle of maximum compound divergence. Moreover, it is more challenging than other splits (except the Length split) according to Furrer et al. (2021). That GECA, which obtains 82% in accuracy on JUMP and Around Right splits, only obtains $< 52\%$ in accuracy on MCD splits in our experiments confirms that MCD splits are more challenging.

C.6 Kernel Analysis

The primary difference between the tree-kernel and string-kernel methods is in the diversity of the examples they select for the meta-test task. The tree kernel selects a broader range of lengths, often including atomic examples, a single word in length, matching a word in the original example from meta-train (see table C.1). By design the partial tree kernel will always assign a non-zero value to an example

Partial Tree Kernel	**top 10**	**100**	**1000**	**LevDistance**	**top 10**	**100**	**1000**
Mean Example Length (chars)	26.71	26.59	29.87	Mean Example Length (chars)	31.04	30.45	29.28
Std dev	± 6.80	± 7.61	± 8.85	Std dev	± 2.80	± 3.77	± 4.78
Mean No. of Atoms	0.46	0.81	1.13	Mean No. of Atoms	0.00	0.00	0.02
Std dev	± 0.67	± 1.05	± 0.81	Std dev	± 0.00	± 0.02	± 0.17

Table C.1: Analyses of kernel diversity. Reporting mean example length and number of atoms for the top k highest scoring examples for each kernel. Note that atoms are only counted that also occur in the original example.

that is an atom contained in the original sentence. We believe the diversity of the sentences selected by the tree kernel accounts for the superior performance of Tree-MAML compared with the other MAML conditions. The selection of a variety of lengths for meta-test constrains model updates on the meta-train task such that they must also accommodate the diverse and often atomic examples selected for meta-test. This constraint would seem to better inhibit memorizing large spans of the input unlikely to be present in meta-test.

C.7 Meta-Test Examples

In Table C.2, we show top scoring examples retrieved by the similarity metrics for two sentences. We found that in some cases (e.g., the right part of Table C.2), the tree-kernel can retrieve examples that diverge in length but are still semantically relevant. In contrast, string-based similarity metrics, especially LevDistance, tends to choose examples with similar lengths.

C.8 COGS Subtask Analysis

We notice distinct performance for different conditions on the different subtasks from the COGS dataset. In Figure C.1 we show the performance of the Uni-MAML and Str-MAML conditions compared with the mean of those conditions. Where the bars are equal to zero the models' performance on that task is roughly equal.

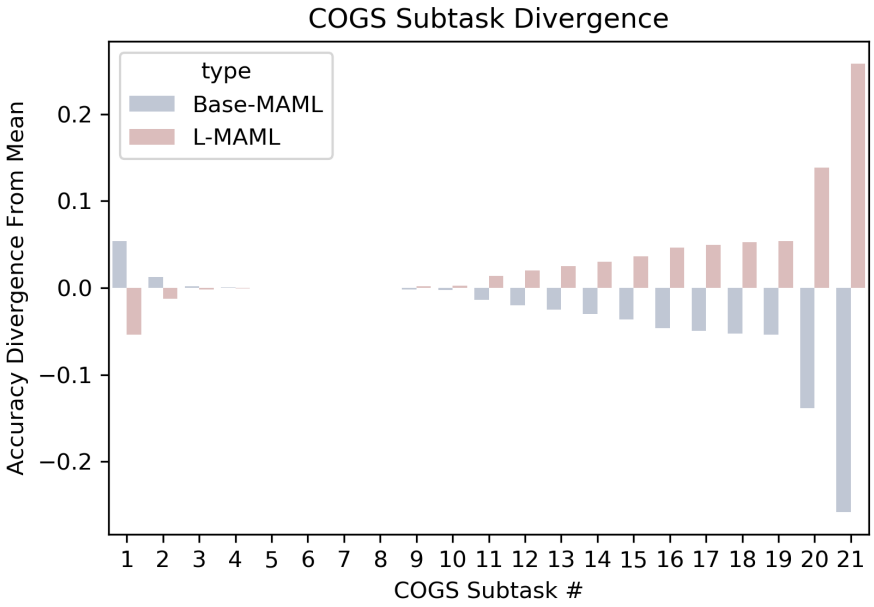


Figure C.1: Performance for the Uni-MAML and Lev-MAML conditions compared to the mean of those two conditions.

(1) prim→subj proper,

(2) active→passive,

(3) only seen as unacc subj → unerg subj,

(4) subj→obj proper,

(5) only seen as unacc subj → obj omitted transitive subj,

(6) pp recursion,

(7) cp recursion,

(8) obj pp→subj pp,

(9) obj→subj common,

(10) do dative→pp dative,

(11) passive→active,

(12) only seen as transitive subj → unacc subj,

(13) obj omitted transitive→transitive,

(14) subj→obj common,

(15) prim→obj proper,

(16) obj→subj proper,

(17) pp dative→do dative,

(18) unacc→transitive,

(19) prim→subj common,

(20) prim→obj common,

(21) prim→inf arg.

Source Example: Emma lended the donut to the dog .

Neighbours using Tree Kernel	Similarity
Emma was lended the donut .	0.74
The donut was lended to Emma .	0.62
Emma lended the donut to a dog .	0.55
Emma lended Liam the donut .	0.55
Emma lended a girl the donut .	0.55
Neighbours using String Kernel	
Emma lended the donut to a dog .	0.61
Emma lended the box to a dog .	0.36
Emma gave the cake to the dog .	0.33
Emma lended the cake to the girl .	0.33
Emma lended the liver to the girl .	0.33
Neighbours using LevDistance	
Emma lended the donut to a dog .	-1.00
Emma loaned the donut to the teacher .	-2.00
Emma forwarded the donut to the monster .	-2.00
Emma gave the cake to the dog .	-2.00
Charlotte lended the donut to the fish .	-2.00

Source Example: The crocodile valued that a girl snapped .

Neighbours using Tree Kernel	Similarity
A girl snapped .	0.55
A rose was snapped by a girl .	0.39
The cookie was snapped by a girl .	0.39
girl	0.32
value	0.32
Neighbours using String Kernel	
The crocodile liked a girl .	0.28
The girl snapped .	0.27
The crocodile hoped that a boy observed a girl .	0.26
The boy hoped that a girl juggled .	0.15
The cat hoped that a girl sketched .	0.15
Neighbours using LevDistance	
The crocodile liked a girl .	-3.00
The boy hoped that a girl juggled .	-3.00
The cat hoped that a girl sketched .	-3.00
The cat hoped that a girl smiled .	-3.00
Emma liked that a girl saw .	-4.00

Table C.2: Top scoring examples according to the tree kernel, string kernel and Levenshtein distance for two sentences and accompanying scores.