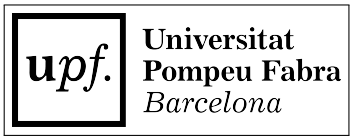
Interpreting the Learning Dynamics of Language Models

Lucas Weber

DOCTORAL THESIS UPF / YEAR 2023

THESIS SUPERVISORS
Elia Bruni, PhD; Dieuwke Hupkes, PhD
Department of Translation and Language Sciences



Acknowledgements

Members of the COLT group established the practice of referring to each other with kinship terms: there are the parents, the kids, siblings, twins and half-twins, but there are also ‘friends’ – which are people beyond the inner ‘family’. This practice reminds me of the proverb ‘it takes a village to raise a child’, which is known in many different African societies. Children are not only the responsibility of their parents but the concern of the entire community, and, this way, children can benefit from the guidance, care and influence of a variety of community members with different skills, views and enriching personalities.

In the COLT group, the PhD students – the reader might have guessed it – are the kids. Just like growing up as a kid, achieving a doctorate is a journey through highs and lows. And just like growing up, overcoming the lows and enjoying the highs of a doctorate becomes much more doable with the support, patience, and goodwill of a metaphorical village. I here want to express my gratitude towards all the residents of my personal village who were so very supportive during my ‘growing up’.

First and foremost, I want to thank Elia Bruni and Dieuwke Hupkes. It takes courage to take on a PhD student who comes from a different field of expertise. Thank you for your patience, time, and forward-thinking, despite the many other engagements you are handling. I have learned a lot from you and I am very thankful for that.

In terms of learning academic practices, Paul Michel and Jaap Jumelet had the best influence on me. I have never before encountered Paul’s understanding of not only research but also of people. His capacity to see through a problem within the shortest amount of time and suggest the appropriate solutions left me in awe and inspired me in many ways. Jaap on the other hand helped me better understand linguistic research and – throughout the early stages of my PhD – he served as an excellent role model of a young researcher.

At the University Pompeu Fabra (UPF), I would like to express my deepest gratitude to Gemma Boleda and Marco Baroni. Both made all the effort to include the cuckoo’s child in their group, supporting me on many

ends and making me feel included. In many cases, they proactively reached out and made many extra efforts to assist me. Ionut Șorodoc and Laura Aina – the 'older siblings' – made COLT and Barcelona accessible during the early times of my project and their knowledge of technicalities, but also the emotional journey, was invaluable. A large 'thank you' to Mateo Mahaut, Carraz Rakotonirina, Emily Cheng, Eleonora Gualdoni, Roberto Dessi, Francesca Franzon, Corentin Kervadec and Thomas Brochhagen, for creating an open and pleasant atmosphere, always being willing to listen and aiding when necessary. I want to express my special appreciation to Eleonora for being a great colleague and to Emily for doing such inspirational work that opened me to new perspectives on model interpretability, for always being there when needed, reminding me to not only overcome the 'lows' but to also enjoy the 'highs', and for being a friend. Ultimately, I hope my 'academic twin' – even though we may not be identical – Xixian Liao, knows how much I appreciate her going the whole way with me. You are a special person and I am happy that it was you I shared this experience with.

Moving beyond UPF, I want to express my appreciation to the members of the board evaluating this thesis: Denis Paperno, Raffaella Bernardi and Barbara Plank. I am pleased that you have agreed to be part of this, and I sincerely appreciate your commitment of time to my work.

The research presented in this thesis would not have been possible without the funding I received through the DTCL scholarship of the UPF and the computing resources at UPF enabled through the European Research Council (ERC) grant of Marco Baroni (agreement No. 101019291). Besides the infrastructure and financial support by UPF, I am also thankful for all the personal assistance I experienced in the day-to-day research work. The engineers at UPF IT helped out with their relentless energy to resolve even the most entangled problems in the HPC infrastructure. Thanks to the security team for letting me access the campus at even the most obscure hours.

A shout out to the hives of friends from Kassel, Vienna, and Amsterdam, for always welcoming me back and being such an amazing, delightful consistency in my life. Equally to everyone in Barcelona who became an

appreciated friend. Ultimately, nothing can be achieved without the most caring people in the background. Laura Castro Moreno por – literalmente – ensenjarme hablar, tu patiencia, el apoyo que me diste en todo lo que hice, y tu gran corazón. Tasja und Jan, für die viele Nachsicht die ein kleiner Bruder verlangt und dass ihr dabei geholfen habt den Menschen zu formen der ich bin. Zu allerletzt, die Beiden, die die Basis geschaffen haben für alles was war, was ist und was noch wird, Renate und Joachim.

Thank you!

Abstract

Language models (LMs) have evolved to become remarkably capable yet similarly complex and intransparant systems. Our ability to understand how they achieve their outstanding traits – i.e. making them interpretable – can be achieved from different angles. In this dissertation, I analyse the learning dynamics of LMs and seek to understand the relationship between the properties of training data and the models' generalization behaviours. I introduce a framework that links generalisation with conceptual knowledge, specifically linguistic theory, which can be used for model analysis or model-driven hypothesis testing. This approach is applied to analyze the pre-training process of LMs. Furthermore, I delve into the dynamics of new learning paradigms, such as in-context learning, contributing to our understanding of their inconsistent prediction behaviour. Recognizing that the analysis of complex systems often demands holistic methods, this dissertation emphasizes and employs innovative and systematic methodologies for interpretability.

Resum

Els models de llenguatge (MLs) han evolucionat per esdevenir notablement capaços, però al mateix temps són sistemes complexos i poc transparents. La nostra capacitat per entendre com aconsegueixen aquestes característiques destacades – és a dir, fent-los interpretables – es pot aconseguir des de diferents punts de vista. En aquesta dissertació, analitzo la dinàmica d'aprenentatge dels MLs. Introdueixo un marc que enllaça la generalització amb el coneixement conceptual, específicament la teoria lingüística, que pot ser utilitzat per a l'anàlisi del model o per a la prova d'hipòtesis dirigida pel model. Aquest enfocament s'aplica per analitzar el procés de pre-entrenament dels MLs. A més, m'endinso en la dinàmica de nous paradigmes d'aprenentatge, com ara el in-context learning, il·luminant les raons del seu comportament de predicció inconsistent. Reconèixer que l'anàlisi de sistemes complexos sovint exigeix mètodes holístics, aquesta dissertació emfatitza i utilitza metodologies innovadores i sistemàtiques per a la interpretabilitat.

Resumen

Los modelos de lenguaje (MLs) han evolucionado para convertirse en sistemas notablemente capaces, pero al mismo tiempo son sistemas complejos y poco transparentes. Nuestra habilidad para entender cómo logran sus características sobresalientes – es decir, haciéndolos interpretables – puede lograrse desde diferentes perspectivas. En esta disertación, analizo la dinámica de aprendizaje de los MLs. Introduzco un marco que vincula la generalización con el conocimiento conceptual, específicamente la teoría lingüística, que puede ser utilizado para el análisis del modelo o para la prueba de hipótesis dirigida por el modelo. Este enfoque se aplica para analizar el proceso de preentrenamiento de los MLs. Además, profundizo en la dinámica de nuevos paradigmas de aprendizaje, como el in-context learning, iluminando las razones de su comportamiento de predicción inconsistente. Reconociendo que el análisis de sistemas complejos a menudo requiere métodos holísticos, esta disertación enfatiza y emplea metodologías innovadoras y sistemáticas para la interpretabilidad.

Statement of Transparency on the Use of AI Tools

During the writing of this dissertation, the following tools were used:

Grammarly: An AI-based writing aid used to correct spelling and improve phrasing (Grammarly, 2023).

OpenAI's chatGPT: A language modelling-based AI-tool used as assistance to phrasing and formatting (OpenAI, 2023).

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Overview

Language is complex; it exhibits such intricate structure that it is hard to formally describe it (Chomsky, 1957, 1965). At the same time, it integrates a multitude of human cognitive functions (Jackendoff, 2002; Baddeley, 2003; Boroditsky, 2001a; Thierry et al., 2009; Lindquist and Gendron, 2013; Fodor, 1975), and its inherent intrapersonal character (Wittgenstein, 1953) allows humans to create and maintain large societies (Deacon, 1997; Sperber, 1996; Tomasello, 2009). Despite this complexity, most humans are capable of learning their native language variation at a young age from relatively sparse exposure (Hart and Risley, 1995; Hoff, 2003; Huttenlocher et al., 2010; Rowe, 2012; Weisleder and Fernald, 2013; Gilkerson et al., 2017). This concurrence of complexity and learnability has long intrigued researchers in many scientific fields. I share this fascination.

In recent years, researchers have been successful in replicating language abilities in-silico through so-called *language models* (LMs) in a way that is almost indistinguishable from the human language faculty (see, e.g. Liang et al., 2022). Considering the mentioned complexities, how are LMs able to acquire such a skill? In this dissertation, I delve into the analysis of learning processes in LMs. More concretely, this dissertation adds to the research on the interpretability of LMs, a young yet established

subfield in machine learning research. Interpretability focuses on making the processing and behaviour of machine learning models understandable for humans. The presented work breaks out of the young discipline's conventions by taking an unusual angle at the topic: the methods of analysis are mostly *holistic*, and the subject of study is not the functionality of common language models but rather how language models obtain their functionality – or in other words – their *learning dynamics*.

The remainder of this introductory chapter lays out the motivation for my research and my approach (Section 1.2), outlines the goals of the dissertation (Section 1.3), and, finally, provides an overview of the structure of the dissertation (Section 1.4). Helpful explanations of important concepts are given in the subsequent Background section (see Chapter 2).

1.2 Motivation

The research presented in this dissertation follows three different leitmotivs: I focus on *language models*, I use *holistic* methods, and the subject of investigation is the *learning dynamics*. This section motivates these themes.

1.2.1 Why language modelling?

Language is one of the most interconnected faculties in the human cognitive apparatus: **within the individual**, cognitive functions such as language are considered to be embodied (i.a. Lakoff and Johnson, 1980; Varela et al., 1991; Barsalou, 1999; Zwaan, 2004; Gallese and Lakoff, 2005; Barsalou, 2008; Casasanto, 2011; Meteyard et al., 2012). Embodiment entails that cognitive functions like language are constantly interacting with the *perceptual and motor systems*, influencing each other and partially even sharing the same neural substrate (i.a. Hauk et al., 2004; Pulvermüller et al., 2005; Pulvermüller, 2005; Boulenger et al., 2006; Martin, 2007; Binder and Desai, 2011; Fedorenko and Thompson-Schill, 2014; Caucheteux and King, 2022). In the realm of perception, our language is highly connected

with our visual, auditory, olfactory or tactual experience (Majid and Levinson, 2011), while the influence of motor control is, for example, evident in many non-verbal behaviours that complement verbal communication (McNeill, 1992; Kelly et al., 2010; Özyürek, 2014). Further, internal states such as our *emotions* or *nociception* influence what we say and how we say it (Niedenthal, 2007; Majid, 2012; Banse and Scherer, 1996; Scherer, 2003; Kousta et al., 2009). Inversely, processing of language can evoke a wide range of emotions. Finally, language requires *cognitive control* for, i.a., the allocation of attention, adaptation to social contexts or management of working memory (Green, 1998; Blumenfeld and Marian, 2011; Kroll and Bialystok, 2013). In summary, the majority of commonly investigated cognitive functions interact with language. The centrality of language to human cognition is so significant that strong interpretations of ideas like linguistic relativity (Sapir, 1929; Whorf, 1940; Boroditsky, 2001b; Spelke and Tsivkin, 2001) – also known as *linguistic determinism* – even go as far as equating language with cognition, proposing that no thought beyond the structure of one's native language is possible. While such extreme views are not very well supported by empirical evidence (see, e.g. Regier and Kay, 2009), it highlights the pivotal role and interconnectedness of language in cognition.

However, language complexity is not limited to intra-personal processes. **Beyond the individual**, language adapts to situational and social contexts (Levinson, 1983), to facilitate the diffusion of information within larger groups to solve problems collaboratively (e.g. Wittgenstein, 1953; Lewis, 1969) or accelerate social learning. In the broader context, it is both the product and the vehicle of the cultural evolution in the larger body of human society (Boyd et al., 2011; Barrett et al., 2007).

Language is central to the human species as individuals, just like our organisation in society. It is highly interconnected but adaptable. We can see how *modelling the language faculty* can be one of the most insightful and exciting scientific endeavours. It is, however, also highly challenging. For a long time, rule-based systems – though vigorously constructed – could not capture the described complexity (see Section 2.1.2 for a short history of language modelling). It is just recently that language models

came into their own: due to the broad adaptation and scaling of distributed machine-learning approaches (Hernandez et al., 2021; Ghorbani et al., 2022; Kaplan et al., 2020; Hoffmann et al., 2022) in the past years, progress on their capabilities has been rapid. Now, language modelling technology is at an inflection point. The abilities of state-of-the-art models are close to the capacities of the human language faculty. It is now that language models become highly auspicious from a theoretical viewpoint.

1.2.2 Why a holistic style of analysis?

As stated in the Overview (1.1), a feature of my work is that its methods tend to be holistic. What motivates this decision? The choice for holistic methods stems from the realisation that the utility of reductionist approaches has limits in language research and – in extension – the analysis of language models. To elaborate on this insight, I will first briefly define reductive science, complex system theory and the criticism of the latter on the former. Then, I will delimit to which extent language and language models are subjects of research that conform with this criticism.

Reductive science explains complex phenomena by breaking them down into constituent parts, analysing the constituents' properties and how they relate to each other. It is based on the assumption of linear relationships and the additivity of the component parts of a subject (Sapolsky and Balt, 1996). Reductionism posits that it is possible to eliminate any variability by increasing the granularity of the constituents. Ultimately, reductionism assumes closed systems that are non-adaptive. This means that they are not influenced by any factors external to the analysed system and, hence, do not change in response to interactions with the external environment (see e.g. Holland, 2000).

Reductive science has been criticised when applied to analysing dynamic and complex systems by proponents of what can be subsumed as *complex adaptive systems theory (CAS)*. CAS has probably first been described by Mill (1856). Opposing the reductionist approach, CAS assumes that complex systems are made up of inseparable subsystems (interconnectedness; Simon, 1962). These interconnected subsystems often interact,

leading to behaviours or properties that are not easily predicted by examining each subsystem in isolation. This tightly connects to the idea of emergentism (O'Grady, 2008), which states that complex systems can have emergent properties that go beyond the sum of their parts. A precondition for emergent properties is the system's self-organisation that creates structures or behaviours without external control or central coordination (Cameron and Larsen-Freeman, 2007). This ability for self-regulation allows for adaptation to influences from outside the system (Cameron and Larsen-Freeman, 2007). One of the most illustrative properties of CAS is that systems can contain non-additivities and non-linearities. The resulting, potentially chaotic, behaviour suggests that minor changes in a component part can lead to unpredictable changes in the system outputs (May, 1976; Feigenbaum, 1980).

When considering the study of language, the reductive approaches – for example, in the tradition of Chomsky (1957, 1965) – have led to great insights into many regularities of human language. They achieved this by searching analytically for basic universal principles and rules of language construction (De Lacy, 2007; Hippisley and Stump, 2016; den Dikken, 2013; Aloni and Dekker, 2016; Gutzmann, 2020). However, the reductive linguistic theories are usually less successful when describing the (among others) interactive (e.g. pragmatics), dynamic and adaptable (e.g. language change), and open (e.g. variation in language use) aspects of human languages (Sperber and Wilson, 1986; MacWhinney, 2001; Clyne, 2003; Trudgill, 2019). A linguistic theory following the reductionist agenda has to remain within its self-delimited range as language in the real world is dynamic. As soon as researchers attempt to increase the ecological validity of their work (e.g. because they conduct applied research), they frequently have to recognise the more entangled nature of their subject (Cameron and Larsen-Freeman, 2007; O'Grady, 2008; De Bot et al., 2007, see Hensley, 2010 for an overview).

Language models (LMs) were initially based on reductionist insights: Rule-based systems used knowledge about, for example, language construction rules from generative grammar to reverse-engineer automatic language generation systems. With the switch to machine learning meth-

ods, LMs diverged from the reductionist path and now have to deal with the real-world complexity of language. Distributed machine learning methods like neural networks embrace interconnectedness and dynamism (Marsland, 2011). Recent developments in machine-learning-based LMs have led to an astonishing ability to resemble human language. As a consequence, LMs became complex in themselves while their interpretability decreased. An illustrative example of the complex properties of modern LMs can be seen in the work of Khashabi et al. (2022), who show that minimal perturbations in the input space of an LLM can result in entirely divergent behaviour in the output space, which is evidence of highly nonlinear behaviour. Reductionist attempts at the explanation of LM behaviour may still give valuable insights into the early processing stages of a model but usually find it unexplainable with increasing depths of processing (e.g. Elhage et al., 2021). The language models I investigate in this dissertation are machine-learning-based distributed systems. Therefore, the methods tend to be holistic rather than purely reductionist.

1.2.3 Why learning dynamics?

As we have seen in the previous section, language in its entirety is a complex system. At the same time, almost every human acquires with relative ease at least one language within their lifetime and does so from comparatively little exposure (Hart and Risley, 1995; Hoff, 2003; Huttenlocher et al., 2010; Rowe, 2012; Weisleder and Fernald, 2013; Gilkerson et al., 2017). For a long time, this was explained through an innate, biologically determined linguistic architecture (Tucker and Hirsh-Pasek, 1993), which enables rapid acquisition and – as a side effect – caused the many congruences across human languages due to its alleged constraints (Chomsky, 1957, 1965). Recently, LMs based on machine learning techniques, especially since the advent of the transformer-architecture (Vaswani et al., 2017), have shown that no language-specific inductive bias is necessary to learn human-level language abilities. While these architectures excel at acquiring human language, they are similarly good at other learning objectives (e.g. in computer vision; see Carion et al., 2020; Wang et al.,

2018; Parmar et al., 2018; Dosovitskiy et al., 2021). Hence, it is possible to learn natural language with universal function approximators like neural networks (Hornik et al., 1989; Csáji et al., 2001).

These developments spawn many exciting questions about the learning process: How does this ability interact with the particular structure of human language? How much do generalisations of LMs correspond to regularities in theoretical linguistics? These questions may open a new window to understanding language and learning processes in distributed systems. However, they remain largely unaddressed and underinvestigated by interpretability research.

1.3 Research objectives (RO)

Generally, this dissertation studies the relationship between data properties and generalisation behaviour in language models. More specifically, the objectives can be summarised as follows:

RO 1. Connect domain knowledge with learning dynamics
Human domain knowledge is based on similarities: If two things are similar in some defining property, we cluster them into the same concept. In this dissertation, I aim to connect human conceptualisation with the learning of language models, which similarly exploit statistical regularities to learn efficiently. I develop a framework to evaluate which concepts a network generalises across and which ones it treats idiosyncratically.

RO 2. Derive 'synthetic linguistic theories' from language models
Just like humans (Watson, 1913; Titchener, 1912; Nisbett and Wilson, 1977), language models have the problem of not being able to reliably introspect their inner processes. Consequently, we must find alternative ways to interface them if we want to learn about their abilities. I aim to use the above framework to derive 'synthetic linguistic theories' (Chowdhury and Zamparelli, 2019) from language models in a form similar to Gardenfors (2004)'s conceptual spaces.

RO 3. Investigate generalisation throughout the learning process
Generalisation behaviour might not be the same throughout the learning process of a language model. Techniques like curriculum learning suggest that learning processes are sequential: understanding more complicated data points builds upon established knowledge of more basic concepts. I aim to investigate how a model's conceptualisation of language changes throughout the training process and whether we can use our insights about the learning process to influence the learning outcomes.

RO 4. Investigate failed generalisation
In some cases, the LMs generalisation does not follow the causal structure of the process that generated the data in 'the real world' (Schölkopf et al., 2012). Instead, they find other (spurious) regularities in the data that they then latch on to, potentially producing unreliable outputs in previously unseen contexts. I aim to investigate whether there are patterns in the brittleness of the new learning paradigm of *in-context learning*. In-context learning shows seemingly chaotic behaviour in response to certain inputs. Holistic methods appear especially sensible in this learning paradigm.

Beyond these objectives, a desideratum of all research work in this dissertation is to employ methods as holistic as possible (as it was motivated in Section 1.2.2).

1.4 Structure

Before presenting the original research work of this dissertation, a general background section in Chapter 2 will familiarise the reader with any potentially unknown but relevant concepts that I will build upon. The subsequent main body is structured into four chapters. Each chapter contains a self-contained research project that has been or will be published in one or multiple research papers. The content of the original publications has been adapted to make this dissertation more cohesive. The research

objectives of the previous section map on either one or multiple chapters. The chapters of the main body will engage with the following desiderata:

Chapter 3 Create a framework to link formal linguistic theory with generalisation in language models (RO 1.).

Chapter 4 Apply the framework from Chapter 3 to an extensive range of linguistic phenomena and create linguistic similarity spaces, link it more directly to the learning signal (gradients) and use it to analyse the change of a language model's conceptualisation of language throughout training (RO 1., 2. and 3.).

Chapter 5 Use an automated curriculum learning technique adapted to language modelling to examine the potential sequentiality of language learning in LMs (RO 2., 3.).

Chapter 6 Explore differences in robustness between learning with and without parameter updates and try to understand data properties that cause inconsistencies in model predictions in the latter (RO 4.).

The chapters of the main body will follow the general structure of research papers in computational linguistics, in addition to more extensive introduction and conclusion sections. Ultimately, I will close with a general conclusion that revisits the original research objectives, evaluates their achievement, summarises the contributions and hints at future directions.

Chapter 2

BACKGROUND

Throughout the dissertation, I will build upon different concepts which might be more or less established in the field. The subsequent section is meant to familiarise the reader with potentially unknown ideas and subfields of research. I will refer back to the respective portions of this background section throughout the main body of the dissertation.

2.1 Language Models

This section familiarises the reader with the notion of language models, starting with a definition (Section 2.1.1) and a short history of language modelling practices (Section 2.1.2). Subsequently, I will review the literature to show why the language modelling objective is very well suited to serve as the basis for transfer learning (Section 2.1.3).

2.1.1 A definition of language models

A language model (LM) is a computational model designed to generate natural or formal language by learning patterns within a vast text corpus. Formally, an LM is a probabilistic model of natural and/or formal language (Jurafsky and Martin, 2000). It assigns a probability distribution

over a given vocabulary conditioned on a context. In the broader definition, the context can consist of data modality provided to the model (which includes, for example, pixels of an image; Wang et al., 2020; Li et al., 2022, 2023a; Rust et al., 2022)). While it is not uncommon to refer to these models as 'language models', in this dissertation, I concentrate on the narrower definition, in which the context consists exclusively of language data. In the narrow definition, for any sequence of tokens $S = \{x_1, x_2, ..., x_N\}$, the LM calculates the probability distribution over the vocabulary given the preceding words $P(x_i|x_{i-1}, ..., x_1)$. By sampling a new word x_i from the predicted distribution and subsequently concatenating the predicted word to the context and applying this operation recursively, an LM autoregressively generates text. The *probability of a sequence* is the joint probability of all tokens given their respective context $P(S) = \prod_{i=1}^{N} P(x_i|x_1, x_2, ..., x_N)$. The model is usually trained by iteratively minimising the negative log-likelihood of all sequences in a training corpus:

$$\mathcal{L} = -\sum_{i=1}^{N} \log P(x_i|x_1, x_2, ..., x_N)$$

Besides autoregressive models, a different subset of LMs — so-called *masked LMs* or MLMs — enjoyed popularity in the late 2010s and early 2020s (e.g. Devlin et al., 2019; Liu et al., 2019c). While there were many attempts at creating alternative LM objectives (e.g. Joshi et al., 2020; Lewis et al., 2020; Wang et al., 2019a; Liu et al., 2019b; Radford et al., 2018; Ramachandran et al., 2017; Dai and Le, 2015) the MLM objective was by far the most successful. MLMs differ from generative language models by reformulating the language modelling objective as a $|V|$-way classification task, where V is the model's vocabulary. They do so by predicting words in a sequence that previously have been masked out: Given a sequence of tokens $S = \{x_1, x_2, ..., x_N\}$, where N is the length of the sequence, they select a subset of tokens $M \subseteq S$, replace the tokens in M with a special [MASK] token to then minimise the negative log-likelihood of the

replaced tokens:

$$\mathcal{L}_{MLM} = -\sum_{x_i \in M} \log P(x_i | S \setminus M)$$

Chapters 3, 4 and 6 mostly investigate with generative LMs, while Chapters 5 and 6 also utilise MLMs.

2.1.2 A short history of language models

To put modern LMs and LLMs into perspective, I will briefly summarise the history of natural language processing with a focus on language models based on a small selection of milestones. The summary is split into two sections: *pre-neural machine learning* and *neural machine learning*, with the boundary between the two marked by the onset of the large-scale use of machine learning methods in the early 2010s. This shift, even though much more gradual than portrayed, was arguably the most profound for the field to date and, similarly, is the most relevant to this dissertation.

Pre-neural machine learning The earliest attempts at creating computational language processing systems took place in the 1950s and 1960s in the form of rule-based systems. Efforts like the Georgetown-IBM Experiment (Weaver, 1952) aimed to automatically translate academic Russian into English, albeit with limited success, as the system's capacity was limited to six grammar rules and 250 lexical items. Pure rule-based systems peaked with elaborate approaches like the chatbot ELIZA (Weizenbaum, 1966), which identified keywords in the user's input and used hard-coded rules to reformulate the input into a question to mimic conversation. While the ideas of *statistical* language models were already formulated by Markov (2006) and Shannon (1948), they only found more widespread adaptation in the 1970s. N-gram and hidden Markov models gained prominence, exploiting the statistical regularities in large language corpora. Refined versions of statistical models as well as hybrid models that integrated rule-based systems and statistical methods, remained prominent in the

field well into the 2000s (see Rosenfeld, 2000). The availability of large-scale language resources through the internet helped to improve 'purely statistical' approaches until neural machine learning algorithms eventually superseded them.

Large scale neural machine learning A significant shift in natural language processing and language modelling started with the increased attention to distributed learning systems such as recurrent neural networks and end-to-end training in the early 2010s (Mikolov et al., 2010; Mikolov, 2012). This shift marked a profound change in models: Up to this point, approaches embraced reductionist ideas like manual feature engineering and hand-crafted rules. However, the resulting language models suffered from limited generalisation, difficulty in dealing with sparsity in language and weak scalability (Rosenfeld, 2000; Bengio et al., 2000; Goodman, 2001; Chelba and Jelinek, 1998; Brown et al., 1992). The new generation of end-to-end learning models delegated feature and rule extraction to emergent processes, overcoming problems with generalisation, sparsity and scalability (Baroni et al., 2014). These developments came at the cost of control over the exact model behaviour and interpretability. The shift gained traction when the deep learning model AlexNet revolutionised the field of Computer Vision by beating its – at the time – most important benchmark by a far margin (Krizhevsky et al., 2012; Russakovsky et al., 2015). Shortly after, the power of implicit feature learning in NLP became apparent with the publication of word2vec (Mikolov et al., 2013a). From here, progress was driven by architectural innovations and scaling. Progress in architectural innovation has been relatively monolithic through the introduction of the transformer architecture (Vaswani et al., 2017), which emphasised the use of attention mechanisms (Bahdanau et al., 2015; Kim et al., 2017) and optimised training through parallel processing. Transformers were initially constructed for the problem of machine translation but later came into their own as the architectural basis for language models (Devlin et al., 2019; Liu et al., 2019c; Brown et al., 2020; Touvron et al., 2023). On the other hand, the importance of scaling the training data (Halevy et al., 2009), model parameters and computation for model

training became very clear. A first glimpse at the power of scaling was given at the introduction of word2vec (Mikolov et al., 2013a,b) and, later became formalised through the discovery of scaling laws for transformer-based models (Hernandez et al., 2021; Ghorbani et al., 2022; Kaplan et al., 2020; Hoffmann et al., 2022). Ultimately, setting the focus on decoder-based, generative models instead of encoder-based, discriminative models (MLMs; see Section 2.1.1) yielded a two-fold advantage: First, scaling with generative models is more efficient, as parameter updates are based on the loss of each token in a sequence instead of only the masked subset M, resulting in a more robust learning signal. Second, generative and discriminative language models (LMs) pursue distinct learning goals. Generative models strive to grasp the entire data distribution, enabling them to produce new data instances (or tokens). Conversely, discriminative models primarily focus on identifying a decision boundary to distinguish between various classes (or tokens). This means that generative models assess the joint probability distribution $P(X, Y)$, which can later be converted to $P(y|x)$. Discriminative models, meanwhile, target the conditional probability $P(y|x)$ directly. While capturing $P(X, Y)$ can be more challenging initially, it compels the model to create a comprehensive representation of the original data distribution, which may offer better adaptability to novel data from the same distribution.

2.1.3 Why are language models so effective?

Until recently – and to a certain degree still today – a major problem of NLP models is a lack of generalisation capacity. Models tended to find deteriorated short-cut solutions wherever possible (Shah et al., 2020; Geirhos et al., 2020; Niven and Kao, 2019). Those solutions perform well on their training distribution but fail when confronted with a slight distributional shift during testing (Hupkes et al., 2023; Kervadec et al., 2021; Teney et al., 2023; Wang et al., 2022; Tu et al., 2020). The robustness improved significantly when practitioners started to use language modelling as foundational models (see Hendrycks et al., 2019, 2020). Why are language models so apt as a base for transfer learning? In the following, I will briefly

lay out some intuition behind the effectiveness of language models.

As mentioned in Section 1.2.1 and 1.2.2, the process of language generation (i.e. humans speaking or writing) is highly complex due to its interconnection with large amounts of external factors. When a large training corpus of natural language is collected from a diverse set of sources, this conditionality on external factors is reflected in the data: The data will necessarily be diverse (span a variety of domains, topics, styles and so on) and entail broad foundational knowledge (Etzioni et al., 2008; Michel et al., 2011). With increased training set diversity, spurious correlations are effectively countered (Kaushik et al., 2020; Weber et al., 2021) and to achieve low empirical error, a language model has to more closely model the underlying structure of the human language production (Schölkopf et al., 2012). More formally, the large scale of modern LMs induces a vast hypothesis space $\mathcal{H}$ (Vapnik, 1982), which then is pruned through the inductive bias that the model receives through pertaining (Baxter, 2000). This over-parameterisation, coinciding with a strong inductive bias, enables the model to capture many aspects of language-relevant tasks. Empirical evidence (Raffel et al., 2020) further suggests that the generative LM objective is the most performant among many proposed alternatives (Joshi et al., 2020; Lewis et al., 2020; Wang et al., 2019a; Liu et al., 2019b; Radford et al., 2018; Ramachandran et al., 2017; Dai and Le, 2015) when controlled for computational cost.

2.2 Learning in neural networks

Learning in machines is substantially different from human learning. Just like human learning, learning in neural networks can be systematised into different types of learning. Different theoretical approaches have been formulated to understand learning dynamics better. This section introduces some concepts in the theory of neural network learning.

2.2.1 Training approaches

The following introduces different approaches to neural network training relevant to this dissertation and sketches out their impact on learning dynamics.

Supervised and self-supervised learning In machine learning, the term 'supervision' refers to the nature of the data from which a model learns (Bishop and Nasrabadi, 2006). Learning problems in which every input data point has a ground truth target label are called **supervised learning**. A model is optimised to predict the target from the respective input. A major consideration in supervised learning problems is the balancing of the bias/variance trade-off (Geman et al., 1992; James, 2003), which roughly states that the practitioner has to ensure that the supervised learner neither overfits nor underfits the input-to-target mapping of the data (Everitt, 1998). To balance the bias/variance trade-off, practitioners have to estimate the complexity of the function they want to learn and match it with a sufficient amount of training data, consider the dimensionality of their input features and match them with an appropriate model complexity. Alternatively, the practitioner can resort to mediation techniques such as regularisation (Bickel et al., 2006). Addressing the bias/variance trade-off is an essential prerequisite for generalisation beyond the training distribution. Supervised tasks often struggle with data availability since data collection tends to be manual and expensive. As a consequence, in modern NLP datasets of labelled data are usually not used as a primary training set but rather as a target for transfer learning from self-supervised models (see following paragraph) or as diagnostic benchmark tasks to estimate and compare capacities across models (Wang et al., 2019c,b; Socher et al., 2013; Rajpurkar et al., 2016; Zhang et al., 2019, among many). The bias/variance trade-off, more specifically the issue of overfitting, is an important notion in non-robust generalisation (Kavumba et al., 2019; McCoy et al., 2019; Niven and Kao, 2019), a concept important for the presented work in Chapter 6.

Another type of supervision relevant to this dissertation is **self-supervised**

learning. Self-supervised learning is a mixed form of unsupervised learning – where there are no external ground truth labels – and supervised learning. Instead of using external labels, in self-supervised learning, labels are generated by the learning algorithm itself. This can be achieved by predicting missing (or 'masked') parts of an input or predicting the following items in a sequence in an autoregressive way. The great advantage of self-supervised training is its ability to leverage vast amounts of unlabeled data, making it easy to scale training. Both the prediction of masked portions of the input and autoregression are common ways to pre-train LMs (see Section 2.1.1). As laid out in Section 2.1.3, representation learning via autoregression has various advantages as the pre-training objective for language models.

Besides supervised and self-supervised learning, there are other types of supervision (such as unsupervised learning, semi-supervised learning, etc.) with partially fuzzy definitions. I will refrain from further detailed explanations as those are less relevant to this dissertation.

Gradient-based and reinforcement learning An important characteristic of a learning problem is its differentiability, i.e., whether a prediction error can be related to the model parameters via calculating the partial derivative of the error with respect to the model parameters. For a function to be differentiable, it must at least be continuous at that point (Rudin, 1953). However, continuity does not always guarantee differentiability. Furthermore, smoothness, which usually implies that a function has continuous derivatives up to a certain order, provides a stronger foundation for differentiability (Adams and Fournier, 2003).

In most use cases of natural language processing, differentiability is given. In those cases, *gradient-based methods* can optimise model parameters directly. This is done by iteratively calculating the partial derivative of the prediction error with respect to the model parameters and then updating model parameters in the direction of the steepest descent of the error (Salakhutdinov, 2014). For deeper models that implement a cascade of parametric operations on the input, updates can be calculated by chaining derivatives through each parameter layer, a technique commonly

known as backpropagation (Rosenblatt et al., 1962; Rumelhart et al., 1986).

On the other hand, many real-world applications require interactions with an external environment that is not part of the learning system. In this case, the differentiability of the whole system is often not given, and the practitioner has to resort to so-called reinforcement learning (RL), which is not subject to this constraint. Instead of minimising empirical error as in supervised and unsupervised learning, in RL, the focus lies on maximising cumulative rewards. This leads to interesting differences in the learning dynamics of RL systems. In conventional gradient-based learning, you must know the optimal strategy to compute the prediction error. In RL, this is not the case. Instead, the optimal strategy has to be discovered by the learner. The emphasis on rewards, rather than errors, influences the learning process. For instance, there's an increased significance in striking a balance between *exploration* (trying out new actions) and *exploitation* (leveraging known beneficial actions). Historically, RL has not been widely adopted in natural language processing. This is attributed to several challenges in the optimization process. These include issues like high sample complexity, instability arising from sensitivity to the reward structure, and sensitivity to hyperparameters (Vapnik, 1999). RL in NLP can also be hampered by sparse rewards and difficulty in designing suitable reward functions (Dulac-Arnold et al., 2019). Despite this, RL has recently found entrance in the training of state-of-the-art LMs in the form of reinforcement learning from human feedback (RLHF; Stiennon et al., 2020; Glaese et al., 2022; Ouyang et al., 2022). Despite its cost (and some inherent problems to the training process; Casper et al., 2023), RLHF allows more precise optimization of LMs based on human preferences, resulting in learning outcomes that are unparalleled by other methods. Up to this date and my best knowledge, there is no formal explanation for the advantage of RLHF for alignment over standard supervised learning (for an attempt at an explanation, see Eysenbach et al., 2020), even though the topic is likely to receive the attention of active research in the nearby future.

Learning with and without parameter updates A dichotomy that emerged recently is learning with versus learning without model parameter updates. While all classical parametric approaches in machine learning update a model's internal parameters to fit a target function, learning without parameter updates – also called *in-context learning* (ICL) – has recently become a new important learning paradigm. ICL describes the reduction in per-token-loss with an increase of their indices in the input sequence (or in other words: the later a word in the input sequence, the lower its loss; Kaplan et al., 2020). Essentially, the model leverages information given in the context to adjust its predictions and reduce prediction errors. As this property becomes more pronounced in recent models, it can be used for interesting purposes. For example, a model can develop the ability to infer a task from its input and condition its output accordingly to solve it. This ability was first recognised in GPT2 (Radford et al., 2019) and evoked larger interest in GPT3 (Brown et al., 2020).

ICL is an emergent property that only arises at a certain model scale (Wei et al., 2022b). A recent large-scale analysis in Lu et al. (2023) shows how most of the impressive abilities of LLMs are due to their capacity to do in-context learning. Currently, the mechanisms of ICL are not fully understood. However, there are three prominent hypotheses about the reasons for this emergent ability: associative memory (Ramsauer et al., 2021), induction heads (Elhage et al., 2021; Olsson et al., 2022) and mesa-optimisation (Hubinger et al., 2019), with the idea of mesa-optimisation receiving the most attention. Mesa optimisation describes an advanced stage of self-organisation. It states that in larger models, mesa-optimisers emerge within the model parameters (Hubinger et al., 2019). Mesa optimisers are internal optimisers that learn simple, temporary functions from the input. Garg et al. (2022); Akyürek et al. (2022); Li et al. (2023b) show how in-context learners can implement standard finetuning algorithms implicitly, while Von Oswald et al. (2023) provide evidence that IC learners implicitly implement gradient descent during inference. The different hypotheses are not mutually exclusive and come to similar conclusions about the phenomenon of ICL (Von Oswald et al., 2023). A recent review of the latest research can be found in Dong et al. (2023).

2.2.2 Theories of Learning

There are different ways to think about learning and generalisation in machine learning and psychological research. I here present the ideas most influential to the research that makes up this dissertation.

Generalisation-focused research At its core, generalization research seeks to understand how computational models can apply learned knowledge to previously unseen data or evaluation setups. 'Good generalisation', however, is a fuzzy concept, entailing many potential scenarios. Traditionally, generalisation was tested through a model's performance on a previously unseen set of data that is i.i.d. to the training distribution. However, within recent years, it became increasingly clear how this type of generalisation does not guarantee model quality: Models may rely on simple heuristics that do not generalise beyond the i.i.d. distribution (Kaushik et al., 2020; Gardner et al., 2020; McCoy et al., 2019), rely on stereotypes (Parrish et al., 2022; Srivastava et al., 2022) or memorisation of the pre-training data rather than genuinely generalising (Lewis et al., 2021; Razeghi et al., 2022). Since then, evaluation methods have grown more fine-grained and elaborate. Hupkes et al. (2023) provide an excellent and comprehensive overview and systematisation of the field. They create a taxonomy of generalisation research, classifying it on five dimensions: *Motivation, Generalisation type, Shift type, Shift source, and Shift locus*. With this taxonomy, research becomes comparable and contextualisable. *Motivation* and *generalisation type* are the most important dimensions to frame my work in the main chapters. Here, I will explain the realisation of relevant dimensions to this dissertation.

Motivation: Research in this dissertation is mainly motivated *cognitively* and *practically*. The *cognitive* motivation can be described as either centred around benchmarking models or deriving hypotheses about the functioning of human cognition: one focuses on evaluating NLP models against human generalisation capabilities, given humans' unique and efficient learning and recombination skills; the other delves deeper into understanding human cognition and language through computational models,

aiming to derive insights about human generalisation rather than enhancing the models themselves. The *practical* motivation aims at assessing whether a model is accurate and reliable in a specific type of application. Chapters 3, 4 and 5 are motivated by both subtypes of cognitive motivation, while chapter 6 is additionally more practically motivated.

Generalisation type: Research in this dissertation investigates primarily *compositional*, *structural*, *cross-task* and *robust* generalisation. *Compositional generalisation* is defined as 'the ability to systematically recombine previously learned elements to map new inputs made up from these elements to their correct output' (Schmidhuber, 1990). *Structural generalisation* is related to compositional generalisation but is less focused on the output space. *Cross-task generalisation* investigates whether a model can exploit shared structure across different learning objectives and their related data distribution shifts. While traditionally concerned with very explicit simultaneous multi-task learning settings, the sequential pretraining-finetuning paradigm in NLP, as well as the in-context-learning-based prompting paradigm, can be considered multi-task learning in the broader sense. *Robust generalisation* implies that a model performs well on new and unseen data but remains resilient against various challenges, anomalies, or adversarial attacks it might face in real-world scenarios. To generalise robustly, a model has to more closely model the underlying structure of the real-world process that generates the original data we are modelling (Schölkopf et al., 2012) instead of latching on superficial clues like spurious correlations. Chapters 3 and 4 investigate *structural* and in its sense *cross-task* generalisation, while Chapter 5 aims to investigate compositional generalisation and Chapter 6 researches *robustness*.

Learning as data compression Machine learning and information theory have long been considered to be closely linked (Solomonoff, 1964; Rissanen, 1978; MacKay, 2003). Especially the view that learning efficient representations of data and (loss-less) compression optimise the same underlying objective. The connection is intuitive: In the act of information-theoretic compression, we seek patterns and regularities that we can then exploit to reduce redundancies in the description of the data. Similarly, in

learning, especially in unsupervised learning, the goal is to discover underlying patterns or structures in the data to learn efficient representations of their distribution.

This connection bears valuable frameworks for thinking about learning in machine learning. Probably most prominent is the information bottleneck method (Tishby et al., 1999), which can be used to describe learning as optimising the trade-off between minimising the mutual information of an input X and some model internal representation T while maximising the mutual information of the same representation with some target Y (given a joint-probability distribution $p(X, Y)$; Tishby and Zaslavsky, 2015). Later work suggested that the different parts of the bottleneck are optimised in different phases of learning (first maximising accuracy – i.e. increasing $I(T, Y)$ – and compressing representations afterwards – i.e. decreasing $I(X, T)$; Shwartz-Ziv and Tishby, 2017). While the universality of these findings is contested (Saxe et al., 2018; Noshad et al., 2019), it remains an idea that productively frames and inspires the discussion about learning.

Considering language, recent research has shown how classification using general-purpose parameter-free compressors such as *gzip* (Gailly and Adler, 1992) can achieve similar performance to expensive representations of parametric deep learning models such as BERT (Devlin et al., 2019) when used as a base for sequence classification tasks (Jiang et al., 2023). Further, Delétang et al. (2023) shows how LLMs trained exclusively on language can function as general-purpose compressors for data of other modalities such as image or speech data. In that sense, creating any LM (whether modern LLMs or the construction of a generative grammar) can be understood as an attempt to find rules for efficient compression of a language context.

The information-theoretic compression described so far has recently been linked to geometric compression in language models (Cheng et al., 2023). Geometric compression is based on the manifold learning hypothesis (Salakhutdinov, 2014), which states that high-dimensional data (e.g. the representations within an LM) can be represented on a low-dimensional manifold. The number of dimensions of this manifold is also called the

intrinsic dimensionality. It can be estimated by dimensionality reduction methods such as PCA (Pearson, 1901; Hotelling, 1933; Wold et al., 1987) under the assumption of flat manifolds (Campadelli et al., 2015) or various non-linear methods for curved manifolds (Grassberger and Procaccia, 1983; Levina and Bickel, 2004; Haro et al., 2008; Albergante et al., 2019). The notion of geometric compression has already found impactful applications in machine learning (e.g. in low-rank adaptation; Hu et al., 2022).

The idea of learning algorithms as compressors is not explicitly used in the empirical work presented in this dissertation but informed the work and will be used in the respective discussions.

2.3 Interpretability

In Sections 1.2.2 and 2.1.2, I outlined the transition from reductionist methods to neural network techniques in NLP. This shift led to a significant increase in the complexity of LMs and a corresponding decrease in their ad-hoc interpretability – i.e. our ability to intuitively understand a model's behaviour and inner workings. Consequently, increasingly sophisticated methods became necessary to disentangle the inner processes of LMs. The research field focusing on the development and assessment of such methods is called interpretability or explainable AI. The relevance of interpretability in the field has increased exponentially with the onset of neural network methods (see Figure 2.1). Why did interpretability become so prominent?

Motivation Interpretability can be motivated from two different angles: First, from an applied angle, one can evaluate the model to ascertain whether the model's generalisations align with the practitioner's intentions. This might entail ensuring the model has grasped accurate language construction rules (e.g. Goldberg, 2019), operates without inherent biases (e.g. Bolukbasi et al., 2016; Caliskan et al., 2017), or avoids suboptimal shortcuts in inference tasks (e.g. Ribeiro et al., 2016). This viewpoint is

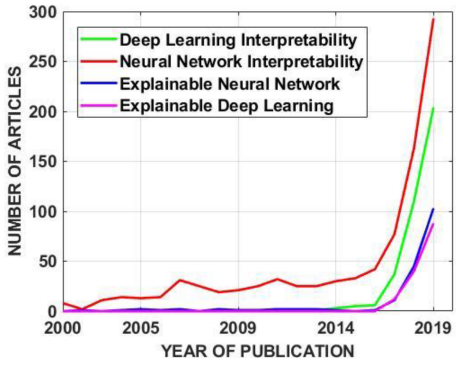


Figure 2.1: Publications on the topic of interpretability and explainability over the years (graphic from Fan et al., 2021)

predominant and primarily driven by the need to foster trust (e.g. Doshi-Velez and Kim, 2017), facilitate model debugging (e.g. Olah et al., 2018), and address ethical and regulatory considerations (e.g. European Parliament and Council of the European Union, 2016; Madiega, 2021). Second, interpretability can be used as a tool to derive insights about a particular subject or topic that the model is describing. Language models are models of the generative process of language, requiring many of the associated cognitive abilities. As such, they have become so proficient that they can be used for, i.a., linguistic theory testing in areas in which they verifiably employ parallel processing to humans (i.e. be used in a deductive way) or even induce theory in itself (e.g. Weber et al., 2021; Baroni, 2022).

Types of methods From a technical perspective, research methodologies in this domain predominantly fall into two categories: *structural* and *behavioural* methods. Structural methods analyse the internal properties of models, which might involve examining the weights, neuron activations, or attention mechanisms. On the other hand, behavioural methods aim to understand a model based on its observable actions in response to specific inputs. The goal is to infer the model's internal logic and capacities based on its behaviour. Overall, interpretability methods have

similarities with approaches in experimental psychology and different branches of neuroscience (such as behavioural neuroscience and systems neuroscience), which try to infer the black-box behaviour of the human brain via behavioural (e.g. reaction time experiments) or structural (e.g. imaging techniques) methods. It is, therefore, not uncommon to see avid cross-pollination in either field (e.g. see McCoy et al., 2019; Linzen et al., 2016; Richards et al., 2019; Yamins and DiCarlo, 2016; Marblestone et al., 2016; Kriegeskorte and Douglas, 2018).

Chapter 3

GENERALISATION AND LINGUISTIC THEORY

This first chapter of the main body introduces a framework to analyse the learning dynamics of language models and empirically verifies it. It lays the basis for later work in Chapter 4.

3.1 Introduction

In this chapter, we introduce and empirically test a framework for understanding learning and generalisation in language models. The framework is based on the idea that human learning is highly interleaved, meaning that many different objectives are optimised simultaneously and not in a strictly sequential manner. By learning from different sources at the same time and exploiting their commonalities, humans can form more general rules about the world, which in turn helps them to subsequently acquire new knowledge faster (Perkins et al., 1992; Schwartz et al., 2005; Cormier and Hagman, 2014; Luriia, 1976). The idea of capitalising on the commonalities of diverse tasks for more abstract generalisation is also very well known to the machine learning community and is most explicitly realised in the subfield of multi-task learning (MTL; Caruana, 1993, 1997). In MTL, multiple tasks are optimised jointly, enabling the transfer of

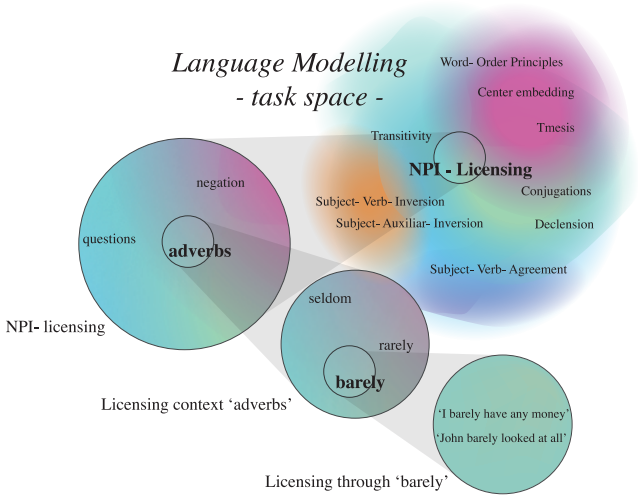


Figure 3.1: A conceptual visualisation of a language modelling task hierarchy, from language modelling as a whole to single examples, with complex similarities between tasks. Colours indicate task similarities.

relevant information across tasks. MTL research yields fruitful results in both application (e.g. Collobert and Weston, 2008; Collobert et al., 2011; Zhang et al., 2014; Donahue et al., 2014; Kaiser et al., 2017) and theory (e.g. Baxter, 2000; Maurer, 2006; Ando and Zhang, 2005; Argyriou et al., 2007). In the proposed framework, we study language modelling as an MTL problem.

In a stricter definition, language modelling does not qualify as MTL. However, both have many commonalities in their learning dynamics. While in MTL, a model has to optimise multiple explicit loss functions, we argue that a language model has to optimise many diverse tasks simultaneously as well. In that sense, language modelling is a conglomerate of many different tasks. For example, different language construction rules have to be learned at the same time. These rules may have more or less overlap in their structure and may be more or less contradictory. However, they all need to be learned to achieve the greater goal of producing acceptable language.

The main difference between our 'implicit' MTL problem and the 'explicit' counterparts is that our tasks are not externally defined but emerge during learning and are conditioned by the broader objective of language modelling (compare reductionism and complex systems in Section 1.2.2). This means that the tasks a model optimises (such as learning a specific linguistic rule) might change throughout the learning process and can be hierarchical (e.g. task: *learn to produce acceptable language* $\longrightarrow$ subtask: *learn language phenomenon* $A \longrightarrow$ sub-subtask: *learn realisation* A_1 *of language phenomenon* A; see Figure 3.1 for an illustration). A potential task hierarchy that could be emerging is given by formal linguistic theory.

On the other hand, if we want to analyse the learning dynamics of a model, our 'implicit' setting comes with much fewer assumptions. Since we consider the task-organisation to be emergent and self-organising, much of the implementational overhead of 'explicit' MTL is no longer necessary: There is no need to decide which tasks to train together (e.g. Bingel and Søgaard, 2017; Standley et al., 2020a); at which hierarchy-level to allow tasks to interact (e.g. Søgaard and Goldberg, 2016); which degree of parameter sharing to employ (Ruder, 2017); which mixture of training data to employ (e.g. Luong et al., 2016), and so on. We think that the analysis of learning dynamics in this non-disruptive way is more natural, as it allows for the self-organising task structures in the complex system of language to take hold (e.g. gradually allowing for other optimisation targets to emerge with increasing skill) and is not biased by the many arbitrary decisions that go into, e.g. the highly constructed learning scenarios of explicit MTL.

Why do we want to study language modelling as an MTL problem? MTL gives us an idea of the way that different learning tasks are interacting. In MTL, similar tasks will help each other during learning (hence facilitating generalisation). In contrast, dissimilar tasks will not interact or harm each other (e.g. Thrun and O'Sullivan, 1996; Passos et al., 2012). What exactly defines 'task similarity' is not universally defined. Using this notion of similarity, we build our bridge to conceptual knowledge from linguistic theory: Linguistic conceptualisation can be considered a prediction about the structural similarity of different utterances. In other words, we find regularities across different utterances and categorise them

into the linguistic phenomenon. The utterances are then considered similar with respect to this phenomenon. In that sense, linguistic theory makes predictions about what data a model might generalise across. Based on our framework, the model's generalisation behaviour can be interpreted in one of two ways:

The model generalises within linguistic concept (a) The linguistic concept is supported by computational modelling; (b) the model picked up on the concept.

The model does not generalise within linguistic concept (a) The linguistic concept is not supported by computational modelling *or*; (b) the model was not able to learn the concept (e.g. because of a lack of expressivity or data).

We see how the framework connects linguistic theory and the learning dynamics in language modelling and might contribute to either field.

Outline Having set the principle framework in the introduction, we will empirically underline it in the remainder of this chapter. First, we will add the necessary basic background on MTL (Section 3.2.1), the subset of linguistic tasks we focus on (*Negative Polarity Items*, and discuss some related work in interpretability (Section 3.2.3). Then, in Section 3.3 and 3.4, we present our practical example and the respective empirical results that showcase the framework. In Section 3.5, we discuss our results and framework in light of their implications for interpretability, MTL and linguistic research. We conclude in Section 3.5.

3.2 Background

In this chapter, we bring together three strands of research: MTL, linguistics and interpretability. As a proof of concept, we focus on one specific complex subset of linguistic tasks: *licensing of Negative Polarity Items*

(NPIs). Below, we give a short overview of the most important characteristics of the three fields of interest. The background on interpretability complements the more general background in Section 2.3.

3.2.1 Multi-task learning

In MTL, multiple tasks are learned together to enable information transfer from one task to another. If the transfer is successful, the benefits might be threefold: the model learns tasks with less training data (i.e. *more efficient*, Collobert et al., 2011; Benton et al., 2017; Kaiser et al., 2017), up to a higher final accuracy (Collobert and Weston, 2008; Kaiser et al., 2017) and in a way that better generalises to new tasks (Baxter, 2000; Collobert and Weston, 2008).

Caruana (1993, 1997) and Ruder (2017) propose several different – but related – processes that might enable positive transfer: related tasks can provide additional training examples for each other on the features they share (called *'statistical data amplification'*), certain features might be easier to learn through one task than through another, but be helpful for both of them (called *'eavesdropping'*), and idiosyncratic features of single tasks can be averaged out, while more general features are reinforced (called *'attention focusing'*)[1].

However, positive transfer is not guaranteed; It is also possible that performance *deteriorates* due to interference between different tasks, resulting in negative transfer, Rosenstein et al. (2005); Pan and Yang (2010); Wang et al. (2019d). Whether the transfer is positive depends on the *task similarity* and whether the model can exploit this similarity (Rosenstein et al., 2005; Thrun and O'Sullivan, 1996; Passos et al., 2012).

The main goal of MTL so far has been to avoid negative- and encourage positive transfer by determining task similarity and regulating the interactions between tasks based on these similarities. Due to its pivotal role, much research effort was spent on determining similarities of tasks and the regulation of information transfer between them (for an overview,

[1]For a complete list of processes please consult the original publications.

see Zhang and Yang, 2017; Ruder, 2017). The disadvantage of these approaches is that assuming fixed tasks and regulating transfer between them based on fixed task similarities puts large constraints on possible transfers between tasks. It neglects the fact that learning processes are dynamic. In real-world learning, however, tasks – as well as their similarities – can change throughout the learning process. As an important difference to standard, 'explicit' MTL, we only use predefined tasks and their similarities to *analyse* the learning behaviour of the model without constraining the learning process in any way.

3.2.2 Negative Polarity Items

We exemplify our idea by analysing the learning behaviour on a complex subset of linguistic tasks: the licensing of Negative Polarity Items (NPIs). The properties of NPI licensing make it an interesting and adequate subset of tasks to study, as it has a high degree of complexity, an appropriate frequency within natural language (more detail on the distribution is given in Table 3.1) and was previously frequently investigated in neural models (see e.g. Warstadt et al., 2019; Jumelet et al., 2021; Marvin and Linzen, 2018; Jumelet and Hupkes, 2018; Futrell et al., 2019a; Hu et al., 2020).

NPIs are characterised by the property that they can only occur within the scope of certain *licensing contexts*. For instance, in the example below, the NPI '**any**' can occur in sentence (1)a., where it is in the scope of a negation, but not in sentence (1)b., where there is no licensor present.

(1) a. Bill did*n't* buy **any** books that day.
b. * Bill did buy **any** books that day.

(2) a. *Nobody* has **ever** been there.
b. * Somebody has **ever** been there.

Licensing contexts are formed based on semantic properties, such as downward entailment (Fauconnier, 1975; Ladusaw, 1980), non-veridicality (Giannakidou, 2011), or scope marking (Barker, 2018). Common licensing contexts include negation, conditionals, or superlatives and are often *triggered* by a specific expression, such as 'not' or 'nobody'. Grasping the

phenomenon of NPI licensing requires an understanding of three different aspects:

1. *The class of NPIs*: there is a group of expressions that are restricted in their occurrence.
2. *Licensing contexts*: there exists a group of expressions that allow NPIs to occur.
3. *Scope and structure*: the licensing contexts have to stand in a specific structural relationship to the NPIs.

We focus on how LMs learn the second aspect by analysing how different licensing contexts interact and generalize throughout training. During learning, they should be able to exploit their similarity in the other two aspects.

3.2.3 Linguistic interpretability of LMs

Interpretability research on LMs has shown that in pre-trained models, such as BERT (Devlin et al., 2019), hierarchical structure emerges throughout the layers and that this structure demonstrates parallels with linguistic theory (Peters et al., 2018; Liu et al., 2019a; Tenney et al., 2019). However, the way that this structure emerges during the learning process has not been investigated.

Considering more specifically the work on NPIs, research has shown that LMs can understand NPI licensing in recent years. Jumelet and Hupkes (2018) evaluate the performance of LMs on data sets containing NPI constructions extracted from large corpora, and Marvin and Linzen (2018); Wilcox et al. (2019); Warstadt and Bowman (2020) test them on artificial data sets containing template-based NPI constructions. In the experiments in this chapter, we will utilise the extensive template-based NPI corpus of Warstadt et al. (2019). Within this chapter, we add another dimension to the stack of interpretability research on NPI by showing how models acquire this ability.

3.3 Approach

We consider two different types of experiments. First, to understand to which extent models can understand and use the similarity between different licensing contexts (our *tasks*) during learning, we exploit the effect that the frequency of the different contexts has on learning. Second, we manipulate the LMs' training corpus to constrain their ability to leverage information from other licensing contexts during learning. In accordance with the MTL literature, we expect the LMs to learn tasks more data-efficient and to a higher final accuracy if they can leverage information across contexts. Before we describe our experiments in more detail, we present our model architecture and training, the evaluation procedure of the licensing contexts, and the filter procedure we use to manipulate the training corpus.

3.3.1 Model

Following previous work in this area, we consider recurrent language models. We focus on uni-directional LSTM models and mirror the hyperparameter setup of Gulordava et al. (2018)[2]. We train the models on the corpus provided by the same authors[3] – a subset of the English Wikipedia – or modified versions of the same for our second experiment (see Section 3.4.2). To track the learning process, we save models every 100 batches of training (i.e. 371 model checkpoints per training epoch). For all experiments, we average performance across five random seeds.

[2]Hyperparameters: batch size = 64, BPTT length = 35, dropout = 0.1, adaptive SGD learning rate = 20, layers = 2, hidden and embedding size = 650, epochs = 40.

[3]`https://github.com/facebookresearch/colorlessgreenRNNs/tree/master/data`

3.3.2 Evaluation

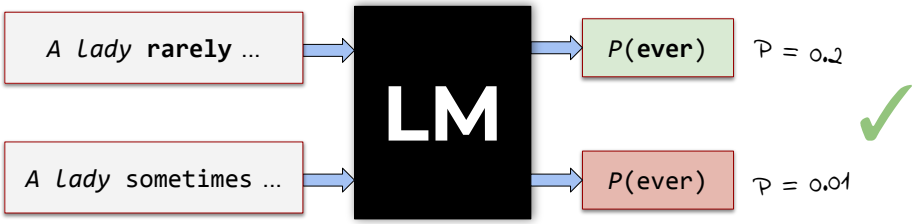


Figure 3.2: The NPI judgement task used for evaluating the LMs. A correct prediction assigns a higher probability to an NPI in a context that licenses it, based on the corpus of Warstadt et al. (2019).

To estimate the LMs' understanding of NPIs and their dependence on the different licensing contexts, we adapt the Cloze task of Warstadt et al. (2019), based on the implementation of Jumelet and Hupkes (2019). This task considers nine different types of licensing contexts (a list of the contexts, including examples, can be found in Table 3.1). For every such context, Warstadt et al. (2019) generated many *minimal pair sentences* containing correctly and incorrectly licensed NPIs. For instance, for the *adverbs* licensing context:

(3) a. A lady *rarely* **ever** thought that the children saw the boy.

b. * A lady *sometimes* **ever** thought that the children saw the boy.

Following previous work, we quantify an LM's understanding of a particular type of licensing context by computing the percentage of minimal pairs in that context for which the model correctly assigns a higher probability to the NPI in the licensing contexts than in the non-licensing contexts. I.e., in the example above, we would compare the probability the model assigns to the word *ever* in the contexts "A lady rarely" and "A lady sometimes" (see also Figure 3.2).

Context	Example	**Frequency** per 100k sentences
Simple Questions	Did he **ever** do a mean thing*?*	10
Adverbs	In the present political culture, there are *hardly* **any** leaders who would avoid limelight and refuse positions of power.	23
Questions	However, various writers attribute it to Putnam, Stark, Prescott or Gridley, while others question *whether* it was said **at all**.	25
Superlative	[...] and caused the *worst* winter flooding **in decades** for river and stream valleys [...].	32
Only	[...] "Those [students] *only* are supposed to pay **anything** who are abundantly able, or prefer to do so.	85
Conditional	In 1997 Li published a paper attempting to replicate <unk>'s results and showed the effect was very small, *if* it existed **at all**.	127
Quantifier	That's *all* you'll **ever** need.	179
Determiner negation	In spite of the <unk> of the disaster, *no* one was **ever** held accountable.	218
Sentential negation	It is *not* judged under **any** subjective points of view, only the clock.	712

Table 3.1: The nine types of licensing contexts are taken from Warstadt et al. (2019), with an example and the context frequency within the training corpus.

3.3.3 Identification of NPIs in training corpus

The Warstadt et al. (2019) corpus provides us with a task to evaluate nine different context types that license NPIs. To manipulate the training corpus for our experiments, we also need to identify sentences in the training corpus of the model in which these contexts licence NPIs. To do so, we need to locate these contexts, as well as establish that they, in fact, licence an NPI in a particular sentence.

We consider the nine context types of Warstadt et al. and the corresponding list of 30 expressions that are associated with these contexts (e.g. for the context type *adverbs*, the list of adverbs that are licensing NPIs). As for the NPIs, we consider an extensive list of 160 distinct NPIs[4], based on a collection provided by Hoeksema (2012). We then identify sentences in which an element of our NPI list is preceded by an element from our context list, ensuring that there is a dependency relation between them using the dependency parser of spaCy (Honnibal and Montani, 2017). When there are multiple potential licensors in a sentence, we use the hierarchical distance in the parse tree between the licensor and the NPI as a heuristic to find the correct licensor. By testing this procedure on a manually labelled set of 200 randomly selected sentences with multiple licensors, we estimate that it identifies the correct among multiple licensors in around 97% of cases. In Table 3.1, we report examples and frequencies of the different licensing contexts in the training corpus based on this filtering scheme.

3.4 Experiments and results

As a first step, we assess whether the LMs can adequately represent all nine categories of the evaluation task. To do so, we train five models on the regular training corpus and compute their final accuracy on our nine tasks. All models show adequate performance on most contexts (see Table 3.2), except the simple question context. Additionally, we observe that the models achieve their accuracy surprisingly fast: after two epochs,

[4]The complete list can be found in Appendix A.1.1.

there are no more substantial changes in empirical error (see Figure 3.3a). In the rest of our experiments, we, therefore focus only on these first two epochs.

Context	**Accuracy** ± std
Simple Questions	0.62 ± 0.05
Adverbs	0.92 ± 0.01
Questions	0.88 ± 0.03
Superlative	0.78 ± 0.03
Only	0.86 ± 0.04
Conditional	0.82 ± 0.06
Quantifier	0.86 ± 0.04
Determiner negation	0.92 ± 0.05
Sentential negation	0.85 ± 0.03

Table 3.2: Performance of the LMs on the evaluation task after 40 training epochs averaged over five runs.

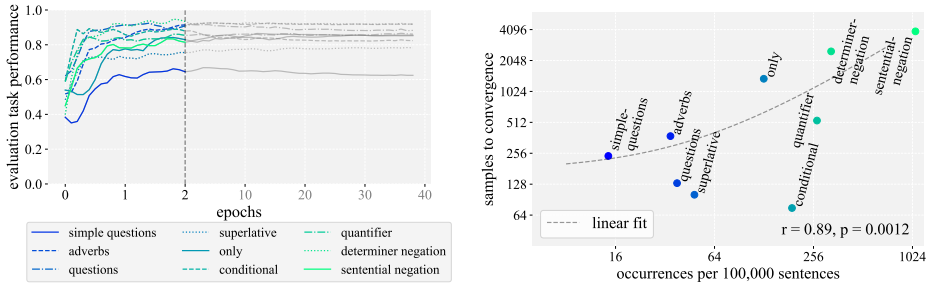
3.4.1 Frequency vs data efficiency

While some licensing contexts are rather common (e.g. negation), others appear scarcely as a licensor (e.g. adverbs). Therefore, throughout the learning process, the LMs encounter many instances of the more frequent contexts before they see an example of an infrequent context. If LMs could leverage information across contexts, less frequent contexts should thus have more prior established NPI understanding that they can bootstrap from. Consequently, the LMs should require fewer training examples to learn less frequent contexts than they need to learn more frequent contexts. In other words, the LM should be more *data efficient* for these infrequent contexts.

In our first experiment, we use this hypothesised relationship between frequency and data efficiency to assess whether LMs can exploit the similarities between different licensing contexts. To compare across different contexts, we quantify the data efficiency of an LM for a particular context

as the number of examples the LM needs to observe until it reaches 95% of its final accuracy for that context.[5] To make this measure more robust, we first apply a Savitzky–Golay noise filter to the learning curve (degree of polynomial = 1, window size = 25; Savitzky and Golay, 1964).

We compute the data efficiency of the trained LMs for all nine contexts and calculate the correlation between a context's frequency and the model's data efficiency concerning that context. In Figure 3.3b, we plot the average data efficiency of each context against the frequency of that context, as well as the linear fit that relates these two variables. The experiment demonstrates a strong relationship between the data efficiency and frequency of a respective context: $r = .89$, $p < .05$. Hence, the less frequent a licensing context is, the fewer examples are needed for the model to learn it, from which we conclude that the model can indeed transfer knowledge from previously acquired knowledge.



(a) (b)

Figure 3.3: (a) Average evaluation task performance. The performance rises steeply during approximately the first two epochs of training and afterwards levels off; (b) Data efficiency of nine different licensing contexts plotted against their frequency, averaged over five runs. The data efficiency is quantified as the number of training examples the model needs to observe to achieve 95% of the trained-out performance.

[5]The *more* data efficient, the *lower* this number thus is.

3.4.2 Transfer from general knowledge

While the presented relationship between frequency and data efficiency demonstrates that LMs can leverage previously learned information to learn less frequent licensing contexts, it does not unequivocally show that it leverages information from *other NPI contexts*. After all, when a less frequent context is encountered, the LM has not only had the opportunity to acquire prior knowledge about NPIs, it has also simply seen more language in general. In other words, the LM may meanwhile also have acquired more *general language knowledge*, which may help it to learn a less frequent licensing context more quickly. In our second experiment, we isolate transfer from general language knowledge and transfer from previously observed NPIs by training LMs on *single-context* corpora.

Single-context corpora contain NPIs licensed only by a single context. LMs trained on these corpora can thus not transfer knowledge acquired from other licensing contexts, as these are not present in the training data. By comparing the data efficiency of contexts between LMs trained on all-context and single-context corpora, we can thus infer how much of the increase in data efficiency for lower-frequent contexts is due to leveraging information from other contexts.

To create our nine single-context corpora, we use the procedure described in § 3.3.3 to identify all sentences containing NPIs licensed by our nine contexts. For every context, we create a corpus in which all sentences containing other contexts licensing NPIs are replaced by a neutral sentence of the same length, sampled from the rest of the corpus. During this replacement procedure, the ordering and composition of the corpus remained otherwise intact.

When we compare the learning of single-context with all-context models, we cannot rely on the previously used data-efficiency metric from Experiment 3.4.1. The data-efficiency measure is bound to how quickly the model reaches its final accuracy and benefits when its final accuracy decreases. As we expect the final accuracy to be lower in the single context models, comparing only data efficiencies between models is likely

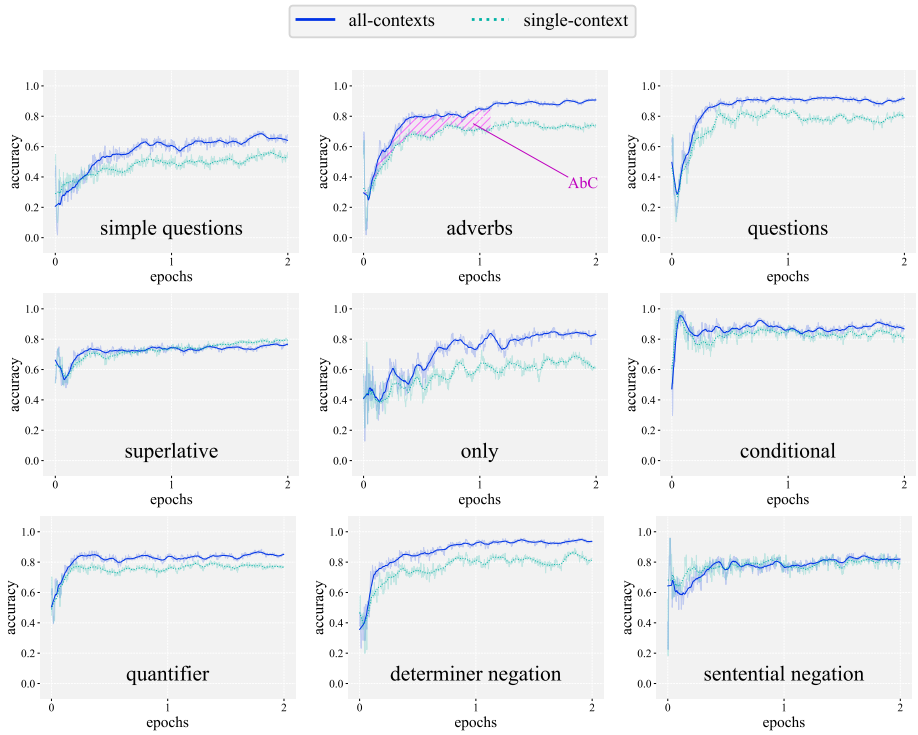


Figure 3.4: The LMs performance on different licensing contexts for the first two training epochs. We obtained these curves by evaluating all models at all 730 training checkpoints on the evaluation task.

to be uninformative.[6]. In this experiment, as explained below, we instead consider the area between the curves (AbC).

Area between Curves (AbC) incorporates both data efficiency and accuracy: for every context, we calculate the area between the all-contexts and single-context learning curves until the point in time where they both

[6]Consider, for instance, the extreme case in which an LM does not learn a particular context at all anymore in the single-context condition, as indicated by a chance accuracy of 0.5. Because it is not learning anything, the model would arrive at its maximum accuracy before seeing any examples, resulting in a data efficiency of 0.

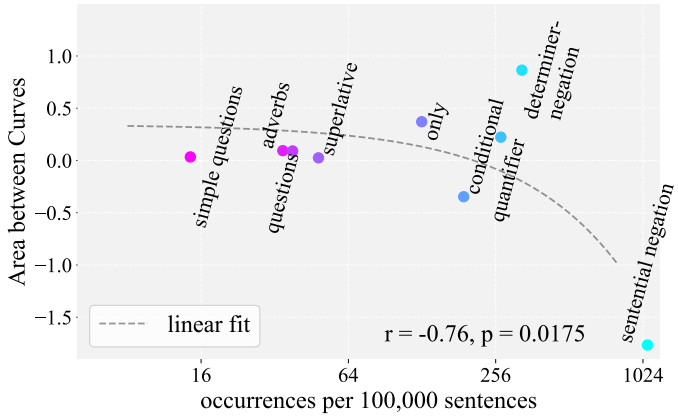


Figure 3.5: Normalised AbC for all licensing contexts until convergence of both contexts to 95% accuracy. AbC > 0 indicates a better performance of the all-context model and vice versa.

have reached 95% of their final accuracy. The larger this area is, the more impactful it is to remove all other NPI contexts and the more the model leveraged from these contexts. The learning curves of all contexts, along with an illustration of the AbC-measure, can be found in Figure 3.4.

As a first interesting observation, we see that for seven of the nine contexts, the all-contexts model learns faster and achieves higher final performance.[7] Both frequent and infrequent contexts thus benefit from information acquired by other licensing contexts in terms of both data efficiency and final accuracy.

This positive transfer can also be seen in Figure 3.5, where we plot the AbC for all licensing contexts against their frequency. This plot also confirms the relationship found in our previous experiments: the less frequent a context is, the more it benefits from other NPIs ($r = .76$, $p < .05$).

[7] A one-sided Welch's test confirms that the calculated AbCs are overall different from zero: $t = 2.61$, $p < .05$.

3.5 General discussion and conclusion

Summary In this first chapter of the main body, we present a framework to study learning dynamics in end-to-end, self-organising machine-learning language models. The framework lends ideas from the subfield of multi-task learning, which it connects to conceptual knowledge, in our case, linguistic theory. In our experiments, we show that neural language models can find and exploit similarities between the different language construction rules or, more precisely, how they learn the task cluster of NPI licensing. We find that LMs pick up on similarities we assume from linguistic theory and exploit them to learn similar language constructions with less data and higher accuracy. Especially less frequent tasks benefit from this effect. The observed transfer behaviour mirrors the positive transfers from traditionally constructed MTL settings. While our framework is mainly focused on creating a scaffolding for the further analysis of learning dynamics, it also has potential implications for other research areas. Here, we illustrate how our setup and results relate and contribute to MTL, linguistics and interpretability research.

Multi-task learning research Studying LMs as multi-task learners, we observe several phenomena known from traditional MTL: when trained in parallel, similar (sub)tasks are learned more efficiently (compare Collobert et al., 2011; Kaiser et al., 2017), and with higher accuracy (Collobert and Weston, 2008; Kaiser et al., 2017), and this effect is stronger for less frequent tasks (Benton et al., 2017; Kaiser et al., 2017).

Our study differs in one crucial aspect from previous research on MTL: it looks at learning dynamics *within* one, larger, natural task instead of between tasks defined by the modeller. Consequently, the learning process itself is not constrained through a priori decisions concerning task selection or how tasks should be optimised together. In our scenario, contrary to traditional MTL, we use tasks and their hypothesised similarity only to *analyse* the learning process of the language model, not to inform its training. As such, the natural setting of our framework allows us to study traditional MTL phenomena, such as data amplification, eavesdropping,

and attention focusing (see Section 3.2.1), independent of arbitrary decisions regarding task selection and optimisation. Insights from the natural setting can then be transferred to scenarios in which more control over the selection of tasks may be required.

Interpretability research Secondly, studying language models as multi-task learners can contribute to the field of interpretability. On the most basic level, our results confirm previous findings in interpretability that LMs can adequately model NPIs (Jumelet and Hupkes, 2018; Wilcox et al., 2019; Marvin and Linzen, 2018). We add to this literature by *explicitly* showing that LMs connect different types of contexts through a common concept (NPI licensing) through their learning behaviour. Contrary to previous work, we are tapping the learning process as a source of information to understand the inner workings of these models better.

Traditional concepts from MTL, such as the earlier mentioned explanations of Caruana (1993) and Ruder (2017) (Section 3.2.1) are valuable to better understanding what models are learning and how. For instance, when we observe that the solution of models improves when more varied NPI material is presented (our single- versus all-context experiment), MTL can aid in formulating concrete hypotheses about *why* this is the case. This, in turn, can help us improve our understanding of the solutions that the model learns. For instance, the single-context models usually level off on a lower accuracy level than the all-context model (see Figure 3.4). This is not merely explainable by the amount of data, as we continue to add training examples in either case. The difference between models instead appears to be due to the variety of the training data. The idea of *attention focusing* (Caruana, 1993, 1997; Ruder, 2017) helps us to understand what is going on: by being trained on more varied NPI material, the model can better sort out which features are relevant and which ones are instead idiosyncrasies correlated with specific contexts. Such hypotheses can then help inform further experiments that investigate – for example – which features specifically are better learned through attention focusing.

Linguistics research Finally, studying language models as multi-task learners can also contribute to linguistics. In our study, we show that LMs can find and exploit similarities between linguistically defined concepts. Turning things around, this generalisation behaviour of models can also be seen as a confirmation of the linguistic task hierarchy that we assumed from the start. The language modelling objective is unconstrained by linguistic theory and, therefore, does not necessarily have to find the exact solutions as linguistics. Similarity derived from the learning behaviour of language models might, therefore, be used as a tool to work on more disputed ideas in linguistics and to form new hypotheses in linguistic theory. While the linguistic insights that can be drawn from the current study are relatively limited, they do provide a proof of concept for future work: we show that domain knowledge and the learning behaviour of neural models can be connected.

Conclusion The work in this chapter is a principal step towards the primary goal of this dissertation in that it formulates a framework to understand the learning dynamics of language models in a holistic and unconstrained way. By understanding the model's conceptualisation as emerging subtasks it has to optimise, we can analyse the self-organisation ability of the language model as it would be expected from complex systems theory (Section 1.2.2): Throughout the learning process, the model uncovers regularities (or 'similarities') among data points – even though they may not be related in the surface structure of the sentence – and learns to pool these data points into the same concept. This generalised concept can be used to learn new data points more efficiently. Conceptualisation may change throughout the training as the model starts to understand the data distribution on a more abstract (or *compressed*) level or when the data distribution shifts as a sign of adaptability – a hallmark of complex systems.

While we concentrated on the conceptualisation and learning of linguistic knowledge, the framework can generally be utilised for any conceptual knowledge a machine learning model learns. We will continue to utilise and expand upon the framework in Chapter 4 by switching the methods

from the undersampling approach from this chapter to oversampling (i.e. fine-tuning) of specific phenomena. This will enable us to apply the approach to many phenomena from the BLiMP benchmark (Warstadt et al., 2020). We additionally connect the framework to the *internal* dynamics of the model by analysing its gradients (compare structural and behaviour interpretability in the general background 2.3).

3.6 Limitations

The limitations of this chapter lie mostly on the implementational level: Firstly, undersampling different linguistic phenomena is difficult, as they are often covert in the surface structure of a sentence. Consequently, it is not straightforward to filter them out of a corpus. Further, prevalent phenomena such as *subject verb agreement*, present in almost every sentence, are impossible to filter out of a corpus and maintain a usable training set. Secondly, the retraining of a language model for each investigated phenomenon is prohibitively expensive and does not scale to many phenomena or larger models. We will address both of these limitations in the following chapter.

Chapter 4

LINGUISTIC TASK-SPACES

In the previous chapter, we introduced a framework to connect the learning dynamics of language models with linguistic theory using ideas from MTL research. Different language phenomena (or 'tasks') share structure, and this shared structure can be leveraged by the model via generalisation. By looking at the generalisation behaviour, we can deduce which tasks share structure (or are 'similar'). If we apply this idea to many tasks, we can construct a 'linguistic similarity space', which is representative of an LM's conceptualisation of language. In this chapter, we will construct such similarity spaces and analyse their change throughout training.

4.1 Introduction

The language faculty of LMs improved greatly in the recent past and is nowadays close to indistinguishable from human abilities when it comes to generating linguistically acceptable language (Liang et al., 2022). With their impressive capabilities, LMs have become increasingly attractive as a subject of linguistic research (Baroni, 2022). One way of employing LMs is by using them as perfectly accessible 'lab animals' for linguistic theory testing (Scholte, 2017; Futrell et al., 2019b). However, the excellent performance of modern LMs comes with the caveat that they are inherently complicated to interpret (see Section 2.3). Language production

is complex, and similarly, a good model of language will be complex in itself. In this chapter, I introduce a method to interpret the language processing of LMs holistically and use it to analyse the change in their language conceptualisation throughout the training process. To do so, I will expand and improve the framework from Chapter 3. The framework uses the generalisation behaviour of LMs to get insights into how they share structure across linguistic phenomena. I will use this idea to show how this shared structure evolves throughout the learning process.

The approach of undersampling from the previous chapter had two principal implementational shortcomings:

1. It is computationally expensive to retrain a new model for each investigated phenomenon;

2. It is hard to filter certain phenomena (especially very common ones or very abstract ones)

This makes it challenging to apply to a more extensive range of linguistic phenomena and construct a meaningful 'linguistic task space'. Here, I will change the approach and rely on the oversampling of phenomena instead (i.e. selectively fine-tuning the phenomena in a pre-trained model). Oversampling is considerably easier to apply and cheaper, enabling us to scale to many phenomena. I will use the oversampling method to construct linguistic task spaces at different stages of pre-training and analyse how the language conceptualisation of LMs develops.

Outline I will first provide additional background information and summarise related work (4.2). Next, I will introduce the used linguistic data that will span our similarity space (4.3.1) and explain the methods I use to estimate similarity among them (4.3.2). In the experimental section 4.4, I pre-train three different language models, apply our method and then analyse the resulting linguistic spaces throughout the pre-training process. Finally, I discuss the results (4.5) and the methodological limitations (4.6).

4.2 Background and related work

4.2.1 Similarity spaces in MTL

The construction of similarity spaces has been a common goal in the multi-task and continual learning literature. Researchers in multi-task learning have been interested in constructing taxonomies of tasks, which relate different tasks in terms of 'similarity' for many years (Ben-David and Borbely, 2008). These similarity spaces can be used to determine the degree of transfer to expect when different tasks are trained jointly. One of the earliest examples of constructing task-similarity spaces can be found in Thrun and O'Sullivan (1996). More recently, Zamir et al. (2019a) constructed a task taxonomy for a range of computer-vision tasks based on how valuable the representations of one task are for solving another. Standley et al. (2020b) improve on Zamir et al.'s work by basing their taxonomy on the transferability of information across tasks. Similarly, Achille et al. (2019) create 'task-embeddings' for visual classification tasks by comparing their task structure through Fisher Information Matrices. More theoretically motivated, Lee et al. (2021) investigate task similarity in a highly controlled setting using synthetic tasks and find their similarity measure predictive of learning outcomes.

4.2.2 Linguistic spaces

In recent years, LMs have become potent tools for various applications. Their impressive language faculty makes them also interesting as theories of language (Baroni, 2022). However, just like humans (Watson, 1913; Titchener, 1912; Nisbett and Wilson, 1977), LMs cannot introspect and report their 'cognitive' processes. As a consequence, to understand *how* LMs are processing language, we have to find methods to interface their internal processes and new subfields such as 'synthetic linguistics' Chowdhury and Zamparelli (2019) have recently emerged. The method we introduce here to construct linguistic task spaces is a way to construct such 'synthetic theories'.

To our knowledge, no work constructs comprehensive task taxonomies for language models. However, there are notable works that utilise similar methodologies: Weber et al. (2021) show how language models are generalising across linguistically similar constructions and suggest that this hints towards an implicit task hierarchy in higher-level tasks like language modelling. Chowdhury and Zamparelli (2019) fine-tune language models on grammatical and ungrammatical constructions and observe that grammatical structures are more straightforward to integrate than their ungrammatical counterparts. Prasad et al. (2019) use a paradigm inspired by psychological priming experiments to determine the relationship between different linguistic phenomena and recover their hierarchical organization. Pérez-Mayos et al. (2021) fine-tune BERT on various downstream tasks and evaluate how fine-tuning interferes with BERT's syntactic understanding using *structural probes*.

Finally, the linguistic similarity spaces that we are suggesting are highly related to the idea of conceptual spaces, a style of knowledge representation between symbolic and distributed approaches known from the cognitive science literature (Gardenfors, 2004, 2014).

4.3 Methods

We combine the work of Zamir et al. (2019a); Standley et al. (2020b); Achille et al. (2019) and Weber et al. (2021) to create a similarity space of linguistic tasks via what we call 'similarity probing'. To do similarity probing, we use either the model's behavioural data ('transfer probing') or its internal dynamics ('gradient probing'). In *transfer probing*, we construct the similarity space by probing the measurable transfer learning across different linguistic tasks. In *gradient probing*, we construct similarity spaces based on the alignment of gradients when we fine-tune on different tasks. Gradient probing is inspired by the work of Yu et al. (2020), who relate properties of gradients in multi-task settings to the generalisation behaviour of MTL models. In the following subsections, we first introduce the data we use for probing and afterwards explain the respective probing

methods in detail.

4.3.1 Data

Our experiments contain two separate training steps: 1) the language model pretraining and 2) probing by fine-tuning linguistic tasks. We pre-train our models on the standard split of a common Wikipedia corpus (*wiki103*; Merity et al., 2017). For the linguistic probing, we use the BLiMP-corpus (Warstadt et al., 2020). BLiMP is a corpus of minimal pairs containing data for 13 higher-level linguistic *phenomena*, which can be subdivided into 67 types of realisations called *paradigms*. A minimal pair consists of two almost identical sentences, only distinguished by a minimal difference that renders one of them agrammatical. Each paradigm contains 1000 individual data points, sizing the whole corpus at 67_000 data points. To construct linguistic task spaces, we consider every paradigm as a separate task and investigate their similarities. We split the data for each paradigm with a ratio of 85% for probe training and 15% for evaluation.

4.3.2 Similarity Probing

Similarity probing estimates how much similarity a language model finds between implicit linguistic tasks A and B by examining different aspects of their learning dynamics. In alignment with previous literature in multi-task learning, we concentrate on two measures of similarity between A and B: their performance transfers (Zamir et al., 2019b; Standley et al., 2020b) and their gradient alignment (Yu et al., 2020). We obtain an estimate of transfers and gradient alignments between A and B by selectively fine-tuning the language model on linguistic task A and measuring its impact on B. Due to the interwoven nature of linguistic 'tasks' in natural language (as described by Weber et al., 2021), it is not as straightforward to fine-tune a language model on a specific linguistic phenomenon in isolation. We will provide details on this challenge and how we resolve it in the following paragraph. Subsequently, we formalise our exact methodology.

Isolating linguistic phenomena Certain phenomena are necessarily present in every sentence. This makes it difficult to fine-tune phenomenon A and estimate its impact on omnipresent phenomenon B. If we use natural language sentences, all training examples for A will automatically also include B. As a result, we cannot tune A in isolation. More concretely, if we choose `subject verb agreement` as B, structural information about the phenomenon of `subject verb agreement` will be contained in training data for any A we are choosing, independent of the relationship between phenomenon A and `subject verb agreement`. Besides the omnipresence of certain phenomena, there is another challenge: natural language data is multi-faceted, and we might find spurious correlations between the distributions of A and B that have nothing to do with the relatedness of the phenomena. For example, suppose A and B come from different paradigms but the same phenomenon. In that case, they may have larger vocabulary overlap compared to when they stem from different phenomena[1]. The similarity of the vocabulary distributions for different tasks therefore can potentially confound transfers as well. Our method has to isolate linguistic phenomena such their similarity measures are not influenced by their occurrence or spurious correlations between their distributions.

Our solution is to identify the relevant parameters for a phenomenon (isolated from potential confounds) and selectively update only those parameters during fine-tuning. To do so, we take advantage of the fact that BLiMP is a dataset of minimal pairs: Each positive sample of a linguistic task has a negative counterpart that – by definition – only differs in the correctness of the respective task. We contrast positive and negative examples to isolate the linguistic phenomenon: With Θ being our model parameters, at every update, we calculate not only the gradients $g_+(\Theta)$ for positive examples but also $g_-(\Theta)$ based on their corresponding negative counterparts. We then identify a parameter subspace S that is relevant to the linguistic task by calculating the difference $g_\delta(\Theta)$ between $g_+(\Theta)$

[1]An illustration of this can be found in Appendix A.2.1.1 showing the Wasserstein distances $\mathcal{W}$ as well as the absolute token overlap between the vocabulary distributions of all BLiMP paradigms.

and $g_{-}(\Theta)$ in which we only include model parameters that differ in $g_{\delta}(\Theta)$ with a margin of $\epsilon = 10^{-3}$ from 0:

$$S = \{\theta : |g_{+}(\theta) - g_{-}(\theta)| > \epsilon\}$$

In other words, we select those parameters where the gradients for positive and negative examples are sufficiently different. We then update only parameters contained in S by using $g_{\delta}(\Theta)$.

Transfer probing We determine the transfer between A and B by fine-tuning a language model on A and measuring the performance on B before and after the fine-tuning. Fine-tuning on paradigm A may have three potential influences on paradigm B. We interpret them as follows:

1. The performance of B increases: we suppose A and B to be related and have high similarity.

2. The performance of B decreases: we suppose A and B to be related and to have low similarity.

3. The performance of B is unchanged: we assume A and B to be unrelated.

To mitigate floor and ceiling effects in performance evaluation, we normalize all transfers. For negative transfers, we normalise by the maximally possible accuracy loss (which is the pre-fine-tuning accuracy), and for positive transfers, we normalise by the maximally possible accuracy gain (i.e., 1 - the pre-fine-tuning accuracy).

Gradient Probing Besides only relating tasks on the level of performance transfers (similar to Standley et al., 2020b; Zamir et al., 2019b; Weber et al., 2021), we can also directly relate tasks in the parameter space by comparing the overlap of their respective subspaces S and the alignment of gradients $g_{\delta}(\Theta)$ in those subspaces. Taking inspiration from the work of Yu et al. (2020) on gradient alignment of different tasks in multi-task learning, we assume that:

1. If A and B have greater subspace overlap and g_δ are aligned, the tasks will benefit each other, and we consider them similar.

2. If A and B have a greater subspace overlap, but g_δ does not align, the tasks will interfere, and we consider them dissimilar.

3. If A and B have a small or no subspace overlap, the tasks will not interact, and we consider them unrelated.

We determine the **overlap** between subspaces of tasks A and B by calculating their Jaccard-similarity:

$$Jaccard(S_A, S_B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|}$$

The alignment of task-relevant subspaces can be determined by calculating the **cosine similarity** of their respective g_δ.

4.4 Experiments

We here employ the similarity probing methods described in the previous section on three different generative language models throughout their training process.

4.4.1 Experimental details

Experiments are separated into pre-training and subsequent probe-training. This subsection provides details on training details in either stage.

Models and pre-training For our experiments, we employ decoder-based generative transformer models based on code from the fairseq library (Ott et al., 2019). The principal difference between the three models is their amount of trainable parameters. The smallest model TLM-27M contains ∼27M trainable parameters[2]. The two other models, TLM-70M

[2]layers = 3, hidden- and embedding-size = 256, attention-heads = 4, ffn-size = 1024

and TLM-203M, double the respective hyperparameters, which results in models with ∼70M and ∼203M trainable parameters. We keep all other hyperparameter settings related to training constant[3]. We use the Adam optimiser during the pretraining phase to achieve higher convergence speed. We pre-train the models for up to 20 epochs[4] to final perplexities of 65.21, 38.32 and 27.61 on the validation set of wiki103.

Probe tuning During the probing phase, we adapt some hyperparameters to avoid potential confounds: We switch to plain stochastic gradient descent (SGD) to preempt interference of Adam's momentum terms (Kingma and Ba, 2015) with the probing process. Additionally, we change the batch size to 850, such that every batch contains all possible data points of the train split of a probed paradigm. The rationale for using a large batch size is to average out as many idiosyncracies of the individual data points in the learning signal as possible. We fine-tune on the data of single paradigms until the model's performance on the same paradigm converges. Our stopping criterion is defined as performance being lower or equal to the average of the last five steps.

4.4.2 Experimental results

We will first consider the general properties of the probing process and the resulting similarity spaces and afterwards demonstrate how they develop throughout the training process.

Probe tuning During fine-tuning, we observe only small increases in perplexity on the *wiki103* validation set for all models. At the same time, performance on the fine-tuned paradigms improves in all paradigms, indicating that our fine-tuning method is indeed selectively updating a specific linguistic task (for fine-tuning details, see Appendix A.2.2.2).

[3]Hyperparameters: batch size = 16, dropout = 0.1, learning rate = 0.0001

[4]Throughout the pre-training, we save model checkpoints for later analysis at 0, 1, 2, 3, 4, 5, 10, 15 and 20 training epochs, respectively.

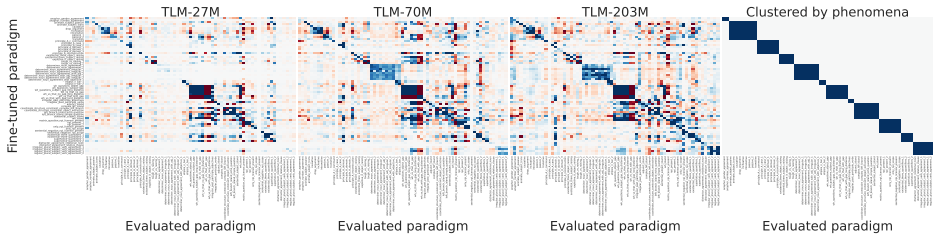


Figure 4.1: Transfer spaces for our three models after two epochs of pre-training. To the right, we can see how paradigms cluster into phenomena according to BLiMP.

Transfer spaces Figure 4.1 shows the resulting transfer matrices from probing the different models after two epochs of pre-training. Each heatmap cell represents the normalised transfer from a fine-tuned to an evaluated paradigm. Next to the transfer spaces, we show how paradigms cluster based on the membership to the higher-level phenomena. The transfer matrices are ordered by the size of their respective models. We can see how, with increasing model size, the transfer pattern across paradigms becomes increasingly similar to the clustering of the phenomena (i.e. models increasingly generalise within linguistic phenomena). We expect the models to generalise according to the clustering by phenomena. However, this is not the only 'hypothesis space' with which we can compare the transfer spaces and hypothesis spaces like this are a straightforward way to test ideas about generalisation patterns. Alternative hypothesis spaces could follow other clusterings (e.g. 'linguistic tasks that require quantifiers').

If a model generalises across the different paradigms that make up a phenomenon, we can say that it has a grasp of the overarching concept of the phenomenon. We quantify this 'grasp' by calculating the average transfer between paradigms from the same phenomenon (see Figure 4.2). A high value indicates that the model strongly generalises the phenomenon. From the table, we see that the larger the model, the more it has an overarching understanding of the different phenomena. Interestingly, many phenomena have low transfer values but very high standard deviations. This means that the paradigms within them form subclusters that are highly interfering with each other (see especially `filler-gap dependencies`

[**FG-DEP**]). The model discovered the paradigms' relatedness but cannot reconcile them. Different `irregular forms` [**IRR-F**] strongly interfere with each other.

We can conclude that language models do generalise within the same phenomena; however, that generalisation is stronger for some phenomena than for others. Larger models are better at exploiting within-phenomena similarities. Within-phenomena subclusters are highly interfering with each other. Other paradigms are just treated idiosyncratically by the model without any interactions with other realisations of the same phenomenon (see, e.g. `binding` [**BIND**]). In the next paragraph, we will relate the transfer spaces with gradient spaces.

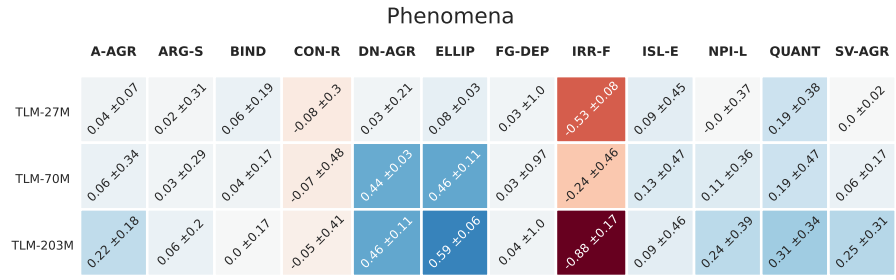


Figure 4.2: The degree of within-phenomena transfer for different models pre-trained for two epochs.

Gradient spaces We will now look at the properties of the subspaces of different paradigms, how they overlap and how aligned they are. We first average the sizes of all subspaces and represent them as a portion of all model parameters to get an idea of how distributed different paradigms are processed. The average subspace size is $|S_{27M}| = 0.57\%(\pm 0.5)$, $|S_{70M}| = 1.39\%(\pm 1.4)$, $|S_{203M}| = 1.45\%(\pm 1.44)$ of all model parameters[5].

Is the overlap of the subspaces of different paradigms (i.e. their $Jaccard$ similarity) predictive of the transfer between them? If we con-

[5]It is essential to remember that this value is influenced by the threshold parameter ϵ

struct a similarity space solely based on $Jaccard$ similarity, correlations with our transfer spaces are very low ($r_{27M} = .09$, $r_{70M} = .15$ and $r_{203M} = .14$, respectively). That means that just because paradigms are processed through the same parameters, it does not imply that they share structure. Additionally to being overlapping, the subspaces also have to be aligned. If we multiply the $Jaccard$ similarity with the cosine similarity of the subspace gradients, the correlations of the gradient space with the transfer space jump to $r_{27M} = .68$, $r_{70M} = .63$ and $r_{203M} = .64$. The alignment of relevant subspaces, hence, predicts the transfer learning between different paradigms.

Testing similarity spaces We can now use our similarity spaces to test them against different hypothesis spaces (like the hypothesis space on the right in Figure 4.1). We correlated the gradient spaces with 'clustered by phenomena' space and different controls and find that phenomena are much more predictive of generalisation than our vocabulary controls (Wasserstein and token overlap; see Table 4.1). This confirms that the models generalise within phenomena across something beyond their shared vocabulary.

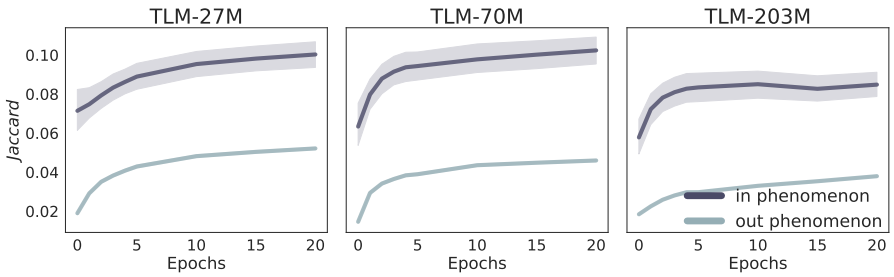
Hypothesis space	TLM-27M	TLM-70M	TLM-203M
Token Overlap	.9	.18	.16
Wasserstein distance	-.21	-.26	-.27
Clustered by phenomena	.39	.42	.44

Table 4.1: Correlations of gradient spaces with different hypothesis spaces.

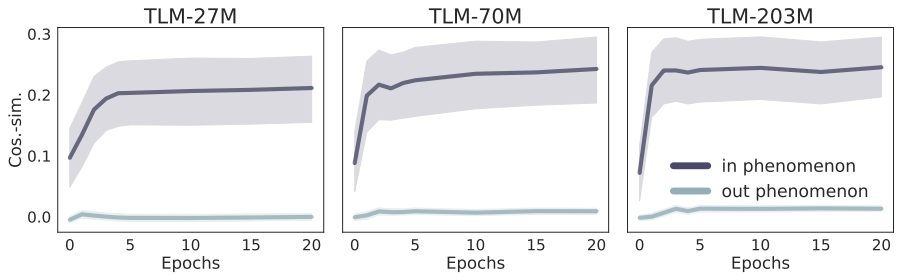
We have seen that we can use our method to construct spaces of an LMs language conceptualisation. How does this conceptualisation change throughout training?

Linguistic spaces throughout training

We will now use our linguistic spaces as an interpretability tool to analyse how LMs conceptualise language throughout the training process. We will first look at how well the LMs learn the different BLiMP paradigms and then investigate the development of their similarity spaces throughout training.



(a)



(b)

Figure 4.3: (a) The average $Jaccard$ similarity of task-subspaces either within the same phenomenon or outside the phenomenon; (b) the average inner product of g_δ of the overlapping subspaces.

BLiMP learning We evaluate all model checkpoints on all paradigms *without fine-tuning*. During pre-training, the performance on the BLiMP benchmark increases steeply in early training and then levels off. The

increase is especially abrupt for the larger models (the learning curves can be found in Appendix A.2.2.1). Interestingly, during the *probing phase*, our contrastive fine-tuning of linguistic tasks improves much quicker and to higher final performance in models that are pre-trained for more epochs (for a visualisation see Appendix A.2.2.2), indicating that our selective updating works better if it can latch on to previous knowledge already contained in the model parameters and subspace selection is, therefore, more meaningful. So, how does the processing of BLiMP paradigms change throughout the pre-training?

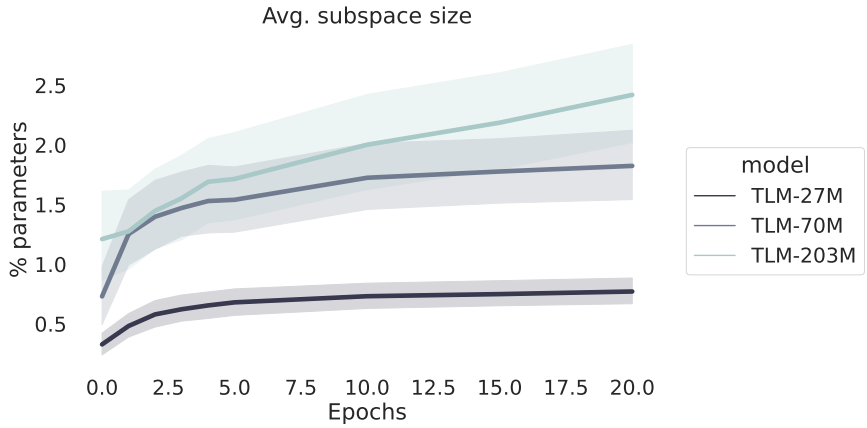


Figure 4.4: The development of subspace size throughout language model.

Development of similarity space When we consider the average size of subspaces $|S|$, we find that the subspaces continuously become larger during pre-training (see Figure 4.4). The LMs appear to process linguistic tasks initially more localised and continuously increase the degree of distributedness of the processing. But does this also increase the *interconnectedness* of different paradigms with more training? We observe that, indeed, the $Jaccard$ similarity within-phenomenon increases with training, meaning that paradigms from the same phenomenon share more

parameters in later training (see Figure 4.3a). At the same time, the overlap between paradigms from different phenomena only increases marginally. Also, the alignments of gradients increase selectively for paradigms from the same phenomenon (see Figure 4.3b). Larger models, again, align related paradigms much faster and to a higher degree than smaller models. This shows how the processing of linguistic phenomena starts idiosyncratic (separated parameters and not aligned), and with training, the sharing of structure increases (shared parameters, where appropriate and aligned).

Similarity space stability Overall, we find that similarity spaces are remarkably stable: similarity patterns are present from very early in training (within the first epochs), and any change that happens later is instead a reinforcement of that pattern rather than a substantial change (see Figure 4.5). This is somewhat surprising if we compare it with patterns of human learning, which are much more marked by stages (Piaget et al., 1952; Gopnik et al., 1999, 2004). As we deepen our knowledge, new patterns emerge – in comparison, language learning in LMs appears to be continuous rather than marked by such incisive shifts.

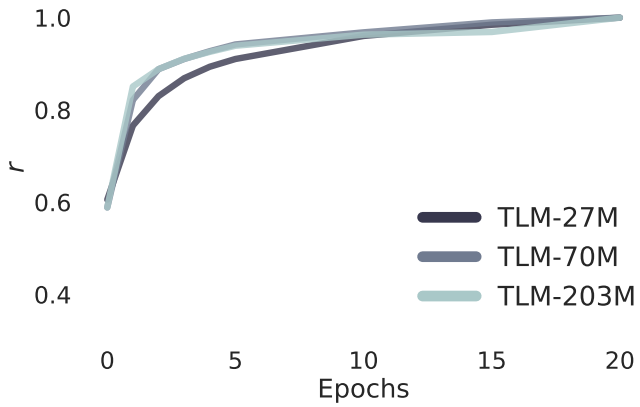


Figure 4.5: Correlation of gradient similarity spaces with the trained-out gradient space. Correlation is very high after only a few epochs, indicating that the overall pattern of gradient similarities only changes minimally.

4.5 General discussion and conclusion

Summary In this chapter, we introduced a method to construct linguistic task spaces, a conceptual space of a language model's language understanding (Gardenfors, 2004, 2014). To do so, we introduce a technique to isolate linguistic phenomena from their entanglement with other phenomena and enable us to fine-tune them selectively despite their latent nature. Thereby, we resolve major issues of the undersampling approach from the previous Chapter (3). We follow the established methods from MTL and obtain *behavioural* similarity estimates by selectively fine-tuning specific linguistic tasks and evaluating their effect on other tasks. We also introduce a new method and produce *structural* similarity estimates by analysing their shared parameters and alignment. The gradient alignment between linguistic tasks predicts their transfer learning on the BLiMP benchmark. Structural methods are usually more expressive, maintaining a larger amount of information (compare 2.3).

Use cases The resulting similarity spaces can be used in different ways. From a linguist's perspective, the space can be used as an explicit hypothesis about the relatedness of different linguistic structures (in the sense of constructing a '*synthetic linguistic theory*'; Chowdhury and Zamparelli, 2019). By constructing a hypothesis space and testing it against the similarity space, a linguist can do simple and quick hypothesis testing. From the machine learning perspective, the similarity space can be used as a tool to interpret the extent of a language model's grasp on different language phenomena and its generalisation behaviour. In our analysis, we concentrated on the machine learning perspective by analysing the learning dynamics of the language model.

Insights into the learning process We observe that larger models are *quicker* to latch on to linguistically meaningful similarities between tasks and, overall, are *better* at exploiting their similarities, and they identify *more* related linguistic tasks. When looking at the changes in relevant subspaces for the different phenomena, we observe that they become larger with training. The overlap primarily increases across related linguistic tasks (within phenomena) and not so much for unrelated linguistic tasks. Similarly, the gradient alignment of the related subspaces selectively increases. Hence, in early training, LMs learn linguistic tasks more idiosyncratically and in isolation but later start distributing and connecting them broadly, sharing structure and using more parameters to encode any linguistic task. This runs against an intuition from learning theory that assumes that learning is finding efficient compression rules (see Section 2.2.2): with a better understanding of a task, we find better rules to compress the input data to a lower *intrinsic dimensionality* (Cheng et al., 2023). The intrinsic dimension describes the number of dimensions required to represent data. How does this reconcile with the increase of *extrinsic* dimensionality we observe here? A more distributed processing of concepts allows for more overarching structure sharing and generalisation across different subconcepts, which might be necessary to achieve a lower intrinsic dimensionality. Intrinsic and extrinsic dimensions might be inversely related in language models. Ultimately, when analysing the

similarity spaces throughout the training process, we find that similarity patterns are surprisingly stable, and learning appears rather to reinforce the existing patterns than let new patterns emerge – a learning behaviour we would expect from more human-like learning.

Future reseach From here, many future routes can be taken for further research. As language models become more apt in many domains, they can become interesting models of cognition beyond language. We have presented an efficient method of extracting conceptual spaces from their generalisation dynamics. Further, linguistically inclined researchers can construct hypothesis spaces based on controversial concepts in the field and test them against linguistic spaces extracted from language models. To improve upon our method for task-space construction, we see multiple viable routes to follow: future research may utilise more potent state-of-the-art LLMs to construct similarity spaces. More advanced models may find more subtle structural similarities that are more informative to linguists. Further, we imagine swapping the 'anchors' that span the space (i.e. in our case, the BLiMP 'paradigms') with anchors derived from the learning dynamics themselves might yield more expressive task spaces. This way, we become less constrained by any assumptions about the language structure made by linguistic theory.

4.6 Limitations

To put the presented method into perspective, I will briefly elaborate on the most important limitations.

Firstly, as discussed in the previous section (4.5), a major weakness of our probing approach lies in the necessary top-down definition of 'anchors' that we use to span the space. We utilise human-defined phenomena and relate them to each other. However, a more accurate linguistic space can probably be described by 'anchors' that are defined through the model itself and span the conceptual space with maximal expressivity.

Secondly, while our approach applies to all types of knowledge do-

mains, it requires *minimal pairs* of phenomena within that domain to fine-tune them selectively. Minimal pairs are primarily used in linguistics and are uncommon in other knowledge domains.

Thirdly, our fine-tuning and evaluation data are i.i.d. and come from a very narrow distribution: the data are not natural but synthetic, and all data are generated using the same templates. We use this very narrow i.i.d. data to asses the fine-tuning success during probing. However, we cannot be entirely sure whether we succeeded in fine-tuning a specific linguistic phenomenon rather than some idiosyncracies of the narrow data distribution. While our contrastive fine-tuning approach might elevate this issue slightly, it does not dispel our doubts completely. The optimal way to guarantee our results would be the evaluation on a set from a separate distribution.

Chapter 5

AUTOMATED CURRICULUM LEARNING FOR INTERPRETABILITY

We have seen in the previous chapter how the learning process of LMs exhibits surprisingly little change in structure (i.e. shifts in generalisation patterns). Here, we will employ another method to analyse the learning process of LMs: automated curriculum learning algorithms. Automated curriculum learning (CL) algorithms select the data points for a target model to learn optimally at any point in training. In this chapter, we will analyse the policy of an automated CL algorithm optimised for language data.

5.1 Introduction

In the first two chapters of this dissertation, we investigated the different linguistic concepts that a language model finds and utilises to improve its language abilities. In this chapter, we are now interested in whether providing certain features in the data with different prioritisation can influence the learning speed and the quality of the learning outcomes. To that end, we use an automated curriculum learning method as an interpretability

tool.

Automated curriculum learning usually consists of small meta-learning algorithms that learn to optimise the distribution and order of the training data of a target model (a detailed explanation of curriculum learning will be given in the background section of the current chapter). By training such an algorithm for language models and analysing its policy, we can infer what is learned at the different stages during the training process. Out of all NLP tasks, language modelling requires by a large margin the most computation and data during training. It is surprising that — to the best of our knowledge — there is no established curriculum learning method for LM training. We, therefore, resort to an automated curriculum learning technique from computer vision called 'commentaries' (Raghu et al., 2021) and apply it to our learning problem. Raghu et al. (2021)'s approach learns a 'teacher' model that takes a data point and a learning state indicator of the target model and predicts a respective weight. This weight is then applied to the target model's loss of that data point. An extensive explanation of the mechanism of commentaries will be given in Section 5.3. Since it is computationally expensive to train teacher models in the commentaries framework, we start our analysis with computationally cheap context-free grammars (CFGs) (Section 5.3.1) and only afterwards transition to full-scale models (Section 5.3.2). While doing so, we will first see how our teacher models create apparently sound curricula which match well with what we would expect from the literature. Under closer inspection, however, we will uncover how they are very brittle and inconsistent. In the second part of this chapter (from Section 5.3 onwards), we dive deeper into the reasons for this brittleness and find that Raghu et al. (2021)'s framework — rather than providing a sound data-based curriculum strategy — is fully data-agnostic and that learning advantages stem from interactions of the curriculum shape with the Adam optimiser (Kingma and Ba, 2015). As a result of the interaction, the parameter updates of the model are scaled in size, similar to a change in learning rate; the curriculum yields no benefit beyond that. Ultimately, we show how the *curriculum-Adam*-interaction is not limited to the commentaries framework but can also explain results in other curriculum learning approaches when they are combined with

optimisation through Adam. Importantly, we show that plain Adam with properly tuned hyperparameters outperforms curricula in all of our tested settings.

Outline We will start by providing additional background information on curriculum learning (Section 5.2) and on the commentaries framework in particular (Section 5.3). We will then dive into our pilot studies using CFGs (Section 5.3.1) and continue on encoder-based language models (Section 5.3.2). In Section 5.4, we will dissect the brittleness of the curricula and show the generality of our uncovered interaction. We will conclude in Section 5.5.

5.2 Background

Inspired by human learning, curriculum learning (CL) exposes machine-learning models to a limited, 'simple' portion of the data distribution at first and only gradually introduces 'complex' examples into the training process until the whole training data is used (Elman, 1993; Rohde and Plaut, 1999; Krueger and Dayan, 2009; Bengio et al., 2009). To this end, every CL approach has to formalise which training examples are 'simple' and which are 'complex' (i.e. determine a *difficulty measure*) and decide on the rate at which to add 'more complex' examples into training (i.e. define a *scheduling function*). Difficulty measures and schedule functions can be determined in different ways. We here shortly summarise a broad grouping of approaches: **hand-crafted curricula** and **automated curricula**.

5.2.1 Hand-crafted curricula

The simplest type of curriculum fixes the difficulty measure and schedule function prior to training without adapting them dynamically according to the learner state. The choice of the difficulty measure is usually based on the practitioner's intuitions and experiences. Common *difficulty measures* in NLP include the sequence lengths of an input (or the closely related

depth of the parse tree) (Tay et al., 2019; Martínez Alonso et al., 2017; Platanios et al., 2019), the number of coordinating conjunctions (Kocmi and Bojar, 2017) or the diversity of the used vocabulary (Platanios et al., 2019). *Schedule functions* typically expand the data distribution towards more difficult examples monotonically, either as a step-function (Bengio et al., 2009; Spitkovsky et al., 2010a; Kocmi and Bojar, 2017) or continuously (Hacohen and Weinshall, 2019; Platanios et al., 2019; Penha and Hauff, 2020; Liu et al., 2018). Examples of step functions can be seen in Figure 5.9a. Hand-crafted curricula have the advantage of being cheap and easy to implement. On the other hand, the choice of the correct setup requires experience or expert domain knowledge, idiosyncracies of data and tasks make them potentially difficult to generalise, and the method is 'coarse', such that it is limited to the predefined structure and cannot dynamically adapt to the current state of the learner.

5.2.2 Automated curricula

There are different approaches to addressing the shortcomings of hand-crafted curricula. We coarsely bin them into (1) non-parametric and (2) parametric solutions. The (1) non-parametric curricula can dynamically adapt the schedule function and/or difficulty measures to the current state of the learner without learning any additional parameters. The most common approach to non-parametric curriculum learning is self-paced learning (SPL; Kumar et al., 2010). In SPL, data points are only included in training when they produce losses that fall under a dynamic threshold. On the other hand, (2) parametric approaches utilise meta-learning to learn additional parameters (often times referred to as 'teacher'-models) that predict a data point's utility towards a target (or 'student')-model's learning objective (for examples, see MentorNet by Jiang et al. 2018, ScreenerNet by Kim and Choi 2018, and learning-to-teach by Fan et al. 2018). The predicted utility is then used to optimise the learning process. As they require no manual work, end-to-end approaches are convenient. However, they come oftentimes with the high computational cost of optimising 'teacher' models, making them too expensive to optimise with large target

models.

5.2.3 Theoretical underpinnings

Theoretical explanations of the efficiency of curriculum learning remain relatively sparse. The two most referred-to explanations can be found in Bengio et al. (2009), which state that CL helps 1) with denoising the dataset and 2) by smoothening of the non-convex optimisation landscape (as a form of continuation method; compare Allgower and Georg, 1980).

Despite all of their different forms and technical implementations, all curriculum learning approaches have in common that they cause a systematic shift in the learning signal the model is receiving. We refer to this universal shift of curricula as the *curriculum structure* throughout this paper. The curriculum structure is central to generalising our findings in later sections.

5.3 Automated CL with Commentaries

We here conduct a case study on *commentaries*, an existing parametric approach to curriculum learning. We start by summarising how the commentaries curriculum (Raghu et al., 2021) is learned and applied.

Mechanism To learn a curriculum, commentaries are formalised as a teacher model $T(x_i, i; \phi) \rightarrow w_i$ with parameters ϕ that takes a batch of data x_i and an indicator of the target model's current learning state i to produce a weight $w_i \in$ [0,1] for every data point in the batch. The indicator i is set to be the number of previous iterations for which the target model has been trained, and we denote I to be the total amount of updates for which we will train a model. Further, we denote the target model as S and its parameters as θ. At every iteration i, the weight-vector w_i is applied to the target model's loss $\mathcal{L}_{\text{train}}$.

The commentaries pipeline is divided into two phases: a teacher-pretraining phase and an evaluation phase. We depict both phases in

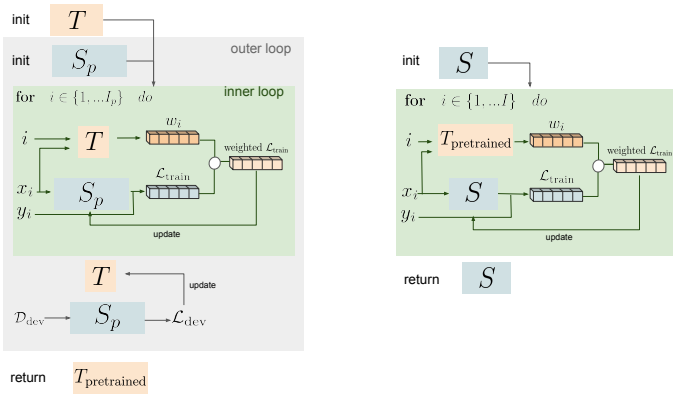


Figure 5.1: A visualisation of the commentaries-framework. The left side illustrates the teacher optimisation: The teacher model (T) is trained in the outer loop to optimise the learning process of the practice target models (S_{p}) in the inner loop. The number of iterations in the inner loop is limited by the amount of memory available. The right side shows how the pre-trained teacher is used to optimise a new student model to convergence.

Figure 5.1. During teacher-pretraining, the teacher is explicitly trained to minimise the loss of S on some held-out data $\mathcal{D}_{\text{dev}}$ by reweighing the training loss of S. To do so, several 'practice' target models S_{p} are trained on $\mathcal{D}_{\text{train}}$ for a limited amount of steps I_{p} while their loss $\mathcal{L}_{\text{train}}$ is weighted by the teacher-predicted w. For all training steps, the computational graph of S_{p} is maintained. Subsequently, S_{p} is evaluated on the held-out set $\mathcal{D}_{\text{dev}}$. Clearly, the resulting loss $\mathcal{L}_{\text{dev}}$ depends S_{p}'s optimised parameters $\hat{\theta}$. At the same time, $\hat{\theta}$ depend on the teacher parameters ϕ through the reweighing of $\mathcal{L}_{\text{train}}$ during training, such that:

$$\frac{\partial \mathcal{L}_{\text{dev}}}{\partial \phi} = \frac{\partial \mathcal{L}_{\text{dev}}}{\partial \hat{\theta}} \times \frac{\partial \hat{\theta}}{\partial \phi} \tag{5.1}$$

This makes it possible to backpropagate $\mathcal{L}_{\text{dev}}$ 'through training' to update the teacher parameters ϕ. The number of S_{p}'s optimisation steps I_{p} in the teacher pretraining phase is limited by the amount of memory that

can be allocated to store the computational graph. [1].

In the evaluation phase – after the teacher parameters ϕ have been pre-trained – a new target model S is trained to evaluate the teacher policy. Since there is no need to save the computational graph of the training at this stage, there is also no limit to the number of training steps I, such that we can now train S to convergence. For additional details, we refer to Raghu et al. (2021).

5.3.1 Pilot study on context-free grammars (CFG)

To have highly controllable experiments with low computational cost, we start by learning curricula for two tasks based on CFGs.

Dataset	Description	Example	Configuration
$a^n b^n$	A grammaticality judgement task in which the student is trained to predict whether a sequence is producible through $a^n b^n$ or not.	aadddd $\rightarrow$ 0 cccbbb $\rightarrow$ 1	$n \leq 6$ $\lvert V \rvert = 41$ $\lvert D \rvert = 15\_600$
$DyckLM$	A next-token prediction (i.e. language modelling) task in which the student is trained to generate different opening or closing parentheses. While it is possible to open a new parenthesis at any point, only the most recently opened parenthesis can be closed at any time.	({[]<> $\rightarrow$ } ({[]<>} $\rightarrow$)	depth ≤ 8 length ≤ 20 $\lvert V \rvert = 18$ $\lvert D \rvert = 5\_000$

Table 5.1: Overview of the CFG datasets

[1]To elevate this limitation, Raghu et al. (2021) suggest an approximation for the gradients of the teacher model (right-hand term in Equation 5.1) through truncated Neumann series and implicit vector-Jacobian products (for details, see original paper). However, Raghu et al. (2021) do not apply this approximation to their CL approach.

Experimental setup - Data

In generative linguistics, a CFG is a set of recursive rewriting rules (or 'productions') used to generate patterns of strings. We look into two types of CFGs: one with a sequence classification objective ($a^n b^n$; described in detail in Table 5.1) and the other with a sequence generation objective, mimicking autoregressive language modelling ($DyckLM$; see Table 5.1). While being a significant simplification of natural language, we can use the experiments on CFGs to get a better understanding of general patterns in the teachers' policy. We generate *training-* / *target-* / *validation-* / *test*-sets by splitting the generated data into portions of 50% / 20% / 10% / 20%, respectively.

Experimental setup - Models

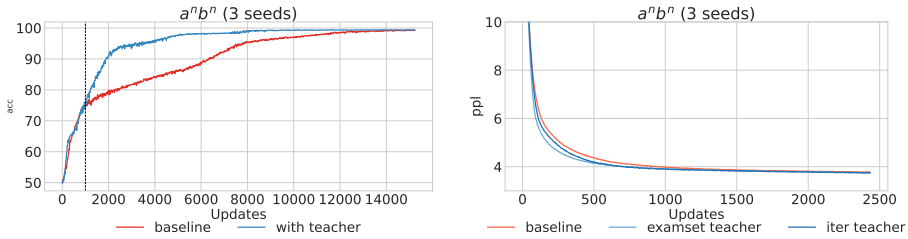
We implement our models using the fairseq-library (Ott et al., 2019). The details on the model architectures can be found in Table 5.2. The teacher architectures mirror the student used during teacher optimisation, replacing the output head with a sigmoid function. Accordingly, the teacher predicts a single weight for the loss of every input *sequence* in the classification tasks and a weight for every *token* in the sequence generation task. We optimise a teacher with practice students learning $a^n b^n$ or $DyckLM$ for I_p = 1000 steps, following the protocol described in Section 5.3. After the teacher optimisation, we use the teacher to train a student with the same architecture to its convergence.

Dataset	Model type	n_{layers}	n_{heads}	dim_{emb}	dim_{ffn}	n_{params}
$a^n b^n$	Transformer-Encoder	2	8	48	32	∼34k
$DyckLM$	Transformer-Decoder	2	8	8	32	∼4k

Table 5.2: Model details of the CFG-pilot experiment

Experimental results

Here, we will first evaluate the performance gains that the commentaries curricula are able to achieve. This is necessary in order to guarantee that the analysis of the curricula is meaningful. Afterwards, we will provide an analysis of the teachers' curriculum policy.



(a) (b)

Figure 5.2: Test-set performance of the student on (a) $a^n b^n$ and (b) $DyckLM$ respectively; the dashed line in both plots indicates the number of steps I_p were taken during the teacher optimization.

Performance Figure 5.2 (a) shows how the teachers' reweighting is able to substantially improve the students' convergence speed during student training for $a^n b^n$ and similarly for $DyckLM$ in Figure 5.2 (b). The dashed line indicates the respective I_p the teacher was trained for. Interestingly, the student in Figure 5.2 (a) only shows improvements over the baseline after it passed the dashed line. The reasons for this peculiar behaviour will become clear in Section 5.4. Our results confirm that the framework works in a comparable way on sequence processing tasks as for computer vision tasks in the original paper.

Analysis of the teacher policy First, we are interested in the *schedule function* that the teacher employs. We illustrate in Figure 5.3 how the average weight in every batch $\overline{w_i}$ rises, while the (normalised) standard

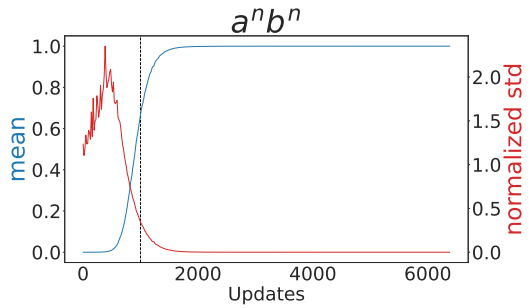


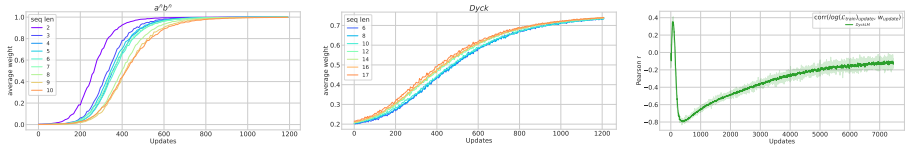
Figure 5.3: Distribution of weighting during the training of a student. The dashed line indicates up to which point the teacher has been trained.

deviation of w_i declines[2]. This means that the teacher model learns a high-variance (i.e. selective) weighting in early training, which includes more and more data points as the training of S progresses. This is in accordance with the principle idea of curriculum learning (Bengio et al., 2009). We observe the same pattern across all datasets and models in our experiments.

Next, we will have a look at the potential *difficulty measures* that the teacher employs. The most popular difficulty measures for CL in NLP are centred around grammatical complexity, measured through, e.g. the maximum depth of the parse tree (e.g. Tay et al. 2019; Martínez Alonso et al. 2017; Platanios et al. 2019). In $a^n b^n$, parse tree depth is directly reflected in sequence lengths, while for $DyckLM$, the parse tree depth is equivalent to the current depths of the stack of opened parentheses for each token. To see whether parse-tree depth is a crucial feature during the learning process, we relate the parse-tree depth of data points to their average received weight throughout training (Figure 5.4a and 5.4b). We can see a very clear pattern in which the teacher prefers shorter sequences early in training and continuously adds longer ones for the sequence classification in $a^n b^n$, while the pattern for $DyckLM$ is present but less

[2]Importantly, small weights do not lead to small updates, as Adam normalises the size of the gradient.

clear. Additionally, we correlate the teacher weights with the students' loss (compare self-paced learning; Kumar et al., 2010) on all samples in a batch and find that the teacher favours low-loss examples at the beginning of training (Figure 5.4c).



(a) (b) (c)

Figure 5.4: (a) weighting separated by sequence lengths for $a^n b^n$; (b) weighting separated by parse-tree depth for $DyckLM$ (c) Correlation student log loss and teacher weighting for $DyckLM$

Conclusion CFG experiments

From our pilot study, we infer that commentaries can be applied to different NLP problems and yield results similar to computer vision. We are further able to extract meaningful features from the teacher policies. At the same time, we observed that the results are brittle and dependent on hyperparameter settings in an unintuitive way (for an illustration with batch sizes, see Appendix A.3.1). We go on to test the approach on full-scale models.

5.3.2 Studies on full-scale models

We go over to experiment with large-scale models using natural data. We replicate Raghu et al.'s results and, in parallel, also transfer the approach to the fine-tuning of LMs. In the following subsection, we list the experimental setups, separated by modality and then go into a joint analysis of the results.

Experimental setup

We first replicate Raghu et al. (2021)'s results on **vision data** by using their original code[3]. We train CNN-based teachers with 2-layer CNN-based S_{p} on the CIFAR10 and CIFAR100 datasets, respectively, following the previously described procedure while sticking to the reported hyperparameter settings. After teacher training, we evaluate the teacher on different target models (2-layer CNN, ResNet18, ResNet34; He et al., 2016).

In parallel, we transfer the commentaries framework to **natural language data**, specifically to the popular NLU tasks from the GLUE benchmark (Wang et al., 2019c). Just like for the CFG datasets, we replace the CNN-based teacher and target models with small transformer encoder models from the fairseq library (Vaswani et al., 2017; Ott et al., 2019). To address the computational limitations of the teacher pretraining phase (mentioned in the previous *mechanism*-paragraph), we use frozen RoBERTa$_{\text{BASE}}$-embeddings (Liu et al., 2019c) instead of high-dimensional mappings from the vocabulary as the input to our teacher and target models. To further reduce the memory requirement of our setup, we average-pool the embeddings with kernel size and stride of 3. We then optimise teachers with this setup on the GLUE tasks. We evaluate the teacher by finetuning the full RoBERTa$_{\text{BASE}}$-model on the different GLUE tasks with their respective teacher.

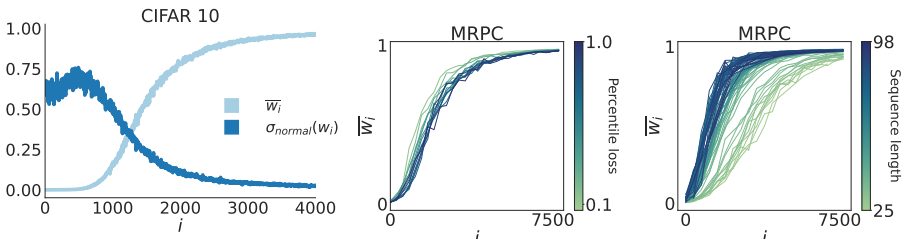
Experimental results

We first analyse the policy of the teacher models and then continue to evaluate their performance.

Commentaries learn reasonable curricula For both, CIFAR and GLUE, we find similar scheduling policies to the CFG experiments, with high variance at the beginning of training and uniform weights later. Considering difficulty measures, we find that the teacher's policy is making

[3] `https://github.com/googleinterns/commentaries`

[4] We choose MRPC as we consider it representative of most GLUE tasks. We find equivalent results for other GLUE-tasks.



(a) (b) (c)

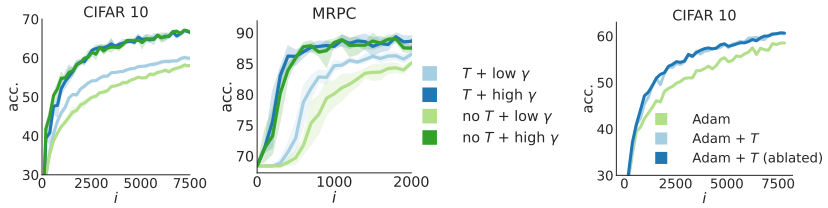
Figure 5.5: All of our pre-trained commentary teachers (vision and language) show the same pattern when predicting weights (a): predicting small value, high-variance weights early in student training to then predict higher and more uniform weights as student training progresses. When trained on an NLU task like MRPC[4](Dolan and Brockett, 2005), the teacher shows a slight preference for training examples with lower loss by assigning higher weights to these examples earlier in training (b). The preference is even clearer for its weighting policy in regard to sequence lengths (c): longer sequences are weighted up the beginning of training, and longer sequences later are only included later. Loss and sequence length are common difficulty measures in CL.

use of sequence lengths (Tay et al., 2019; Martínez Alonso et al., 2017; Platanios et al., 2019) and losses (Kumar et al., 2010). The teacher schedules long sequences at first and only gradually weighs up short sequences later in training (see Figure 5.5c). Similarly, examples with low losses are introduced first, and higher losses are only weighted up afterwards (Figure 5.5b). Both of these results, the scheduling as well as the difficulty measures, are in line with what we would expect from the literature (compare Section 5.2).

Commentaries' performance is brittle We replicate the learning speed improvements that are reported in the original paper (see Figure 5.6a). In our GLUE setup, we find similar results (Figure 5.6b; for results on other GLUE tasks, see Appendix A.3.5). However, as we have observed brittle performances in the CFG experiments, we engage in an extended hyperparameter search. We find that improvements are limited to a certain set of suboptimal hyperparameters. As soon as we properly tune the hyperparameters, we learn faster by using the plain Adam optimiser without a teacher (for replication results with all datasets and models, see Appendix A.3.3). In all properly tuned settings, Adam, without curriculum, performs equally or better.

In summary, the commentary teachers' policy very well resembles other successful setups from the CL literature. Despite this, we also find that the curricula's benefits during the evaluation phase are not consistent: Changes in hyperparameters that should not strongly influence the effectiveness of the curriculum – such as changes in learning rate or batch size – erase any curriculum advantage. A proper hyperparameter search makes commentaries ineffective. Why is this the case, and why are the commentaries working in certain settings to begin with? To address these questions, we stop to analyse the teacher policy and have an in-depth look at how they actually produce their learning advantage.

Commentaries are data independent CL assumes that it matters at which point we train on which data point. We conduct an ablation experiment to see whether this is really what is driving the commentaries'



(a) 2-layer CNN (b) RoBERTa$_{\text{BASE}}$ (c) 2-layer CNN

Figure 5.6: The left side shows a 2-layer CNN (a) trained on CIFAR10 and RoBERTa$_{\text{BASE}}$ (b) trained on GLUE-MRPC respectively with and without a commentaries teacher (T). We see how the teacher improves learning speed for either model when trained with low learning rates (γ). However, there is no improvement when hyperparameters are chosen optimally. Figure (c) repeats the low γ data from Figure (a) but adds the ablated teacher from Section 5.3.2 in dark blue for comparison.

improvements. We replace the original weighting w_i – which applies an *individual* weight for each data point in a batch – by the batch average $\overline{w_i}$. This ablation erases not only the data dependence of the weights but also the distribution of the weights within a batch. Surprisingly, this ablation does not degrade the curriculum's performance (see Figure 5.6c) *at all*. The exact mapping of data points and weights is thus, apparently, irrelevant. The learning benefits must originate from the mere *shape* of the curriculum (i.e. the *curriculum structure*) by shifting from small to large weights with increasing i. We corroborate this intuition by conducting an additional small experiment with toy curricula that employ different simple weight shifts as their weighting policy:

$$
\begin{aligned}
T_{\uparrow\text{linear}}(i) &= \tfrac{i}{\kappa} && \text{– Increase } w \text{ linearly} \\
T_{\downarrow\text{linear}}(i) &= 1 - \tfrac{i}{\kappa} && \text{– Decrease } w \text{ linearly} \\
T_{\text{constant}}(i) &= 0.5 && \text{– Keep } w \text{ constant} \\
T_{\text{sigmoid}}(i) &= \sigma((i-\lambda) * \kappa) && \text{– Increase } w \text{ non-linearly}
\end{aligned}
$$

with κ and λ being constants and σ being the sigmoid function. We illustrate these toy policies and their performance on CIFAR10 in Ap-

pendix A.3.4. Interestingly, some of these toy curricula produce learning advantages akin to commentaries. In fact, effective curricula shift weights from smaller towards larger values throughout training, suggesting that such shifts are underpinning the success of the curriculum.

5.4 Curriculum-Adam interactions

We have seen how simple shifts from small to large loss weights can mimic the effects of the commentary curriculum. How is this possible? First, we know that the effect works across datasets, modalities and models and must therefore originate in the data- and model-agnostic optimisation process. In our case, optimisation centres around the Adam optimiser (Kingma and Ba, 2015). Second, the effective component in our toy curricula is the *change* of weighting with time. In the Adam optimiser, the only components sensitive to changes with time are the two momentum terms m_i and v_i. In the following, we will analyse the momentum terms of Adam (see Algorithm 1) to find a potential source of the learning advantages in commentaries.

Algorithm 1 Adam (simplified)

Inputs: γ (lr), β_1, β_2 (decay-rates), θ (parameters), $f(\theta)$ (objective)
initialise $m_i \leftarrow 0$, $v_i \leftarrow 0$
for $i \in \{1, \dots I\}$ **do**
  $g_i \leftarrow \Delta_\theta f_i(\theta_{i-1})$
  $m_i \leftarrow \beta_1 m_{i-1} + (1-\beta_1) g_i$
  $v_i \leftarrow \beta_2 v_{i-1} + (1-\beta_2) g_i^2$
  $\hat{m}_i \leftarrow m_i / (1-\beta_1^i)$
  $\hat{v}_i \leftarrow v_i / (1-\beta_2^i)$
  $\Delta\theta_i \leftarrow \hat{m}_i / (\sqrt{\hat{v}_i} + \epsilon)$
  $\theta_i \leftarrow \theta_{i-1} - \gamma \Delta\theta_i$
end for
return θ_i

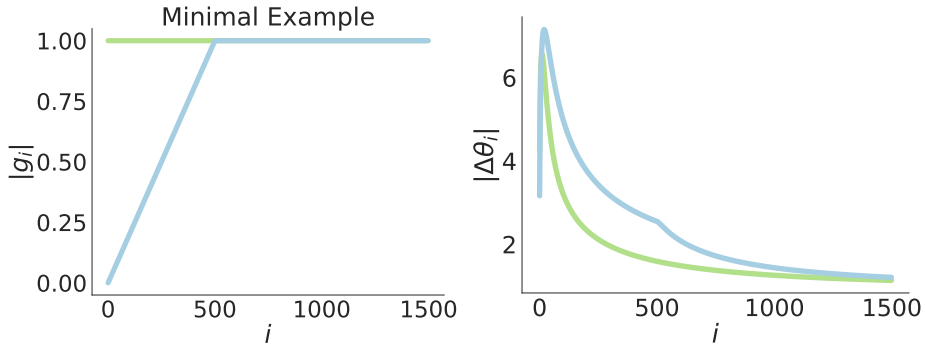
Asymmetric momenta In the Adam algorithm, both momenta, m_i and v_i, are determined by the current gradient g_i as well as their previous states (m_{i-1} and v_{i-1}, respectively). They are used to calculate the final parameter update $\Delta\theta_i$. For either term, the influences of past states are decayed at their own rate β_1 and β_2 (see line 5 & line 6). By default, β_1 and β_2 are set to largely different values[5]. Its progressive decay rate β_1 makes m_i more dependent on immediately preceding states, while v_i is largely influenced by more distant states. Both momenta are, therefore, asymmetric in their past dependence. To calculate the parameter update $\Delta\theta_i$ (line 9), the faster decaying term m_i is divided by the square root of the slower decaying v_i. This step is done to normalise the size[6] of the parameter-update $|\Delta\theta_i|$, and in a regular setup, the asymmetry of decay is irrelevant as the size of m_i and v_i remains (more or less) constant throughout training.

Interaction between momenta and curricula In our toy experiments (and in commentaries), we scale our losses (and therewith the gradients g_i) by w_i to become larger with time. If we systematically increase the size of g_i with time, $|m_t|$ grows faster than $|v_t|$. By normalising the m_t term by the therewith smaller v_t term, we artificially increase the size of the update $|\Delta\theta_i|$. There thus exists an interaction between the momentum terms and the shape of the curriculum. This effect is easy to empirically exemplify in a minimal example.

We consider a simple case with only a single parameter. We create two conditions: In the first condition, we linearly increase the gradient size $|g_i|$ from 0 to 1, where it levels off (similar to the linear toy curriculum). In the baseline condition, the $|g_i|$ remains fixed at the value of 1 (Figure 5.7a). For the first condition, the size of the update returned by Adam is systematically larger compared to the baseline condition (Figure 5.7b). We hypothesise that this scaling of $|\Delta\theta_i|$ is behind the observed learning improvements of

[5]Kingma and Ba (2015) recommend: β_1 = 0.9; β_2 = 0.999

[6]For simplicity, we refer to the l2-norm (calculate as $|v|_2 = \sqrt{v_1^2 + .. + v_n^2}$) of a vector as its 'size' throughout this and the following sections. Further, we simplify its notation to be $|v|$.



(a) (b)

Figure 5.7: Minimal example with a single parameter: If we increase the gradient of the parameter linearly (left), Adam produces larger parameter updates $|\Delta\theta_i|$ compared to a constant gradient size (right).

commentaries and our toy experiments. We can test whether this is true by checking the following two entailments:

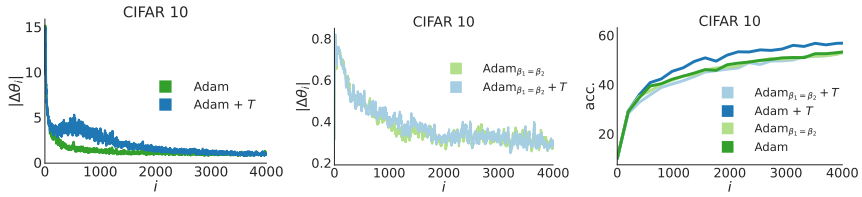
Entailment 1: The size of the update $|\Delta\theta_i|$ for commentaries is larger than for the baseline while w_i increases in size. Afterwards, $|\Delta\theta_i|$ drops to normal levels.

Entailment 2: Making m_i and v_i equally dependent on past $|g|$ by setting the decay-factors to $\beta_1 = \beta_2$ leads to the curriculum losing its effect.

We go on to empirically test these entailments for commentaries. Moreover, other curricula that cause systematic shifts in gradient sizes can result in similar effects. We, therefore, continue to test different other curricula.

5.4.1 Interactions with Commentaries

Experiments For the first set of experiments, we apply minimal necessary changes to the original setup of Raghu et al. (2021). We reutilise the



(a) (b) (c)

Figure 5.8: (a) Akin to the minimal example in Figure 5.7, the commentary teacher also produces larger parameter update $|\Delta\theta_i|$ due to Curriculum-Adam-interactions. (b) We can neutralise the Curriculum-Adam-interactions by setting Adam's β parameters to equal values ($\beta_1 = \beta_2 = 0.99$). With this intervention, the difference of $|\Delta\theta_i|$ that we observed in (a) vanishes. As a consequence, the performance of the commentaries' curriculum drops to the baseline (c). This shows how the interaction-dependent increase in $|\Delta\theta_i|$ is crucial for the learning speed gains of commentaries.

teacher model from 5.3.2 to train a new target model on the CIFAR10-dataset (Krizhevsky et al., 2009).

We test the first entailment by recording the size of the student's parameter updates $|\Delta\theta_i|$ and of the baseline model without loss reweighting during the training. Comparing the two, we find that the model with loss-reweighing experiences an increase in $|\Delta\theta_i|$ compared to training without a teacher (Figure 5.8a). The 'boost' in the update norm corresponds neatly to the range of iterations i in which w_i increases starkly (compare Figure 5.5a). Our observations are very similar to the minimal example described in Section 5.4 and are in line with **Entailment 1**. This experiment provides supportive evidence for our hypothesis, but it is not yet sufficient: the observed 'boost' could potentially arise from factors such as the enhanced properties of the optimization landscape, as discussed in Bengio et al. (2009).

We rule out such alternative explanations by eliminating the effect of the *Adam-curriculum*-interactions while keeping potential other effects of the curriculum unaffected. To do so, we equalise the past dependence of the

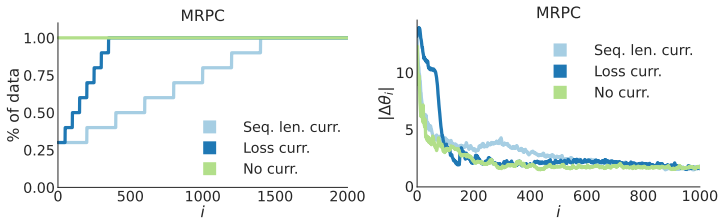
momentum terms by setting both of Adam's βs to the same value ($\beta_1 = \beta_2 = 0.99$). This results in Adam becoming equivalent to standard stochastic gradient descent (SGD) with a normalised momentum term[7]. We train an additional set of target models with this alternative hyperparameter setting. As a consequence, the difference in $|\Delta\theta_i|$ disappears (see Figure 5.8b) and the learning advantage in accuracy vanishes (Figure 5.8c). This verifies **Entailment 2**.

We have seen so far that the *Adam-curriculum*-interactions scale the parameter updates $|\Delta\theta|$. Doing so should ultimately have the same effect as increasing the learning rate γ (see line 10 in Algorithm 1). Hence, instead of using a curriculum, we can simply adjust γ. We show that this has the same effect by training three sets of target models (with and without loss-reweighting) with learning rates spanning three orders of magnitude. We find that only for very low values of γ, the compensating effect of commentaries helps learning (Figure A.8 in Appendix A.3.3). With a properly tuned γ, the difference between the baseline and commentary condition vanishes.

Conclusions We can summarise the results of our first set of experiments as follows: First, the effectiveness of the commentaries curriculum is a result of *Adam-curriculum*-interactions that scale parameter updates to become larger. Second, we can eliminate the effect of interactions by setting Adam's $beta$ parameters to equal values. This eliminates any learning advantage. Third, the observed learning advantages are only possible due to suboptimal hyperparameters; as soon as we set hyperparameters optimally, vanilla Adam outperforms the curriculum.

Automated approaches to curriculum learning are especially vulnerable to this interaction, as they can adapt their schedule function to optimally compensate for suboptimal hyperparameters. But is this a broader problem that can potentially affect any other CL setting? In what follows, we investigate the impact of Curriculum-Adam-interactions on other types of curricula.

[7]The βs can be chosen in the same way as the decay factor β in SGD



(a) (b)

Figure 5.9: On the left, we illustrate the schedule functions used for our experiments in Section 5.4.2. On the right side, we see the corresponding sizes of parameter updates $|\Delta\theta|$. We see an increase in parameter updates at the largest relative change of the data distribution.

5.4.2 Interactions with hand-crafted curricula

We now investigate other common hand-crafted and non-parametric curricula, such as pacing via sentence length or loss (e.g. Spitkovsky et al., 2010b; Platanios et al., 2019; Tay et al., 2019). These curricula do not have explicit shifts of gradient sizes from small to large. However, we have reason to believe that they might be affected by interactions with Adam nevertheless: We suspect that difficulty measures like **sequence lengths** (Spitkovsky et al., 2010b; Platanios et al., 2019; Tay et al., 2019) or **loss** (Kumar et al., 2010) are oftentimes correlated with the size of the gradients $|g|$ that they produce. When we finetune $\text{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019c) on a selection of GLUE-tasks, we find that this is indeed the case (a plot relating sequence lengths and losses to the size of the resulting gradients $|g|$ can be found in Appendix A.3.6). A curriculum that orders training examples according to these difficulty measures, hence, also implicitly orders them according to their gradient sizes. As a consequence, classical hand-crafted curricula potentially also trigger interactions with Adam. We will test such curricula for interactions in the following paragraph.

Experiments We implement two simple but common hand-crafted curriculum setups, which use (1) sequence length and (2) cross-entropy-loss

as difficulty measures and employ the discrete schedule functions shown in Figure 5.9a. Ahead of training, we order the training data according to either their sequence length or the losses obtained by a $\text{RoBERTa}_{\text{BASE}}$ model that we finetuned on the respective task. The curriculum randomly samples from an incrementally larger portion of the ordered dataset. We determined the hyperparameters of the schedule functions by conducting a grid search, determining the best-performing setup on a subset of the validation data. We then finetune $\text{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019c) with both an optimal and a slightly suboptimal learning rate on the MRPC-task from the GLUE-dataset (Wang et al., 2019c).

Table 5.3 reports results for the hand-crafted curricula. If the learning rate is low, both of our improvised curricula let RoBERTa learn much faster compared to training without curriculum (as shown by the performance after i = 750 steps). However, as soon as we increase the learning rate to an optimal level, vanilla Adam outperforms all other conditions. Analogously to our experiments with commentaries, we find the size of the parameter updates $|\Delta\theta|$ to be increased during the time of the largest change in data distribution (Figure 5.9b). The gain in $|\Delta\theta|$ for hand-crafted curricula is not as prolonged as for commentaries. This makes sense, as the shift in training distribution is especially large at the beginning of training, while in later steps, the relative change is neglectable. Despite gains in $|\Delta\theta|$ being relatively small and early in training, we observe that they are crucial for the performance gains of the curricula: If we eliminate the interaction with Adam by setting $\beta_1 = \beta_2$, the advantage of this simple curriculum vanishes (see Figure A.12c in Appendix A.3.7).

Conclusions In summary, we find that interactions between curriculum structure and Adam can also occur in hand-crafted curricula. This is the case if the difficulty measures are correlated with the gradient norms that they produce (e.g. if long sequences produce small gradients and short sequences produce large gradients). The interaction produces learning speed improvements when finetuning $\text{RoBERTa}_{\text{BASE}}$ with slightly suboptimal learning rates and, again, Adam, with optimal hyperparameter settings is able to outperform the curriculum.

SETUP		$i = 750$	CONVERGED
γ_{LOW} +	NO CURR.	77.8±1.5	88.2±0.6
	SEQ. LEN. CURR	82.8±1.1	87.8±0.5
	LOSS CURR	**84.2±0.51**	88.4±0.7
γ_{OPTIMAL} +	NO CURR.	**87.6±1.4**	90.1±0.3
	SEQ. LEN. CURR	83.2 ±3.5	89.1±1.4
	LOSS CURR	79.5±9.6	90.0±0.9

Table 5.3: MRPC-validation accuracies of $\text{RoBERTa}_{\text{BASE}}$ for hand-crafted curricula at an early stage ($i = 750$) and after convergence.

5.5 General discussion and conclusion

Summary In this chapter, we set out to use automated curriculum learning as an interpretability method. The goal was to better understand the connection of data features with the learning behaviour of language models. While the automated CL framework that we employed produced reasonable curricula, it turns out that — upon more rigorous investigation — the produced curricula actually produce no learning advantage. We show that non-functional curricula can be remarkably deceptive: the Commentaries curriculum closely resembles known curricula from the literature, even though it ultimately works for very different reasons.

Implications for CL in NLP Unfortunately, this chapter contributes little to the declared goal of the thesis. However, our findings still have important implications for the field of natural language processing (NLP): While curriculum learning has been successful in certain research areas (most notably in reinforcement learning; Narvekar et al., 2020), it has had mixed success in the field of NLP. In a very common setting of state-of-the-art NLP – consisting of language model pretraining and subsequent fine-tuning – curriculum learning has seen no success in the pretraining stage (e.g. Surkov et al., 2022; Campos, 2021) and only produced marginal

improvements in the fine-tuning stage (e.g. Xu et al., 2020). Here, we come to the conclusion that these mixed results might be related to the widespread use of the Adam optimiser (Kingma and Ba, 2015) in the field. Optimising a model using a curriculum in combination with Adam can lead to unintended interactions between the two. These interactions scale the parameter updates applied to the model, equivalent to a temporary scaling of the learning rate. Larger parameter updates lead to faster learning when hyperparameters (such as the learning rate) are chosen suboptimally (as shown for Raghu et al., 2021, and exemplary for common hand-crafted curricula). However, if hyperparameters are chosen correctly, vanilla Adam without curriculum always outperforms any curriculum learning approach that we employed. Our findings can fully explain the learning advantages attributed to the curriculum in all cases.

Implications for the learning process of LMs Besides the reported results, working on this chapter also produced the interesting observation that none of our attempts to create a curriculum for language modelling was successful: besides the automated approach, we worked on multiple alternative curriculum strategies. All of them failed without exception. This is in line with multiple previous attempts at creating curriculum learning strategies in language models (e.g. Surkov et al., 2022; Campos, 2021). We can relate this to our observations from Chapter 4: the learning process of language models is remarkably continuous. There, different linguistic concepts were acquired *gradually*, without any major shifts or clearly separable stages in the generalisation patterns. Such a continuous learning process likely does not require major shifts in the distribution of the learning data like a curriculum provides them.

5.6 Limitations

It is important to understand the limitations of the presented work to estimate its impact. We investigate interactions between Adam and general curriculum structures in multiple settings. From here, it is clear that in-

creasing the sizes of gradients will cause Adam to increase the sizes of updates. However, it is empirically impossible to reassess all different subtypes of curriculum learning methods in a single paper. We can, therefore, not tell how much previous research might be affected, and we caution rather to critically reexamine previous methods than to dismiss them. Further, interactions are directly dependent on the mechanics of the Adam optimiser, and there are no interactions to be expected in other popular optimisers, such as plain stochastic gradient descent or similar.

Chapter 6

ROBUSTNESS IN PROMPT-BASED LEARNING

In the previous two chapters, we have been investigating the pre-training process of LMs. In this chapter, we will direct our attention to learning dynamics in the adaptation phase of pre-trained models. While pre-trained LMs have remarkable capabilities, it is challenging to interface them reliably. We will investigate the data dependence of robust generalisation of pre-trained LMs, focusing on update-free adaptation methods such as in-context learning.

6.1 Introduction

In the previous chapters, we have been investigating the predominant type of learning in machine learning, which consists of iteratively updating model parameters to converge on a low empirical loss on some objective function. However, with the emergent ability of *in-context learning* (ICL) of large language models (LLMs), a completely new learning paradigm has gained prominence (Brown et al., 2020; Wei et al., 2022b). ICL is an alternative to updating model parameters for a specific task (from here on task tuning or *TT*) to interface the information that pre-trained language models accumulated in their parameters and adapt it to the task of interest

Zhou et al. (2023); Ouyang et al. (2022). ICL offers certain benefits: It eliminates costly, task-specific fine-tuning and provides greater flexibility, as a single model can be applied to many tasks.

The phenomenon is not yet completely understood. While there are already some hypotheses about the mechanisms behind ICL (compare Section 2.2.1), many aspects still require thorough research. For example, ICL outcomes suffer from instabilities that appear to result from non-robust generalisation. Still, it is not entirely clear whether they are similar to the weaknesses of TT models (for an overview, see Hupkes et al., 2023). As a result, ICL currently yields overall weaker performance compared to task-tuning and is less stable and reliable on many benchmarks (see, e.g. Bang et al., 2023; Ohmer et al., 2023; Min et al., 2022; Lu et al., 2022; Zhao et al., 2021). ICL has been suggested to be less susceptible to issues with out-of-distribution generalisation (Awadalla et al., 2022; Si et al., 2023). Other generalisation weaknesses appear to be behind the inconsistencies: for example, the format, order, or semantics of the provided in-context examples can have a considerable influence on the learning outcomes, as does the ratio of labels in the context and the exact labels used (Liang et al., 2022). Minor changes to the prompt can have unforeseeable consequences on the prediction outcomes (Khashabi et al., 2022). Further, it appears that the effects of different aspects of the prompt interact in their influence on the prediction (Wei et al., 2023; Yoo et al., 2022).

The learning dynamics of ICL appear to be complex, exhibiting many of the properties we listed in Section 1.2.2. We will, therefore, resort to an evaluation across many evaluation setups and a rigorous statistical analysis to get a better understanding of ICL's inconsistencies. In the experimental section of this chapter, we conduct a detailed exploration of vanilla and instruction-tuned LLMs across various shifts and setups to understand their robustness. We start with one of the prominent themes in robustness studies for TT models: robustness to spurious correlations between input and label distributions (Kavumba et al., 2019; McCoy et al., 2019; Niven and Kao, 2019) and find that in ICL, spurious correlations do not have a significant impact on learning outcomes. In a second set of experiments, we go on to investigate ICL's sensitivity to other features

of the adaptation context. To this end, we utilise the ICL consistency test (Weber et al., 2023), a contribution to the GenBench Collaborative Benchmarking Task (CBT; Hupkes et al., 2023). The test provides prompts for the same data points across many different setups and enables us to test a model's prediction consistency. Importantly, these 'setups' differ only in simple design choices (e.g. the formulation of the instructions given to a model) and do not change the nature of the tested task but how they are presented. A robust model should ignore these irrelevant changes to the prompt and make the exact prediction when confronted with the same data point across setups.

To better understand how different design decisions influence the prediction outcomes, we conduct a statistical analysis of the results and shed light on their inter-dependencies. Our holistic analysis reveals which exact design features in the in-context data trigger unreliable changes in the model predictions.

Outline The outline of this chapter is the following: We first present background literature on robustness issues and inconsistencies in TT (Section 6.2.1) and ICL models (Section 6.2.2). We then go on to evaluate in-context learners on data with spurious correlations between the input and target distribution – a known issue for TT learners (Section 6.3. In the subsequent Section 6.4, we present the ICL consistency test and evaluate it on eight different LLMs and include a detailed statistical analysis of the effects and interactions of different design choices.

6.2 Background and related work

In the following, we shortly define TT and ICL and then cover known problems with model robustness.

6.2.1 Task tuning

Task tuning (TT) describes the procedure of aligning a pre-trained model with a specific task via iteratively updating its parameters to minimise its prediction loss on some adaptation data. In our definition here, TT does not include finetuning on more abstract objectives like instruction tuning (IT; Wei et al., 2022a). TT models oftentimes fit spurious correlations between inputs and the associated labels that are idiosyncratic artefacts to the specific dataset (Niven and Kao, 2019; Kavumba et al., 2019; McCoy et al., 2019; Geva et al., 2019; Poliak et al., 2018; Gururangan et al., 2018; Kavumba et al., 2022) and do not align with the causal structure of the process that generated the data in 'the real world' (Schölkopf et al., 2012). Such adaptations (sometimes also referred to as 'shortcut solutions'; Geirhos et al., 2020) usually fail as soon as the data distribution shifts between the adaptation and test phase. Pretraining improves robustness compared to task training from scratch (Hendrycks et al., 2019, 2020). However, the necessary posthoc task adaptation still overfits spurious correlations (Niven and Kao, 2019). An effective way to mitigate issues in task adaptation is to expose the model to counterexamples of spurious correlations (Kaushik et al., 2020).

6.2.2 In-context learning

ICL describes the adaptation of a model to a task by inferring the task from the input given to the model. ICL can be subdivided into (1) few-shot learning, where in-context examples (consisting of input-output pairs) are given in the left-handed context of a tested input, and (2) zero-shot learning, referring to the case in which there are no examples. In this paper, we investigate few-shot scenarios.

In contrast to TT, ICL is a considerably cheaper adaptation method as it requires no parameter updates. Akyürek et al. (2022) and Garg et al. (2022) show that adaptation of transformer models via ICL exhibits the same degree of expressivity as simple linear algorithms, small neural networks or decision trees. While ICL emerges spontaneously with increasing size of untuned LLMs Brown et al. (2020), the ICL performance of such *'vanilla'*

LLMs lags behind the tuned state-of-the-art on almost all common NLP benchmarks (Liang et al., 2022).

Previous research has also shown that ICL is highly unstable. For example, the order of in-context examples (Lu et al., 2022), the recency of certain labels in the context (Zhao et al., 2021) or the format of the prompt (Mishra et al., 2022) as well as the distribution of training examples and the label space (Min et al., 2022) strongly influence model performance. Curiously, whether the labels provided in the examples are *correct* is less important(Min et al., 2022). However, these findings are not uncontested: Yoo et al. (2022) paint a more differentiated picture, demonstrating that in-context input-label mapping *does* matter, but that it depends on other factors such as model size or instruction verbosity. Along a similar vein, Wei et al. (2023) show that in-context learners can acquire new semantically non-sensical mappings from in-context examples if presented in a specific setup.

From this listing, we see that ICL entails many design choices, that task-unrelated design choices change prediction outcomes and that the effects of design choices do not exist in isolation. The field is only beginning to understand the complex interplays of different prompting setups.

6.3 Experiment I: Robustness to spurious correlations

In the current chapter, we clarify open questions about the robustness of in-context learners by shedding light on their sensitivity to factors to which they should be invariant (from here on *invariance factors*). First, we focus on one of the most prominent forms of non-robustness in TT models: susceptibility to spurious correlations between inputs and labels (Kavumba et al., 2019; McCoy et al., 2019; Niven and Kao, 2019). In the first set of experiments, we test how different models behave when spurious correlations are contained in their adaptation data.

6.3.1 Setup

Datasets We use different common NLU datasets (from here on *base datasets*), which are known to contain spurious correlations between input and label distributions (Gururangan et al., 2018; Geva et al., 2019; Poliak et al., 2018), as well as *adversarial datasets* of the same tasks. Adversarial datasets are designed to not contain the spurious correlations of the base datasets; then, they can be used to test whether models use short-cut solutions. Our base datasets span three different types of NLU tasks: natural language inference (NLI), paraphrase identification (PI) and extractive question answering (QA). An overview can be found in Table 6.1 and additional details about dataset properties and their construction in Appendix A.4.3.

Task	Base dataset	Adversarial dataset
NLI	MNLI (Williams et al., 2018)	HANS (McCoy et al., 2019)
		ANLI (Nie et al., 2020)
PI	QQP Wang et al. (2017)	PAWS (Zhang et al., 2019)
QA	SQuAD (Rajpurkar et al., 2016)	SQuAD adv. (Jia and Liang, 2017)
		adv. QA (Bartolo et al., 2020)
		SQuAD shifts (Miller et al., 2020)

Table 6.1: Tasks and corresponding datasets as used in Section 6.3.

Models Our first experiment compares TT models with models that perform tasks through ICL. For the latter, we consider two types of models: '*vanilla*' LLMs, and LLMs that are tuned to follow instructions (*IT* see e.g. Wei et al., 2022a; Zhong et al., 2021).

For TT, we use models based on $\text{RoBERTa}_{\text{BASE}}$ and $\text{RoBERTa}_{\text{LARGE}}$ (Liu et al., 2019c). If available, we reutilise finetuned versions of RoBERTa that have been open-sourced through the huggingface hub (Wolf et al., 2019); if not available, we finetune the respective models ourselves (with training details in Appendix A.4.2).

Our vanilla LLMs consist of the series of LLaMA models (7B, 13B, 33B, 65B; Touvron et al., 2023). Based on the same LLaMA models but

Type of learning	Model
TT	RoBERTa-base
	RoBERTa-large
ICL + vanilla	LLaMA 7B, 13B, 30B, 65B
ICL + Instruction-tuning	Alpaca 7B, 13B, 30B, 65B

Table 6.2: Adaptation types and the respective models, as used in Section 6.3. We use the same ICL models in Section 6.4.

additionally fine-tuned via low-rank adaptation (LoRA; Hu et al., 2022) on the alpaca self-instruct dataset (Taori et al., 2023; Wang et al., 2023), we use the freely available 'Alpaca' models as IT equivalent. We run all models using mixed-precision decomposition as described by Dettmers et al. (2022). For an overview of all used models, see Table 6.2.

Evaluation We evaluate ICL models by concatenating the target example x and with k labelled in-context examples and hard-sample from the probability distribution over possible labels $y \in \mathcal{C}$ using

$$argmax_{y \in \mathcal{C}} P(y|x_1, y_1 ... x_k, y_k, x)$$

where $\mathcal{C}$ is the set of possible labels. Every data point x is wrapped by an *instruction* template. Instructions explain the task the model should solve in natural language. The label space $\mathcal{C}$ is determined by the type of instruction template and can differ across templates. We mitigate the influences of the template format, order of (x_i, y_i), imbalanced distribution of y_i and semantics of x_i by a pseudo-random sampling x_i for every new inference in which we ensure that for every inference the in-context labels y_i are balanced over all possible labels (similar to Wei et al., 2023; Brown et al., 2020, inter alia). Moreover, we use multiple instruction templates sourced from FLAN Wei et al. (2022a) to avoid systematic bias.

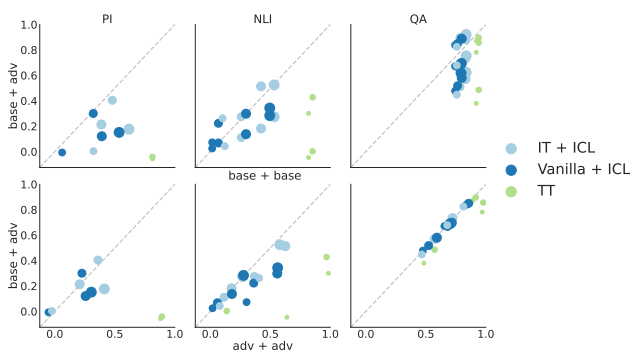
6.3.2 Results

We here evaluate the generalisation capabilities of in-context learners under *covariate shift* between the adaptation data (finetuning data in TT and in-context data in ICL) and the test data (compare GenBench; Hupkes et al., 2023).

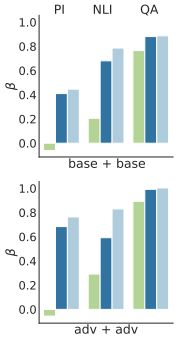
Base data in-context First, we adapt the TT and ICL models on the base data and then compare their performance between the base data and the respective adversarial counterparts. If an approach is robust to spurious correlations in the adaptation data (which are the fine-tuning data or in-context examples, respectively), it should perform approximately equally on the base dataset and the adversarial dataset. We relate both scores in the first row of Figure 6.1.

Results from in-context learners land generally closer to the diagonal, hence indicating – despite overall weaker performance – that they are more robust to the spurious correlations in their adaptation data. To quantify this visual result, we fit a linear regression model on the data presented in the scatterplot in Figure 6.1a (hence, predict the adversarial- from the base accuracies) with the intercept fixed at $\beta_0 = 0$. The coefficient β_1 can then be interpreted as a degree of robustness to the different adaptation data, with $\beta_1 = 1$ indicating complete robustness and $\beta_1 = 0$ complete reliance on non-generalisable patterns in the base data. The β_1 values for different adaptation types can be found in the top row of Figure 6.1b. The β_1 values across all tasks are significantly closer to the parity value of 1 for ICL models than for TT models, with IT models having the edge over vanilla models.

Our results demonstrate that ICL models are much less sensitive to spurious correlations in their adaptation data than TT models. However, the fact that ICL models do not reach the parity value of 1 means that gains on adversarial data are smaller compared to gains on the base data. This suggests that ICL may still be mildly sensitive to spurious correlations, or, alternatively, that the adversarial datasets used are simply inherently more



(a)



(b)

Figure 6.1: Figure (a) shows the f1-scores of different models – normalised for random accuracy – on different data sets when adapted via base or adversarial data. On each y-axis, we plot accuracy under distributional shift (*base* + *adv*), while on each x-axis, there is no shift (*base* + *base* and *adv* + *adv*). Each column shows a different type of task. Marker size represents the model size, and colour represents the type of task adaptation. Figure (b) shows the β-parameter of the linear regression (fixed intercept) on the data of Figure (a). We fit a linear regression for each task and adaptation type separately. Values close to 0 indicate very strong sensitivity to adaptation data, while values close to 1 indicate no sensitivity.

difficult, resulting in lower performances compared to the base data[1]. We will further explore this question in the next experiment.

Adversarial data in-context As a follow-up experiment, we consider what happens when the adaptation data contains adversarial examples. As those examples do not contain the same spurious correlations, models cannot overfit them (Kaushik et al., 2020). This should not make a difference for models that are robust to spurious correlations, but we expect a performance drop between these two conditions for models that learned

[1]An illustrative example of the base data being easier: adversarial QA contains only a single answer alternative while squad contains three.

solutions that exploited those correlations. As we are now evaluating the adversarial data points in both scenarios, we eliminate the potential impact of the dataset difficulty on the scores. In the second row of Figure 6.1, we plot performances with base adaptation examples in the context against the performance with adversarial adaptation data, noting that ICL models are mostly unaffected by adaptation data type while TT models land far underneath the diagonal again. A regression analysis shows almost all β-values of ICL models moving closer to parity, showing us how the dataset difficulty impacted the results. However, even without the effect of dataset difficulty on the β-values, they are still not quite equal to 1, suggesting that the type of adaptation data *has* a small influence on ICL learners.

6.4 Experiment II: Consistency evaluation in ICL

In the previous section, we saw that the inconsistency of in-context learners is likely caused by other factors than by spurious correlations in the in-context data. Although previous studies have reported the susceptibilities of LLMs to various factors, the impact of different design decisions and their interactions in the context of ICL robustness has not been systematically evaluated. Here, we test the effects of an extensive range of these factors on prediction outcomes in consistency and accuracy. To that end, we present the *ICL consistency test*. The ICL consistency test combines the same data points with a large set of different setups and compares model predictions across them. Subsequently, we follow up with a large-scale evaluation and analysis of the consistency of 8 ICL models.

6.4.1 Setup - *The ICL consistency test*

We will here present the ICL consistency test. We will first explain our rationale for constructing the test in the following **Motivation** paragraph. In the paragraph **Data**, we will explain which resources the test uses, such as the *instruction templates*, the *data sets* and how these resources are used.

Then, we will dive into the composition of **Factors** into **Setups**. Ultimately, we show which **Metrics** we use to estimate a model's consistency.

Motivation The ICL consistency test evaluates a model's ability to make consistent predictions on the same data point, independent of the respective evaluation setup. To do so, it compares a model's prediction across many different prompting setups. We define setups through the presence or absence of different binary *factors*, which are simple choices in the prompt design (e.g. do I use instruction A or B to prompt the model).

The motivation for this is the insight that consistency measures are complementary to accuracy: imagine a scenario in which a model is evaluated with two different, but equally valid, setups. For example, one could query a model for the sentiment of a sentence `<x>` using either of the following instructions:

Instruction 1 Please state whether the following sentence is positive, negative, or neutral: `<x>`

Instruction 2 Given the sentence: "`<x>`", please classify its sentiment as positive, negative, or neutral.

While both prompts are superficially different, their conveyed query is exactly the same. Let's assume that the model predicts the same proportion of correct labels in either setup but does so on a different subset of the evaluation data. The accuracy score has the same value in either setting and, therefore, could let us assume that we have to improve the model's ability to solve the task at hand. In reality, however, the main issue is the model's questionable generalisation and lack of robustness to irrelevant changes in the prompt. We have seen in the background section that prompt-based learners lack this type of robustness more often than not. It is, therefore, crucial for accurate error analysis to have a tool to estimate reliability by systematically evaluating a model's consistency.

Data We use different freely available and established data sources to construct the ICL consistency test. Instructions explain in natural

language to a model which task it should solve and wrap the original input x from a given data set. For **instructions**, we use different subsets of the crowdsourced *promptsource templates* (from here on 'P3'; Bach et al., 2022), with the exact template being used depending on the specific setup that is evaluated. Exact information on which instructions are employed is given in the following paragraph, *'Setups and factors'*. We use the p3 instructions templates to wrap **data** points from the ANLI (Nie et al., 2020) and MNLI (Williams et al., 2018) datasets. For each of the datasets, randomly draw a subset of 600[2] data points from the respective validation sets and – in the case of ANLI – we draw to equal parts from the validation sets of its three distinct subsets. We provide solved examples in the left-handed context of the model input as an aid for the model to infer the task it has to solve (as done in Brown et al., 2020). These **in-context examples** are constructed in the same manner as the target examples but have their ground truth label concatenated. To select in-context examples, we randomly draw data points from the respective full training sets. The label space, the instructions, the number or even the task of in-context examples can, again, differ depending on the specific setup that is evaluated. Examples of prompts can be found in Appendix A.4.5.2.

Setups and factors We estimate the robustness of a model by evaluating the consistency of its prediction on the same data point across many different setups. We define each setup through the absence or presence of each of a range of binary factors λ. We include the nine factors listed in Table 6.3 in our test[3].
Besides the first seven data-related factors, we also augment the ICL consistency test with two additional model-related factors using the code implementation submitted to the GenBench CBT (for details, see Appendix A.4.6). These additional factors make it possible to relate specific robustness issues to specific models or evaluation types. Arranging the

[2]We found 600 examples to yield sufficiently similar results to evaluating the whole dataset, tested on a small subset of setups

[3]For more detailed explanations on the different factors and the respective motivation to include them, we refer to Appendix A.4.7

Factor	Description
n-shots	Many (k = 5) or few (k = 2) in-context examples in the prompt.
Instruction quality	Two groups of semantically equivalent but *differently* performing instruction templates (high- vs. low-performing; more details in the paragraph 'Probing instructions').
Balanced labels	In-context examples with labels balanced across all classes or randomly sampled examples.
Cross-templates	Randomly drawn in-context instructions from all available P3 templates or the same instructions as target.
Cross-task	In-context examples from another task (QQP; Wang et al., 2017) or from the same task as the target (ANLI / MNLI).
Instructions	Semantically equivalent target instructions that perform *similarly* (more details in the paragraph 'Probing instructions').
One label	In-context examples with a single randomly selected label or randomly selected in-context examples.
Instruction tuning (Model)	Models are either instruction-tuned or not ('vanilla' models).
Calibration (Model)	Calibrate model outputs using *content-free* prompts following Zhao et al. (2021) or not.

Table 6.3: Factors used to create setups

listed factors in all possible combinations results in 1536 setups. Combining the 1536 setups with our randomly sampled 600 data points results in 921_600 samples that we will evaluate.

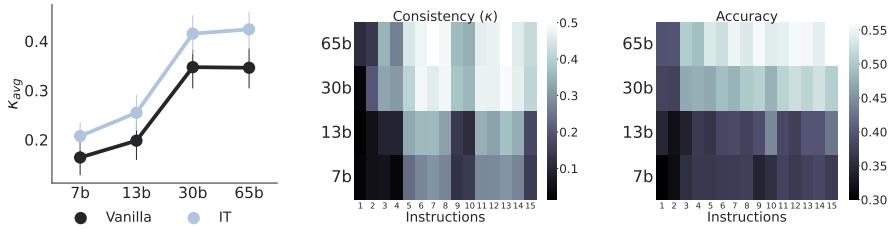
Metrics The ICL consistency test uses consistency and accuracy metrics. Their main features are explained in Table 6.4.

Metric	Description
Cohen's κ	We measure the consistency of model predictions using Cohen's κ (Cohen, 1960), a measure of interrater agreement adjusted for agreement by chance. The metric κ equals 1 if two (or more) sets of predictions perfectly align while agreement by chance results in κ equalling 0. In our case, we calculate κ to compare the predictions of a model before and after we change the value of a factor λ across all possible setups. For example, we take the predictions from all setups in which in-context examples have the same label (the factors `one label` is present) and compare it to the case in which we have different labels for the in-context examples (the factors `one label` is absent). With all other factors being constant, we can estimate how much this factor changed the model prediction (or, inversely, how robust a model is) by calculating κ. To ensure meaningful scores, we mask out all predictions that are not within the label distribution of the respective task. Finally, we get the overall model consistency κ_{avg} by averaging across the κ values of all factors.
Main effects of λ	Next to κ – the primary metric –, we also provide the auxiliary metric in the form of the main effects of factors. The main effects show how much the presence or absence of a factor influences the accuracy of the model on average. The main effects of the factors help to interpret their κ values: Does the change in prediction occur because the factor actually improves the model accuracy or is it due to model inconsistency? To obtain measures of the main effects, we fit a simple linear regression model to predict accuracy scores from the presence or absence of each factor $Acc_\pi = \beta_1\lambda + \beta_0$. We can then interpret the coefficient β_1 of λ as its main effect ('How much does the factor on average change accuracy scores?').

Table 6.4: Metrics used in the ICL consistency test.

Probing instructions To find a set of high- and low-performing instructions for the `instruction quality` factor, we run a preliminary analysis where we probe model behaviour in response to all 15 available P3 ANLI instructions. We assess the performance of different instructions based on accuracy and consistency.

We first get a general picture of the average consistency κ_{avg} of each model π across all templates. We find that κ_{avg} increases with the number of parameters and is overall higher when a model has been instruction tuned (Figure 6.2a).



(a) (b) (c)

Figure 6.2: Figure (a) shows the consistency of a model π when used with all 15 different P3 instructions, in an otherwise fixed setup. A value of 1 indicates no consistency; Figure (b) shows how consistent individual instructions are with all other instructions. A value of 0 indicates a complete change of predictions while a value of 1 indicates perfect agreement; Figure (c) shows the respective accuracies of the instructions in Figure (b).

We then consider the consistency of each individual instruction and find a congruent pattern of consistency across all models (Figure 6.2b) that corresponds generally to the accuracy scores of the same instructions (compare Figure 6.2c). Interestingly, we also find two groups of high-accuracy instructions making very different predictions (see the consistency scores of 9, 10 and 15 vs. rest). Based on these observations, we choose the two highest- and lowest-performing instructions to constitute the `instruction quality` factor and templates 14 and 15 as realisations of the `instructions` factor.

Experimental details To remain within reasonable computational cost, we focus our analysis on the ANLI dataset Nie et al. (2020). To structure the subsequent analysis, we also divide the factors test into two groups: Firstly, factors that constitute interventions to improve consistency and

performance and, hence, from which we want a model to *change* their response when we change their value. We will call these **variance factors** or λ_{var}: These factors are `n-shots`, `Instruction quality`, `Balanced labels`, `Instruction tuning` and `Calibration`. Secondly, factors from which we want a model to *not change* their response (or 'be robust to') when we change their value. We will call these **invariance factors** or λ_{inv}: `Cross-templates`, `Cross-task`, `Instructions` and `One label`.

6.4.2 Results

We now evaluate the LLMs from Experiment I (see Table 6.2) on all possible combinations of λ_{var} and λ_{inv}. Appendix A.4.8 shows the distribution of accuracy scores across all runs for different models. The spread of scores is strikingly wide, with the large models scoring from below chance to up to 67% accuracy, depending on the overall setup. This extreme variability underlines the importance of better understanding the impact of different design decisions and prediction consistency in ICL.

The analysis of the results will be partitioned into two parts: First, we will look at the main effects, i.e. how much does a single factor impact the consistency and the accuracy across many setups? Afterwards, we will investigate interactions, i.e. when we disentangle the main effects, do we find systematic interactions across pairs or triplets of factors? The subsequent section comprehensively summarises the results of our statistical analysis.

Main effects Figure 6.3 presents the main effects separated by model size, illustrating the impact of each factor in isolation.
The variance factors we chose are generally thought to improve accuracy and hence should have positive main effects. We find two out of five *variance factors* significantly improve performance on average, from which `instruction quality` stands out as the most influential factor across all model sizes. Similarly, we find that `instruction tuning` is consistently beneficial while `balancing` the in-context labels and the number of in-context examples (`n-shots`) have on average small, non-significant

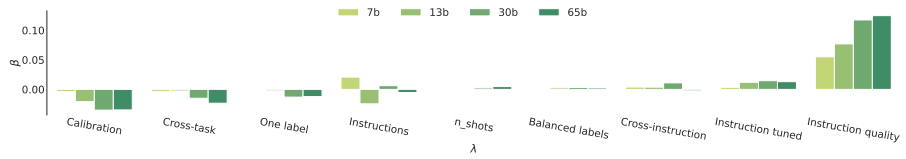


Figure 6.3: The β-values of the main effects of each individual factor across many different runs. The values can be directly interpreted as '*expected accuracy gain/loss*' when a factor is present compared to when it is absent.

effects. Surprisingly, `calibration` harms rather than helps performance for all but our smallest model.

Different from variance factors, invariance factors are chosen such that they should not influence a robust model's predictions. Accordingly, the main effects should be optimally close to 0. We find that models are generally robust to having varied instructions in-context (`cross-instruction`) or even having a slightly positive effect. This is intriguing, as this factor entails considerable changes to the in-context setup, and we previously saw how the type of *target* instructions plays a crucial role. Otherwise, we identify vulnerabilities of large models to the factors `cross-task` and `one label`. The ambivalent effect of the `instructions` factor suggests high volatility.

These main effects give us a general idea of the tendencies of factors. To better understand all main effects, we will investigate interactions in the following paragraph.

After considering the accuracy-based results, we now also look at the prediction consistency κ of the factors (as defined in Table 6.4) The κ score shows us the degree of robustness of a model to an invariance factor by quantifying the degree of prediction change when a factor is changing. We see in Figure 6.4 how robustness increases with size and instruction tuning. The very low κ scores for the detrimental `cross-task` factor come as no surprise, while low scores in the `instructions` factor corroborate the previous suspicion that instructions are highly volatile: If we change the type of `instructions` we use, the predictions across a lot of setups

change.

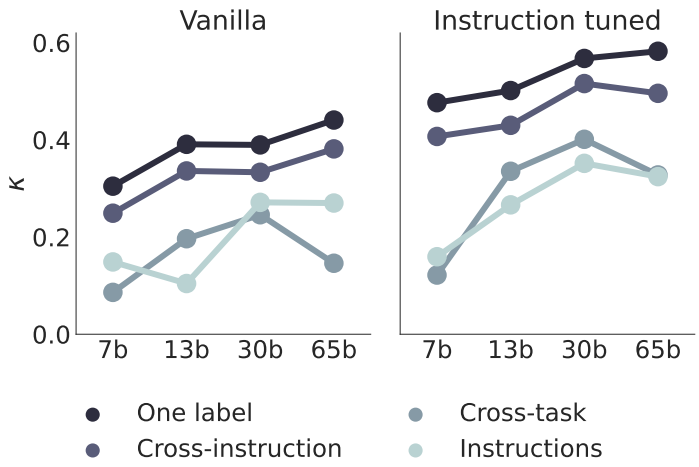


Figure 6.4: The consistency values across all other factors comparing predictions when a specific factor is present or not. A value of 0 indicates a complete change of predictions, while a value of 1 indicates perfect agreement. Hence, a low value indicates that a model is not robust to a change in a specific factor.

Interactions The main effects give us a good idea of the general direction of the impact of a single factor. However, the main effects do not tell the whole story: Consider the case in which factor `A` improves performance if it is paired with factor `B`, but performance deteriorates when paired with `C`. `A`'s overall main effect might be close to zero even though it influences certain settings. To better understand the impact of each factor, we will have to investigate its interactions.
To analyse interactions, we fit a factorial ANOVA considering the effect of all possible 2- and 3-way interactions[4] on the accuracy of predictions. We

[4]We exclude the *instructions* factor because the independence of *instruction quality* is not given. Moreover, we adapt the significance levels via Bonferroni correction for multiple comparisons ($\alpha < 0,00059$) and show only significant interactions.

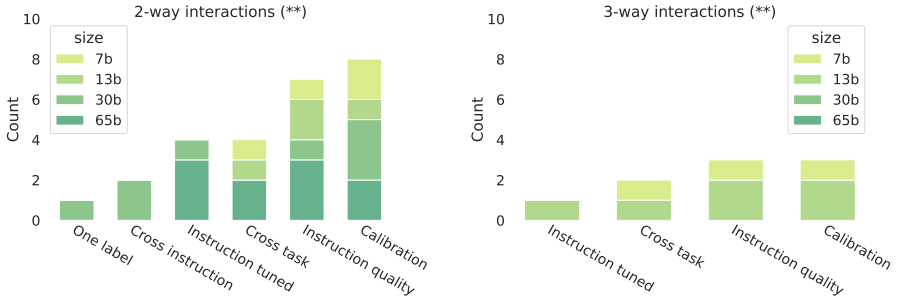


Figure 6.5: The number of interactions per factor with other factors. A large number of interactions means that the outcome of a change in these factors depends on a lot of other variables.

then count the number of significant interactions every factor maintains with other factors. A larger number of interactions suggests that a factor is volatile, changing predictions depending on the overall setup. Further, a large number of interactions for a single model suggests non-robustness as we generally assume the factors to be orthogonal. On the other hand, if factors are not interacting, we can interpret their main effects directly.

Figure 6.5 shows the number of interactions that each factor maintains. A general observation is that large models tend to have simpler 2-way interactions, while smaller models tend to have more complex 3-way interactions. We, most importantly, find that the impact of the `instruction quality` and of the `instructions`[5] themselves are very sensitive to the setup. This demonstrates the intricacy of the factor: The `instruction quality` has the largest positive impact on prediction outcomes, but at the same time, the instructions are highly interactive and volatile, with their effects depending on the setup in which they are used.

[5]We fit another ANOVA excluding `instruction quality` while keeping `instructions` as a factor to ensure that the effect is not only due to large performance differences between the two realisations of instruction quality. We find similarly strong interactions for the `instructions` factor (see Appendix A.4.9).

Otherwise, we observe that `calibration` is the most volatile, with 8 significant interactions with other factors. The previously observed main effect has to be seen in this perspective: `calibration` is not generally detrimental, but its effects depend very much on the setup in which it is used. For example, we find on closer inspection that `calibration` leads to the highest overall accuracies for the 7B parameter models when presented with specific `instructions` and paraphrase identification in-context examples (`cross-task`).

On the other end of the spectrum, we find that factors like the number of in-context examples (`n-shots`), the `balancing` of in-context labels or using a `one label` who little to no interactions at all. Conveniently, we can therefore interpret their main effects directly, as they are most likely to be stable across setups. For example, suppose it is possible to increase the number of examples in the context. In that case, we can reliably expect small gains in accuracy without the danger of otherwise interfering with the learning process. Similarly, balancing labels leads to reliable small improvements and having just a single label in the context reliably reduces accuracy for large models.

6.5 General discussion and conclusion

Summary In this chapter, we investigated the properties of learning without parameter updates (i.e. in-context learning), more precisely, the consistency and stability of this learning approach. We first evaluate ICL's sensitivity to spurious correlations and find it to be more robust than TT previously. In the second part of the chapter, we present a new test for robustness in prompt-based learning, the ICL consistency test, and subsequently conduct a comprehensive analysis of the influence of different setups on predictions of ICL models.

Findings While ICL learners are not sensitive to spurious correlations, the issue of robustness is not resolved. Insignificant changes to the prompting setup can lead to unpredictable changes in the model output. We show

that depending on the evaluation setup, ICL accuracy in our experiments differs up to 40%, and the primary metric κ reveals that none of the tested models performs with high consistency for any minimal setup change (i.e. across any change in factors). Considering different setups, our analysis shows that choosing adequate `instructions` promises the largest performance gains across many setups. At the same time, `instructions` are among the most volatile factors of all: they are very sensitive to the setting in which they are used and interact with most other factors. On the other hand, we show that factors that relate to the exact organisation of the in-context examples, such as the label distribution or in-context instructions (`cross-instructions`), have surprisingly small impacts. Factors like `n-shots` – among others – are not interactive, which makes them much easier to handle: their expected gain or loss should, in most cases, correspond to our observed main effects. The results indicate that the quality of the generalisation of ICL in LLMs can be improved: If predictions are consistent, the model correctly disregards irrelevant context information; if it is inconsistent, it lets irrelevant context information influence its predictions.

Implications With respect to the dissertation, the results in the current chapter confirm the need for more holistic thinking in model analysis and interpretability: Previous research concentrating on single factors in the evaluation setup tends to find positive results in a specific setup but does not validate them across alternative setups, just to be refuted by a follow-up study (as we have seen in work of Min et al. Min et al., 2022, which was quickly followed by relativisation in Yoo et al. 2022 and Wei et al. 2023). We show that ICL is ruled by complex dynamics and interactions instead of simple linear relations between input properties and how an analysis which keeps this complexity in mind yields valuable insights. It might be advised to use more diverse evaluation setups and a rigorous statistical analysis of the results to guarantee the generality of results and avoid Type-I errors in publications (Ioannidis, 2005).

What do these findings imply for the field of NLP? To get hold of inconsistent predictions in ICL, finding the exact properties of instructions

that so strongly influence model predictions is a sensible next step (potentially with a similar methodology as it is presented here). Insights into the impact of instruction properties can help us to find the source of inconsistencies and avoid them in production, while they can also contribute to the theoretical understanding of in-context learning which is currently still under investigation. While our analysis focused on the few-shot setting, it also significantly impacts the increasingly popular zero-shot learning, as instructions are central in that setting. For model deployment, our findings demand caution as minor changes to certain parts of prompts (e.g. the instructions) can change the performance of the general setup. This is especially true for employing smaller, untuned models. A consistent finding across all our experiments is that instruction tuning improves consistency and robustness to irrelevant factors across all setups. Therefore, we advocate for the use of tuned models to improve robustness.

From anecdotal evidence, we can conclude that adaptation methods like RLHF (as we discussed them in the general background section 2.2.1) are improving the consistency of ICL learners even more than instruction tuning. This hints at the special dynamics of RL that we are not able to achieve with regular gradient-based methods. With more research into these currently still nebulous dynamics of RL, it might be worthwhile to investigate whether their advantages can be emulated by less expensive gradient-based methods.

6.6 Limitations

The research presented in both experimental sections has several limitations. For the first set of experiments in Section 6.3, the comparison between TT models and in-context learners is not 'fair'. Model sizes are not comparable, the amount of adaptation data differs significantly (thousand for task-tuning compared to 5 for ICL), and some of the adversarial datasets were created with some of the TT models 'in-the-loop' (e.g. ANLI). However, our motivation here is not to be fair but to show practically relevant effects in either type of task adaptation. For a fair

comparison, see Mosbach et al. (2023).

For the second set of experiments in Section 6.4, we only consider a subset of factors that we deemed the most relevant or interesting. Albeit we consider our choice of factors appropriate, they are in no way exhaustive. Adding more factors would enrich the analysis. However, the number of inferences to compute grows exponentially with the number of considered factors, which sets a limit for the number of analysed factors. However, we think that the performance on the ICL consistency test can be a good indicator of the quality of the generalisation that an LLM is making. For potential follow-ups, we suggest a more fine-grained investigation of different instruction designs for the target example, as this potentially yields interesting insights on what exactly leads to high-performance gains and large volatility. Our study is coarse in this aspect. Our analysis was further hampered by the decision to use the relatively 'hard' ANLI dataset to run our evaluation: smaller models produce very low ICL accuracies for hard datasets like ANLI across many factors and therefore provide little variance for meaningful analysis.

Chapter 7

GENERAL DISCUSSION AND CONCLUSIONS

In the following, I first summarise the findings of the individual chapters and how they map onto the research objectives from Section 1.3. Then, I will put all individual findings into a broader context and discuss my work's contributions to the fields with which it intersects. Finally, I will provide an outlook on how the intersecting fields will develop, and future work can potentially build productively on my contributions.

7.1 Revisiting the chapters

Chapter 3: Generalisation and linguistic theory

Summary The work in Chapter 3 is a principal step towards the primary goal of this dissertation in that it formulates a framework to understand the learning dynamics of language models in a holistic and unconstrained way. We show how similarity relations from linguistic theory connect to the language modelling generalisation behaviour: concepts that are considered similar in linguistic theory can be used by the language model to generalise and share structure across different data points: throughout the learning process, the model uncovers regularities (or 'similarities')

among data points unrelated to the surface structure and learns to pool these data points into the same concept. In this way, we can analyse the self-organisation ability of the language model without interfering with the dynamics of the complex system (Section 1.2.2). Our experiments show how language models generalise different realisations of the same concept (i.e. licensing of NPIs). Interestingly, suppose we expose a model just to one realisation of a concept instead of a broader range of them (i.e. we prevent generalisation). In that case, the model is performance comparably much poorer, even on the presented realisation itself.

Contextualisation within the literature The work in this chapter touches upon different ideas in cognitive science, machine learning and theory of learning: The idea of a 'linguistic similarity space' is closely related to the notion of 'conceptual spaces' (Gardenfors, 2004, 2014) and ideas of geometric representations (Kriegeskorte and Kievit, 2013) in the cognitive sciences. Our framework can be understood as an approach to realise the idea of 'conceptual spaces' of language via computational means. On the other hand, the generalisation of linguistic concepts can be understood from the perspective of learning as compression. Learning as compression is the idea that efficient learning compresses information in the input data efficiently to a low-dimensional representation (see Section 2.2.2. Finding similarities in linguistic structure and learning linguistic rules to exploit them can be seen as an efficient strategy for compressing language data into a low-rank representation.

Limitations The main limitations of this chapter are implementational: We apply undersampling of a specific linguistic concept (i.e. eliminate it from the training data of the language model) to investigate the effects on other language concepts. This approach is problematic in two ways: First, it is computationally expensive to train a new language model on the modified training data to investigate its effects on other concepts. If we plan to construct larger language spaces encompassing many different language concepts, this approach is not viable. Besides the computational cost, it is also challenging to filter training corpora for certain language

concepts, as frequently, these concepts are not observable in the surface form of a sentence.

Further, the work presented in Chapter 3 does not yet allow any statement about the *learning process* of language models. The similarity of different linguistic concepts within the model should change throughout training as the model starts to understand the data distribution better. More efficient (i.e. abstract) compression rules might emerge, and generalisation patterns might change. Understanding these learning processes is a declared goal of this dissertation.

We address these three limitations in Chapter 4.

Chapter 4: Linguistic task-spaces

Summary The work in Chapter 4 extends and improves upon the framework we introduced in Chapter 3. The pivotal idea here is that by looking at the generalisation behaviour of an LM across different linguistic tasks, we can deduce which tasks share structure (or are 'similar' to each other). If we apply this idea to many tasks simultaneously, we can construct a 'linguistic similarity space' representing an LM's conceptualisation of language. We resolve major issues of the undersampling approach from the previous chapter by replacing it with oversampling and a technique to isolate linguistic phenomena from their entanglement within natural language. This enables us to fine-tune them selectively despite their latent nature and follow established methods from MTL to obtain *behavioural* similarity space based on the transfer learning across linguistic tasks. We further produce *structural* similarity estimates by analysing their shared parameters and alignment. We use the similarity spaces to investigate the learning process of LMs and discover that their processing of linguistic tasks becomes more distributed and interconnected with training. We also uncover that LM learning is remarkably continuous, where most linguistic similarity is discovered early in training, and this pattern is merely reinforced later on. We do not observe any larger shifts in the observed patterns, as we would expect from human-like learning. We also find that large models are faster in uncovering linguistic similarities and also

form more general concepts of phenomena (shown through the higher within-phenomena generalisation).

Contextualisation within the literature The approach and the findings in this chapter connect with different branches of research. I will shed light on the four most interesting connections. First and foremost, our work connects with previous research in MTL (Caruana, 1993, 1997). In MTL, the goal is to exploit shared structure across tasks to improve the data efficiency and performance on the involved tasks. To do so, researchers have been estimating the shared structures in tasks via different *similarity* measures for many years (Ben-David and Borbely, 2008). In the domain of computer vision, such similarity estimates have been used to create taxonomies of tasks (or 'task-spaces' Zamir et al., 2019a; Standley et al., 2020b; Achille et al., 2019). While there have been many studies using similar methodologies on LMs (Chowdhury and Zamparelli, 2019; Prasad et al., 2019; Pérez-Mayos et al., 2021), in this chapter, we constructed for the first time large-scale linguistic spaces based on many different linguistic tasks. We further construct similarity spaces based on *structural information* by identifying task-relevant parameter subspaces and relating them via their gradient alignment, a method inspired by Yu et al. (2020).

Secondly, the way we suggest linguists interpret linguistic task spaces has many similarities to the idea of Gardenfors (2004, 2014)'s notion of conceptual spaces. Using language model representations as a conceptual space of language is an established idea, especially in the realm of lexical semantics (Baroni and Lenci, 2010; Mikolov et al., 2013b,a). Our method presented here is different from the previous approaches proposed here, as it can be used with any language concept and can disentangle them from the spuriously correlated – but unrelated patterns – that they usually occur with. After a language task space is constructed, it is straightforward to use it for linguistic research by testing explicit linguistic theories against it, making them a means to more tightly connect deep learning techniques with linguistic theory (Baroni, 2022).

Thirdly, after constructing linguistic spaces, we dive into the analysis of their change throughout the LM training process. We here observe the

change of relevant subspaces throughout the training process. Curiously, as the LM forms better, more generalising representations of the different phenomena, the representations become more distributed. This increase in extrinsic dimensionality (i.e. the number of dimensions that are used within the model to represent the data) opposes the assumed low intrinsic dimensionality (i.e. the number of dimensions that the model requires to represent the data, independent of the extrinsic dimension) that the model requires as it becomes more refined. Research on the intrinsic dimensions of language models (Aghajanyan et al., 2021; Cheng et al., 2023) is still young, and their relationship with the extrinsic dimensionality of single concepts is still unknown.

Ultimately, we find that the learning process of LMs is remarkably stable. This insight might explain why there are, to this day, only a few research papers in curriculum learning (CL) for language models (Campos, 2021; Surkov et al., 2022). The training of state-of-the-art language models is among the most computationally expensive endeavours, making it a prime target for optimisation via CL. As we see here, the learning process of LMs is not marked by stark shifts of generalisation patterns but rather by a continuous reinforcement of early patterns. As such, stark shifts in the data distribution, as we would enforce them on the model via CL, are not prone to yield any improvement.

Limitations One of the goals of the framework that we introduce in Chapter 3 and 4 is to detach the constructed linguistic spaces and the corresponding analysis of the learning dynamics of LMs as much as possible from prior assumptions upon the task structure. By predetermining the tasks (or 'anchors') that we use to span the space, we are unnecessarily constraining the expressivity of the resulting space. For a more expressive linguistic space, the anchors that span the space have to be determined through the dynamics of the language model itself. Further, while our approach applies to all types of knowledge domains, it requires *minimal pairs* of phenomena within that domain to fine-tune them selectively. Minimal pairs are primarily used in linguistics and are uncommon in other knowledge domains. Ultimately, there is concern about the narrow distri-

bution of the synthetic BLiMP data that we have been using for fine-tuning and evaluation of the different linguistic tasks.

Chapter 5: Automated curriculum learning for Interpretability

Summary In Chapter 5, we set out to use automated curriculum learning (CL) as an interpretability method for better understanding the connection of data features with the learning behaviour of language models. The idea is to create a curriculum that improves LM learning using automated CL strategies and analyse the curriculum policy. The curriculum policy should give us insights into the learning dynamics of the LM. While the automated CL framework we employed produced reasonable curricula, it turns out that — upon more rigorous investigation — the created curricula produce no learning advantage. At the same time, the non-functional curricula are remarkably deceptive: the curricula closely resemble known policies from the literature, even though they ultimately work for very different reasons. We found that optimising a model using a curriculum in combination with Adam can lead to unintended interactions between the two. These interactions scale the parameter updates applied to the model, equivalent to a temporary scaling of the learning rate γ. Scaling γ, in return, leads to faster learning if hyperparameters are chosen suboptimally, while optimally-tuned hyperparameters with plain Adam lead to the best performances.

Contextualisation within the literature The fact that pure Adam performs the best throughout all of our experiments might hint at why CL in NLP is rarely used: Adam is the most common optimiser in NLP, and it appears to only benefit from curricula with bad hyperparameters. This chapter warrants particular caution for future research: research in curriculum learning using Adam has to be accompanied by a rigorous hyperparameter search to make reliable claims about the success of the curriculum beyond reducing the need for hyperparameter selection.

Chapter 5 produced an insight that might be valuable for the field of NLP. However, unfortunately, it contributes only indirectly to the declared

goal of the thesis. An interesting observation is that none of our attempts were successful at creating a curriculum for the task of language modelling: the automated approach failed. Further, during pilot experiments, other curriculum strategies were equally unsuccessful. This is in line with multiple previous attempts at creating curriculum learning strategies in language models (e.g. Surkov et al., 2022; Campos, 2021). This might relate to our observations from Chapter 4 that the overall learning process of language models is remarkably continuous: different linguistic concepts are acquired gradually without any major shifts or clearly separable stages in generalisation patterns. A continuous learning process does not require major shifts in the distribution of the learning data.

The inefficiency of curricula in language models starkly contrasts their efficiency in humans, where a well-designed curriculum is critical for learning success. This might highlight the difference in learning processes between humans and machines: In psychology and neuroscience, converging evidence suggests that top-down regulation of learning processes through higher-level control functions is crucial for human efficiency in conceptualisation (see, e.g. Blair and Razza, 2007; Efklides, 2008; Sigman et al., 2014; Metcalfe and Kornell, 2005). Top-down control and feedback might give more structure to learning and make curricula useful. In that case, future generations of language models that more closely resemble humans and use more structured learning dynamics might become sensitive to curricula. A glimpse at models that learn generalisations following more human-like patterns has been recently given by Lake and Baroni (2023), who motivate their model architecture by their ability to make more human-like compositional generalisations.

Limitations The analysis of the interactions includes an extensive range of settings, encompassing different training regimes (toy-setting, training from scratch and fine-tuning pre-trained models), different modalities (vision and language) and different types of curricula (automated vs handcrafted)'. Our findings can fully explain the learning advantages attributed to the curriculum in all cases. However, it is important to say that we cannot make claims about the number of potentially affected curriculum

learning strategies. From our investigation, it is clear that increasing the sizes of gradients will cause Adam to increase the sizes of parameter updates. However, it is empirically impossible for us to reassess all different subtypes of curriculum learning methods. Therefore, we ask to be wary and critically reexamine previous methods instead of dismissing them.

Chapter 6: Robustness in prompt-based learning

Summary In Chapter 6, we investigated the properties of learning without parameter updates (i.e. in-context learning). We mainly focus on the consistency and stability of this learning approach. We show how ICL is mildly sensitive to spurious correlations (or generally the properties of the adaptation data), but this sensitivity has a comparably small influence on the learning outcomes. Other factors play a much greater role: previous literature has shown that presumably insignificant changes to the prompting setup can lead to unforeseeable changes in the model output (Lu et al., 2022; Zhao et al., 2021; Mishra et al., 2022; Min et al., 2022). Research that predominantly looks at single factors and tries to gauge their impact has not come to clear solutions so far. We, therefore, introduce a new test for robustness in prompt-based learning, the ICL consistency test and subsequently conduct a comprehensive analysis of the influence of different setups on predictions of ICL models. The ICL consistency test evaluates the consistency of model predictions on the same data points across many different setups. ICL accuracy in our experiments differs up to 40% across setups, and the primary metric κ reveals that none of the tested models performs with high consistency for any minimal setup change (i.e. across any change in factors). However, there is a tendency: larger and instruction-tuned models generally perform more consistently and robustly. Single factors cause different degrees of inconsistencies and are more 'disruptive' to the general prediction setup: The type of natural language instructions that are used strongly influence the predictions the model makes and the effect of other factors.

Contextualisation within the literature The chapter highlights the need for more holistic thinking in the model analysis and interpretability when the model dynamics require it: ICL currently exhibits chaotic learning dynamics (as shown by e.g. Khashabi et al. (2022)). As a consequence, research might find not reliable or contradictory results when investigating ICL and concentrating only on single factors (Lu et al., 2022; Zhao et al., 2021; Mishra et al., 2022; Min et al., 2022). Holistic evaluations, which test the impact of a factor on many setups at the same time, can give more reliable results. The need for more holistic evaluation to get a clearer picture of the capacities as well as the mechanisms in LLMs, with initiatives like i.a. HELM (Liang et al., 2022), GenBench (Hupkes et al., 2023), BIG-bench (Srivastava et al., 2023) or evaluation pipelines like eval-harness (Gao et al., 2021) gaining prominence.

Limitations The main limitations of our experiments and the ICL consistency test are the number of factors that we chose to include in the test. Albeit we consider our choice of factors appropriate, they are in no way exhaustive. Adding more factors would enrich the analysis. Especially in retrospect, adding more factors that dissect different properties of the `Instructions` that so strongly influence prediction outcomes would have been insightful. However, we think that the performance on the ICL consistency test can be a good indicator of the quality of the generalisation that an LLM is making. Our analysis was further hampered by the decision to use the relatively 'hard' ANLI dataset to run our evaluation: smaller models produce very low ICL accuracies for challenging datasets like ANLI across many factors and provide little variance for meaningful analysis.

7.2 Revisiting the research objectives

In the Introduction (Section 1.3), I specified four research objectives for this dissertation. I will briefly repeat these objectives and evaluate how far the objectives have been met by the work presented in the main body chapters.

1. Connect domain knowledge with learning dynamics

Summary: The goal was to develop a framework which relates formal linguistic theory and learning dynamics in language models.

Evaluation: I created a framework that relates linguistic knowledge with the learning dynamics of language models based on ideas from multi-task learning (MTL) in Chapter 3. As discussed in the previous section (7.1), the experimental work using the framework was limited by implementational details in Chapter3. We mostly addressed these issues in Chapter 4 and placed the framework on a stronger empirical base using more rigorous experiments on the alignment of gradient subspaces. The framework maintains strong connections with the literature on MTL and conceptual spaces.

2. Derive 'synthetic linguistic theories' from language models

Summary: Use the framework from goal 1 to generate a 'linguistic task space' that represents the language model's conceptualisation of language.

Evaluation: In Chapter 4, I created high-dimensional linguistic task spaces using performance transfers of language models across a large range of linguistic concepts. Additionally, I connected these transfers to parameter subspaces within the models and generated similarity spaces based on the alignment of gradients for different tasks within them. Similarity spaces show how language models parse language; which concepts do they share structure across? Which ones can not be reconciled with each other? A major weakness that still has to be addressed in the necessary top-down definition of 'anchors' that we use to span the space: We utilise human-defined phenomena and relate them to each other. However, a more accurate linguistic space can probably be described by 'anchors' that are defined through the model itself and span the conceptual space with maximal expressivity.

3. Investigate generalisation throughout the learning process

Summary: Use different techniques to investigate the change of model generalisation throughout the training process.

Evaluation: I investigated linguistic generalisation throughout the train-

ing process of language models in Chapters 4 and 5. The results of Chapter 4 show that linguistic concept building in LMs happens at different paces for different concepts. However, the process appears relatively continuous and not marked by strongly delimitated 'stages' of learning. This starkly contrasts with human learning, as discussed in the previous section (7.1), and might be one of the reasons for the ineffectiveness of curricula in language modelling, as we observed in Chapter 5. A less marked learning trajectory highlights an essential difference between human and machine learning.

4. Investigate failed generalisation
Summary: Investigate how different data properties lead to inconsistent and non-robust model predictions in learning without parameter updates.
Evaluation: I contributed to a better understanding of inconsistencies and non-robustness in the young learning paradigm of learning without parameter updates by narrowing down the source of inconsistencies in the input data and eliminating potential other sources. However, this is just a first step to understanding what determines ICL predictions: an in-depth analysis of volatile factors will help to understand ICL intricacies better. Showcasing the effectiveness of more holistic methods and rigorous statistics will hopefully convince others of the usefulness of our methodology.

7.3 Contributions

How do my findings translate to concrete, tangible contributions to the fields of linguistics, cognitive sciences, NLP and beyond? In the following, I will broadly cluster the contributions into three subgroups: contribution of our framework for generalisation behaviour in LMs (see **Linguistic spaces** below), our findings regarding the learning process of LMs (see **Learning process of LMs**) as well as contributions on a methodological level (see **Holistic methods**).

Linguistic spaces In the work presented in this dissertation, I create a framework to generate a linguistic conceptual space from a language model's learning dynamics. These linguistic spaces can be understood as a 'synthetic linguistic theory', which can become theoretically interesting for empirically inclined linguistic researchers to quickly test new or contested latent constructs against a language model. A test of a hypothesis can be done by selecting 'anchors' that contain the latent construct under discussion and comparing their generalisation with an alternative subset randomly selected from the remaining 'anchors'. Especially with refined 'anchors' as discussed in Section 7.2 and using LLMs instead of regular LMs, the resulting linguistic spaces can become an exciting tool for linguistic research.

Beyond linguistics, the suggested framework can generally be utilised for any conceptual knowledge a machine learning model learns, and we have data that isolates specific concepts we are interested in. With language models becoming increasingly potent, constructing conceptual spaces can be a helpful tool to interface their latent abilities and overcome the introspection problem of humans and LMs, as I described in Objective 2 in Section 1.3. Constructing conceptual spaces from language models can ultimately help us to create *'synthetic science'* in which we analyse the generalisation behaviour of LLMs to reconstruct their conceptualisations.

Learning process of LMs One of the main ideas behind constructing linguistic task spaces was to see how a language model's linguistic conceptualisation changes throughout its training process. A desideratum was to find major shifts in the conceptualisation at different stages of training and, in addition to that, have clear indicators of how the model's language understanding changes. However, we uncovered that the linguistic conceptualisation of LMs is instead continuous and does not entail significant sudden shifts in its generalisation behaviour. This can be seen as a major contribution to the understanding of the learning dynamics of LMs and is interesting from multiple perspectives: Firstly, it highlights an important difference between human and machine learning dynamics (as I discussed in Section 7.1). Secondly, it gives us an intuition as to why the data pro-

vided to an LM is essential, but the ordering is not and why we do not have any viable curriculum learning strategies for language models up to this day.

Holistic methods In this dissertation, I discuss the contrast between reductionist and holistic methods (mostly Section 1.2.2). I suggest that for analysing complex systems such as language models, certain portions of their dynamics might be hard to address with conventional, strongly reductionist approaches. In many cases, holistic methods are hard to realise and potentially computationally expensive, as they do not allow breaking the problem into manageable portions. However, they can give clearer or complementary insights into otherwise inaccessible research domains. For example, previous to our research in chapter 6, multiple studies looked at different single aspects of ICL setups in isolation (as discussed in Section 7.1). The results were difficult to reconcile with one another, while our holistically oriented approach was evident in its implications. One of the contributions of this dissertation is to showcase a variety of holistic methods, encourage more holistic evaluations where necessary and advise caution with strongly reduced setups with complex subjects. We can learn from experiences in psychological research. In psychology, researchers handle an inherently complex subject: the human mind. Using strongly reductionist methodology on complex subjects contributes to the ongoing replication crisis in the field Ioannidis (2005); Collaboration (2015); Camerer et al. (2018). Awareness of when a methodology is appropriate and applying it rigorously can contribute significantly to the productivity of the field.

7.4 Outlook

I sketch out in the Introduction how language modelling is at the time of writing at a culmination point. Each chapter in this dissertation uses a different model architecture (from LSTMs to Transformers to pre-trained Transformers to large pre-trained transformers) and different training

regimes (from training from scratch to fine-tuning to in-context learning). This rate of change exemplifies how we move at an unprecedented speed, and it is hard to foresee where the current developments will lead us. My outlook will give a concentrated view of the potential future of the research work presented in this dissertation.

The importance of AI and its interpretability Language modelling technology has recently found broad adaptation in the public. The increased utility of language models for all kinds of applications in the recent past promises that large language models and related technology will, in the near future, impact our day-to-day lives. In such a scenario, the understanding of such models is only increasing in importance. Interpretability, however, is multi-faceted, and we can only get a complete picture of a model by using a multitude of methods (Carvalho et al., 2019). Research in interpretability should, therefore, draw on all resources that can help us understand a model, including the learning dynamics and its generalisation patterns. With the increasing complexity of models, better and more creative approaches to interpretability will make an important difference in minimising risks and biases in models, enabling us to benefit from their capacities fairly and safely.

Models of (language) cognition and 'synthetic science' Machine learning models have achieved impressive performances in many application domains in recent years. With LMs in particular, we have, for the first time, a machine-learning model that excels in many environments at the same time. While LMs process their inputs profoundly differently from humans (Fodor and Pylyshyn, 1988; Lake et al., 2017; Lake and Baroni, 2023), they can serve as a model of (non-human) cognition. As such, they can provide us with new ways of conceptualising already profoundly studied domains, potentially blazing the trail to new perspectives. However, extracting conceptual knowledge is challenging as LMs are similarly unable to introspect their inner processes as humans. Improved methods of extracting concepts from LMs can open a rich field of 'synthetic science', helping us to understand our topic of interest from the perspective

of an alternative cognitive system. If constructing conceptual spaces from learning dynamics is the best way to achieve this, it remains to be seen.

To further improve the framework presented in Chapters 3 and 4, a logical next step would be to find different 'anchors' to span the linguistic space. Currently, the data points that span the space are defined by formal linguistic theory. However, anchors directly derived from the generalisation dynamics of the model might be more expressive of the model's language understanding. A space anchored by a more diverse set of data – allowing for more unforeseen interactions between data points – could be more expressive of the models underlying language conceptualisation.

Holistic methods of analysis When extrapolating the development of LLMs, we can easily see how their complexity will continue to increase as scaling laws promise more gains and new approaches start to integrate additional modules such as tools (Schick et al., 2023), perceptual modalities (Bubeck et al., 2023) or scenarios like embodiment become a reality (Driess et al., 2023). As I argued throughout the dissertation, with the increasing complexity of the analysed system, we are more likely to require more holistic methods of analysis. The economic and scientific prospects of more sophisticated LMs are great, and the corresponding shearing force will propel us towards more complex models. Adequate methods to understand or improve them and make them fair and safe will be necessary. Just like for certain aspects of human cognition, for an extensive range of analytic needs in LMs, adequate methods can only be holistic.

Bibliography

Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C. C., Soatto, S., and Perona, P. (2019). Task2vec: Task embedding for meta-learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6429–6438. IEEE.

Adams, R. A. and Fournier, J. J. (2003). *Sobolev spaces*. Elsevier.

Aghajanyan, A., Gupta, S., and Zettlemoyer, L. (2021). Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. (2022). What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*.

Albergante, L., Bac, J., and Zinovyev, A. (2019). Estimating the effective dimension of large biological datasets using fisher separability analysis. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Allgower, E. L. and Georg, K. (1980). *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media.

Aloni, M. and Dekker, P. (2016). *The Cambridge handbook of formal semantics*. Cambridge University Press.

Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853.

Argyriou, A., Micchelli, C. A., Pontil, M., and Ying, Y. (2007). A spectral regularization framework for multi-task structure learning. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 25–32. Curran Associates, Inc.

Awadalla, A., Wortsman, M., Ilharco, G., Min, S., Magnusson, I., Hajishirzi, H., and Schmidt, L. (2022). Exploring the landscape of distributional robustness for question answering models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5971–5987, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bach, S., Sanh, V., Yong, Z. X., Webson, A., Raffel, C., Nayak, N. V., Sharma, A., Kim, T., Bari, M. S., Fevry, T., Alyafeai, Z., Dey, M., Santilli, A., Sun, Z., Ben-david, S., Xu, C., Chhablani, G., Wang, H., Fries, J., Al-shaibani, M., Sharma, S., Thakker, U., Almubarak, K., Tang, X., Radev, D., Jiang, M. T.-j., and Rush, A. (2022). PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Baddeley, A. D. (2003). Working memory and language: an overview. *Journal of communication disorders*, 36(3):189–208.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun,

Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., et al. (2023). A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *ArXiv preprint*, abs/2302.04023.

Banse, R. and Scherer, K. R. (1996). Acoustic profiles in vocal emotion expression. *Journal of personality and social psychology*, 70(3):614.

Barker, C. (2018). Negative polarity as scope marking. *Linguistics and Philosophy*, pages 1–28.

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

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Barrett, H. C., Cosmides, L., and Tooby, J. (2007). The hominid entry into the cognitive niche. *Evolution of mind, fundamental questions and controversies*, pages 241–248.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and brain sciences*, 22(4):577–660.

Barsalou, L. W. (2008). Grounded cognition. *Annu. Rev. Psychol.*, 59:617–645.

Bartolo, M., Roberts, A., Welbl, J., Riedel, S., and Stenetorp, P. (2020). Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Baxter, J. (2000). A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198.

Ben-David, S. and Borbely, R. S. (2008). A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine learning*, 73:273–287.

Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 932–938. MIT Press.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In Danyluk, A. P., Bottou, L., and Littman, M. L., editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Benton, A., Mitchell, M., and Hovy, D. (2017). Multitask learning for mental health conditions with limited social media data. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 152–162, Valencia, Spain. Association for Computational Linguistics.

Bickel, P. J., Li, B., Tsybakov, A. B., van de Geer, S. A., Yu, B., Valdés, T., Rivero, C., Fan, J., and van der Vaart, A. (2006). Regularization in statistics. *Test*, 15:271–344.

Binder, J. R. and Desai, R. H. (2011). The neurobiology of semantic memory. *Trends in cognitive sciences*, 15(11):527–536.

Bingel, J. and Søgaard, A. (2017). Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain. Association for Computational Linguistics.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Blair, C. and Razza, R. P. (2007). Relating effortful control, executive function, and false belief understanding to emerging math and literacy ability in kindergarten. *Child development*, 78(2):647–663.

Blumenfeld, H. K. and Marian, V. (2011). Bilingualism influences inhibitory control in auditory comprehension. *Cognition*, 118(2):245–257.

Bolukbasi, T., Chang, K., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4349–4357.

Boroditsky, L. (2001a). Does language shape thought?: Mandarin and english speakers' conceptions of time. *Cognitive psychology*, 43(1):1–22.

Boroditsky, L. (2001b). Does language shape thought?: Mandarin and english speakers' conceptions of time. *Cognitive psychology*, 43(1):1–22.

Boulenger, V., Roy, A. C., Paulignan, Y., Deprez, V., Jeannerod, M., and Nazir, T. A. (2006). Cross-talk between language processes and overt

motor behavior in the first 200 msec of processing. *Journal of cognitive neuroscience*, 18(10):1607–1615.

Boyd, R., Richerson, P. J., and Henrich, J. (2011). The cultural niche: Why social learning is essential for human adaptation. *Proceedings of the National Academy of Sciences*, 108(supplement_2):10918–10925.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Lai, J. C., and Mercer, R. L. (1992). An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv preprint*, abs/2303.12712.

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Camerer, C. F., Dreber, A., Holzmeister, F., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Nave, G., Nosek, B. A., Pfeiffer, T., et al. (2018). Evaluating the replicability of social science experiments in nature and science between 2010 and 2015. *Nature human behaviour*, 2(9):637–644.

Cameron, L. and Larsen-Freeman, D. (2007). Complex systems and applied linguistics. *International journal of applied linguistics*, 17(2):226–240.

Campadelli, P., Casiraghi, E., Ceruti, C., and Rozza, A. (2015). Intrinsic dimension estimation: Relevant techniques and a benchmark framework. *Mathematical Problems in Engineering*, 2015:1–21.

Campos, D. (2021). Curriculum learning for language modeling. *ArXiv preprint*, abs/2108.02170.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann.

Caruana, R. (1997). Multitask Learning. *Machine Learning*, 28(1):41–75.

Carvalho, D. V., Pereira, E. M., and Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832.

Casasanto, D. (2011). Different bodies, different minds: the body specificity of language and thought. *Current Directions in Psychological Science*, 20(6):378–383.

Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., et al. (2023). Open problems and fundamental limitations of reinforcement learning from human feedback. *ArXiv preprint*, abs/2307.15217.

Caucheteux, C. and King, J.-R. (2022). Brains and algorithms partially converge in natural language processing. *Communications biology*, 5(1):134.

Chelba, C. and Jelinek, F. (1998). Expoiting syntactic structure for language modeling. *arXiv preprint cs/9811022*.

Cheng, E., Kervadec, C., and Baroni, M. (2023). Bridging information-theoretic and geometric compression in language models.

Chomsky, N. (1957). *Syntactic Structures*. De Gruyter Mouton, Berlin, Boston.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. The MIT Press, 50 edition.

Chowdhury, S. A. and Zamparelli, R. (2019). An LSTM adaptation study of (un)grammaticality. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 204–212, Florence, Italy. Association for Computational Linguistics.

Clyne, M. G. (2003). *Dynamics of language contact: English and immigrant languages*. Cambridge University Press.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Collaboration, O. S. (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. In Cohen, W. W., McCallum, A., and Roweis, S. T., editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM.

Collobert, R., Weston, J., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Cormier, S. M. and Hagman, J. D. (2014). *Transfer of learning: Contemporary research and applications*. Academic Press.

Csáji, B. C. et al. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24(48):7.

Dai, A. M. and Le, Q. V. (2015). Semi-supervised sequence learning. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3079–3087.

De Bot, K., Lowie, W., and Verspoor, M. (2007). A dynamic systems theory approach to second language acquisition. *Bilingualism: Language and cognition*, 10(1):7–21.

De Lacy, P. (2007). *The Cambridge handbook of phonology*. Cambridge University Press.

Deacon, T. W. (1997). *The symbolic species: The co-evolution of language and the brain*. Number 202. WW Norton & Company.

Delétang, G., Ruoss, A., Duquenne, P.-A., Catt, E., Genewein, T., Mattern, C., Grau-Moya, J., Wenliang, L. K., Aitchison, M., Orseau, L., et al. (2023). Language modeling is compression. *ArXiv preprint*, abs/2309.10668.

den Dikken, M. (2013). *The Cambridge handbook of generative syntax*. Cambridge University Press.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. (2022). Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *ArXiv preprint*, abs/2208.07339.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter*

of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dolan, W. B. and Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 647–655. JMLR.org.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. (2023). A survey for in-context learning. *ArXiv preprint*, abs/2301.00234.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *ArXiv preprint*, abs/1702.08608.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al. (2023). Palm-e: An embodied multimodal language model. *ArXiv preprint*, abs/2303.03378.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. (2019). Challenges of real-world reinforcement learning. *ArXiv preprint*, abs/1904.12901.

Efklides, A. (2008). Metacognition: Defining its facets and levels of functioning in relation to self-regulation and co-regulation. *European psychologist*, 13(4):277–287.

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.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

European Parliament and Council of the European Union (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council.

Everitt, B. (1998). The cambridge dictionary of statistics. In *The Cambridge dictionary of statistics*. Cambridge University Press.

Eysenbach, B., Kumar, A., and Gupta, A. (2020). Reinforcement learning is supervised learning on optimized data.

Fan, F.-L., Xiong, J., Li, M., and Wang, G. (2021). On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(6):741–760.

Fan, Y., Tian, F., Qin, T., Li, X., and Liu, T. (2018). Learning to teach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Fauconnier, G. (1975). Polarity and the scale principle. *Chicago Linguistics Society*, 11:188–199.

Fedorenko, E. and Thompson-Schill, S. L. (2014). Reworking the language network. *Trends in cognitive sciences*, 18(3):120–126.

Feigenbaum, M. J. (1980). Universal behavior in nonlinear systems. *Universality in chaos*, pages 49–84.

Fodor, J. A. (1975). *The Language of Thought*. Harvard University Press.

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

Futrell, R., Wilcox, E., Morita, T., Qian, P., Ballesteros, M., and Levy, R. (2019a). Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Futrell, R., Wilcox, E., Morita, T., Qian, P., Ballesteros, M., and Levy, R. (2019b). Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Gailly, J.-l. and Adler, M. (1992). Gnu gzip. *GNU Operating System*.

Gallese, V. and Lakoff, G. (2005). The brain's concepts: The role of the sensory-motor system in conceptual knowledge. *Cognitive neuropsychology*, 22(3-4):455–479.

Gao, L., Tow, J., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., McDonell, K., Muennighoff, N., Phang, J., Reynolds, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. (2021). A framework for few-shot language model evaluation.

Gardenfors, P. (2004). *Conceptual spaces: The geometry of thought*. MIT press.

Gardenfors, P. (2014). *The geometry of meaning: Semantics based on conceptual spaces*. MIT press.

Gardner, M., Artzi, Y., Basmov, V., Berant, J., Bogin, B., Chen, S., Dasigi, P., Dua, D., Elazar, Y., Gottumukkala, A., Gupta, N., Hajishirzi, H., Ilharco, G., Khashabi, D., Lin, K., Liu, J., Liu, N. F., Mulcaire, P., Ning, Q., Singh, S., Smith, N. A., Subramanian, S., Tsarfaty, R., Wallace, E., Zhang, A., and Zhou, B. (2020). Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. (2022). What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58.

Geva, M., Goldberg, Y., and Berant, J. (2019). Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Ghorbani, B., Firat, O., Freitag, M., Bapna, A., Krikun, M., Garcia, X., Chelba, C., and Cherry, C. (2022). Scaling laws for neural machine translation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Giannakidou, A. (2011). Negative and positive polarity items: Variation, licensing, and compositionality. In *Semantics: An International Handbook of Natural Language Meaning*, pages 1660–1712. Walter de Gruyter.

Gilkerson, J., Richards, J. A., Warren, S. F., Montgomery, J. K., Greenwood, C. R., Oller, D. K., Hansen, J. H. L., and Paul, T. D. (2017). Mapping the early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

Glaese, A., McAleese, N., Trebacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., et al. (2022). Improving alignment of dialogue agents via targeted human judgements. *ArXiv preprint*, abs/2209.14375.

Goldberg, Y. (2019). Assessing bert's syntactic abilities. *ArXiv preprint*, abs/1901.05287.

Goodman, J. T. (2001). A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434.

Gopnik, A., Glymour, C., Sobel, D. M., Schulz, L. E., Kushnir, T., and Danks, D. (2004). A theory of causal learning in children: causal maps and bayes nets. *Psychological review*, 111(1):3.

Gopnik, A., Meltzoff, A. N., and Kuhl, P. K. (1999). *The scientist in the crib: Minds, brains, and how children learn.* William Morrow & Co.

Grammarly, I. (2023). Grammarly: Free writing ai assistance. Accessed: -.

Grassberger, P. and Procaccia, I. (1983). Measuring the strangeness of strange attractors. *Physica D: nonlinear phenomena*, 9(1-2):189–208.

Green, D. W. (1998). Mental control of the bilingual lexico-semantic system. *Bilingualism: Language and cognition*, 1(2):67–81.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Gutzmann, D. (2020). *The Wiley Blackwell companion to semantics*. Wiley-Blackwell.

Hacohen, G. and Weinshall, D. (2019). On the power of curriculum learning in training deep networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2535–2544. PMLR.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE intelligent systems*, 24(2):8–12.

Haro, G., Randall, G., and Sapiro, G. (2008). Translated poisson mixture model for stratification learning. *International Journal of Computer Vision*, 80:358–374.

Hart, B. and Risley, T. R. (1995). *Meaningful differences in the everyday experience of young American children*. Paul H Brookes Publishing.

Hauk, O., Johnsrude, I., and Pulvermüller, F. (2004). Somatotopic representation of action words in human motor and premotor cortex. *Neuron*, 41(2):301–307.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Hendrycks, D., Lee, K., and Mazeika, M. (2019). Using pre-training can improve model robustness and uncertainty. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2712–2721. PMLR.

Hendrycks, D., Liu, X., Wallace, E., Dziedzic, A., Krishnan, R., and Song, D. (2020). Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Hensley, J. (2010). *A brief introduction and overview of complex systems in applied linguistics*. Oxford University Press.

Hernandez, D., Kaplan, J., Henighan, T., and McCandlish, S. (2021). Scaling laws for transfer. *ArXiv preprint*, abs/2102.01293.

Hippisley, A. and Stump, G. (2016). *The Cambridge handbook of morphology*. Cambridge University Press.

Hoeksema, J. (2012). On the Natural History of Negative Polarity Items Syntax View project Morphology View project. *Linguistic Analysis*, 44(2):3–3–3.

Hoff, E. (2003). The specificity of environmental influence: Socioeconomic status affects early vocabulary development via maternal speech. *Child development*, 74(5):1368–1378.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al.

(2022). Training compute-optimal large language models. *ArXiv preprint*, abs/2203.15556.

Holland, J. H. (2000). *Emergence: From chaos to order*. OUP Oxford.

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 498–520.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Hu, J., Gauthier, J., Qian, P., Wilcox, E., and Levy, R. (2020). A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Hubinger, E., van Merwijk, C., Mikulik, V., Skalse, J., and Garrabrant, S. (2019). Risks from learned optimization in advanced machine learning systems. *ArXiv preprint*, abs/1906.01820.

Hupkes, D., Giulianelli, M., Dankers, V., et al. (2023). A taxonomy and review of generalization research in nlp. *Nature Machine Intelligence*, 5:1161–1174.

Huttenlocher, J., Waterfall, H., Vasilyeva, M., Vevea, J., and Hedges, L. V. (2010). Sources of variability in children's language growth. *Cognitive psychology*, 61(4):343–365.

Ioannidis, J. P. (2005). Why most published research findings are false. *PLoS medicine*, 2(8):e124.

Jackendoff, R. (2002). *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press.

James, G. M. (2003). Variance and bias for general loss functions. *Machine learning*, 51:115–135.

Jia, R. and Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Jiang, L., Zhou, Z., Leung, T., Li, L., and Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2309–2318. PMLR.

Jiang, Z., Yang, M., Tsirlin, M., Tang, R., Dai, Y., and Lin, J. (2023). "low-resource" text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6810–6828.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Jumelet, J., Denic, M., Szymanik, J., Hupkes, D., and Steinert-Threlkeld, S. (2021). Language models use monotonicity to assess NPI licensing.

In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4958–4969, Online. Association for Computational Linguistics.

Jumelet, J. and Hupkes, D. (2018). Do language models understand anything? on the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Jumelet, J. and Hupkes, D. (2019). diagnnose: A neural net analysis library.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall series in artificial intelligence. Pearson Prentice Hall.

Kaiser, L., Gomez, A. N., Shazeer, N., Vaswani, A., Parmar, N., Jones, L., and Uszkoreit, J. (2017). One model to learn them all. *ArXiv preprint*, abs/1706.05137.

Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *ArXiv preprint*, abs/2001.08361.

Kaushik, D., Hovy, E. H., and Lipton, Z. C. (2020). Learning the difference that makes A difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Kavumba, P., Inoue, N., Heinzerling, B., Singh, K., Reisert, P., and Inui, K. (2019). When choosing plausible alternatives, clever hans can be clever. In *Proceedings of the First Workshop on Commonsense Inference*

in Natural Language Processing, pages 33–42, Hong Kong, China. Association for Computational Linguistics.

Kavumba, P., Takahashi, R., and Oda, Y. (2022). Are prompt-based models clueless? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2333–2352, Dublin, Ireland. Association for Computational Linguistics.

Kelly, S. D., Özyürek, A., and Maris, E. (2010). Two sides of the same coin: Speech and gesture mutually interact to enhance comprehension. *Psychological science*, 21(2):260–267.

Kervadec, C., Antipov, G., Baccouche, M., and Wolf, C. (2021). Roses are red, violets are blue... but should VQA expect them to? In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2776–2785. Computer Vision Foundation / IEEE.

Khashabi, D., Lyu, X., Min, S., Qin, L., Richardson, K., Welleck, S., Hajishirzi, H., Khot, T., Sabharwal, A., Singh, S., and Choi, Y. (2022). Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3631–3643, Seattle, United States. Association for Computational Linguistics.

Kim, T.-H. and Choi, J. (2018). Screenernet: Learning self-paced curriculum for deep neural networks. *ArXiv preprint*, abs/1801.00904.

Kim, Y., Denton, C., Hoang, L., and Rush, A. M. (2017). Structured attention networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference*

on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Kocmi, T. and Bojar, O. (2017). Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 379–386, Varna, Bulgaria. INCOMA Ltd.

Kousta, S.-T., Vinson, D. P., and Vigliocco, G. (2009). Emotion words, regardless of polarity, have a processing advantage over neutral words. *Cognition*, 112(3):473–481.

Kriegeskorte, N. and Douglas, P. K. (2018). Cognitive computational neuroscience. *Nature neuroscience*, 21(9):1148–1160.

Kriegeskorte, N. and Kievit, R. A. (2013). Representational geometry: integrating cognition, computation, and the brain. *Trends in cognitive sciences*, 17(8):401–412.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Bartlett, P. L., Pereira, F. C. N., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114.

Kroll, J. F. and Bialystok, E. (2013). Understanding the consequences of bilingualism for language processing and cognition. *Journal of cognitive psychology*, 25(5):497–514.

Krueger, K. A. and Dayan, P. (2009). Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394.

Kumar, M. P., Packer, B., and Koller, D. (2010). Self-paced learning for latent variable models. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 1189–1197. Curran Associates, Inc.

Ladusaw, W. A. (1980). *Polarity Sensitivity as Inherent Scope Relations*. PhD thesis, University of Texas, Austin.

Lake, B. M. and Baroni, M. (2023). Human-like systematic generalization through a meta-learning neural network. *Nature*, pages 1–7.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Lakoff, G. and Johnson, M. (1980). *Metaphors We Live By*. University of Chicago Press.

Lee, S., Goldt, S., and Saxe, A. M. (2021). Continual learning in the teacher-student setup: Impact of task similarity. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6109–6119. PMLR.

Levina, E. and Bickel, P. J. (2004). Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 777–784.

Levinson, S. C. (1983). *Pragmatics*. Cambridge university press.

Lewis, D. (1969). *Convention: A philosophical study*. John Wiley & Sons.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lewis, P., Stenetorp, P., and Riedel, S. (2021). Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Li, J., Li, D., Savarese, S., and Hoi, S. (2023a). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv preprint*, abs/2301.12597.

Li, J., Li, D., Xiong, C., and Hoi, S. C. H. (2022). BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S., editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.

Li, Y., Ildiz, M. E., Papailiopoulos, D., and Oymak, S. (2023b). Transformers as algorithms: Generalization and implicit model selection in in-context learning. *ArXiv preprint*, abs/2301.07067.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., et al. (2022). Holistic evaluation of language models. *ArXiv preprint*, abs/2211.09110.

Lindquist, K. A. and Gendron, M. (2013). What's in a word? language constructs emotion perception. *Emotion Review*, 5(1):66–71.

Linzen, T., Dupoux, E., and Goldberg, Y. (2016). Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Liu, C., He, S., Liu, K., and Zhao, J. (2018). Curriculum learning for natural answer generation. In Lang, J., editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4223–4229. ijcai.org.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019a). Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Liu, X., He, P., Chen, W., and Gao, J. (2019b). Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019c). Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Lu, S., Bigoulaeva, I., Sachdeva, R., Madabushi, H. T., and Gurevych, I. (2023). Are emergent abilities in large language models just in-context learning? *ArXiv preprint*, abs/2309.01809.

Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. (2022). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Luong, M., Le, Q. V., Sutskever, I., Vinyals, O., and Kaiser, L. (2016). Multi-task sequence to sequence learning. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Luriia, A. R. (1976). *Cognitive development: Its cultural and social foundations*. Harvard university press.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

MacWhinney, B. (2001). Emergentist approaches to language. *Typological studies in language*, 45:449–470.

Madiega, T. A. (2021). Artificial intelligence act. *European Parliament: European Parliamentary Research Service*.

Majid, A. (2012). Current emotion research in the language sciences. *Emotion Review*, 4(4):432–443.

Majid, A. and Levinson, S. C. (2011). The senses in language and culture. *The Senses and Society*, 6(1):5–18.

Marblestone, A. H., Wayne, G., and Kording, K. P. (2016). Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, 10:94.

Markov, A. A. (2006). An example of statistical investigation of the text eugene onegin concerning the connection of samples in chains. *Science in Context*, 19(4):591–600.

Marsland, S. (2011). *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC.

Martin, A. (2007). The representation of object concepts in the brain. *Annu. Rev. Psychol.*, 58:25–45.

Martínez Alonso, H., Agić, Ž., Plank, B., and Søgaard, A. (2017). Parsing Universal Dependencies without training. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 230–240, Valencia, Spain. Association for Computational Linguistics.

Marvin, R. and Linzen, T. (2018). Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Maurer, A. (2006). Bounds for linear multi-task learning. *Journal of Machine Learning Research*, 7(Jan):117–139.

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467.

McCoy, T., Pavlick, E., and Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

McNeill, D. (1992). *Hand and mind: What gestures reveal about thought*. University of Chicago press.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2017). Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Metcalfe, J. and Kornell, N. (2005). A region of proximal learning model of study time allocation. *Journal of memory and language*, 52(4):463–477.

Meteyard, L., Cuadrado, S. R., Bahrami, B., and Vigliocco, G. (2012). Coming of age: A review of embodiment and the neuroscience of semantics. *Cortex*, 48(7):788–804.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Team, G. B., Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., et al. (2011). Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.

Mikolov, T. (2012). *Statistical Language Models Based on Neural Networks*. Ph.d. thesis, Brno University of Technology, Faculty of Information Technology.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. *Interspeech 2010*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Mill, J. S. (1856). *A System of Logic, Ratiocinative and Inductive: 1*, volume 1. Parker.

Miller, J., Krauth, K., Recht, B., and Schmidt, L. (2020). The effect of natural distribution shift on question answering models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6905–6916. PMLR.

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. (2022). Rethinking the role of demonstrations:

What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mishra, S., Khashabi, D., Baral, C., Choi, Y., and Hajishirzi, H. (2022). Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Mosbach, M., Pimentel, T., Ravfogel, S., Klakow, D., and Elazar, Y. (2023). Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12284–12314, Toronto, Canada. Association for Computational Linguistics.

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P. (2020). Curriculum learning for reinforcement learning domains: A framework and survey. *J. Mach. Learn. Res.*, 21:181:1–181:50.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2020). Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Niedenthal, P. M. (2007). Embodying emotion. *Science*, 316(5827):1002–1005.

Nisbett, R. E. and Wilson, T. D. (1977). Telling more than we can know: Verbal reports on mental processes. *Psychological review*, 84(3):231.

Niven, T. and Kao, H.-Y. (2019). Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Noshad, M., Zeng, Y., and III, A. O. H. (2019). Scalable mutual information estimation using dependence graphs. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 2962–2966. IEEE.

O'Grady, W. (2008). The emergentist program. *Lingua*, 118(4):447–464.

Ohmer, X., Bruni, E., and Hupkes, D. (2023). Evaluating task understanding through multilingual consistency: A chatgpt case study. *ArXiv preprint*, abs/2305.11662.

Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A. (2018). The building blocks of interpretability. *Distill*, 3(3):e10.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. (2022). In-context learning and induction heads. *ArXiv preprint*, abs/2209.11895.

OpenAI (2023). Chatgpt by openai. Accessed: *.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Özyürek, A. (2014). Hearing and seeing meaning in speech and gesture: Insights from brain and behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1651):20130296.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. (2018). Image transformer. In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4052–4061. PMLR.

Parrish, A., Chen, A., Nangia, N., Padmakumar, V., Phang, J., Thompson, J., Htut, P. M., and Bowman, S. (2022). BBQ: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105, Dublin, Ireland. Association for Computational Linguistics.

Passos, A., Rai, P., Wainer, J., and III, H. D. (2012). Flexible modeling of latent task structures in multitask learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

Penha, G. and Hauff, C. (2020). Curriculum learning strategies for ir. In *European Conference on Information Retrieval*, pages 699–713. Springer.

Pérez-Mayos, L., Carlini, R., Ballesteros, M., and Wanner, L. (2021). On the evolution of syntactic information encoded by BERT's contextualized representations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2243–2258, Online. Association for Computational Linguistics.

Perkins, D. N., Salomon, G., et al. (1992). Transfer of learning. *International encyclopedia of education*, 2:6452–6457.

Peters, M. E., Neumann, M., Zettlemoyer, L., and Yih, W.-t. (2018). Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium. Association for Computational Linguistics.

Piaget, J., Cook, M., et al. (1952). *The origins of intelligence in children*, volume 8. International Universities Press New York.

Platanios, E. A., Stretcu, O., Neubig, G., Poczos, B., and Mitchell, T. (2019). Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Prasad, G., van Schijndel, M., and Linzen, T. (2019). Using priming to uncover the organization of syntactic representations in neural language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Pulvermüller, F. (2005). Brain mechanisms linking language and action. *Nature reviews neuroscience*, 6(7):576–582.

Pulvermüller, F., Hauk, O., Nikulin, V. V., and Ilmoniemi, R. J. (2005). Functional links between motor and language systems. *European Journal of Neuroscience*, 21(3):793–797.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

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

Raghu, A., Raghu, M., Kornblith, S., Duvenaud, D., and Hinton, G. E. (2021). Teaching with commentaries. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ramachandran, P., Liu, P., and Le, Q. (2017). Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark. Association for Computational Linguistics.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Adler, T., Kreil, D. P., Kopp, M. K., Klambauer, G., Brandstetter, J., and Hochreiter, S. (2021). Hopfield networks is all you need. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Razeghi, Y., Logan IV, R. L., Gardner, M., and Singh, S. (2022). Impact of pretraining term frequencies on few-shot reasoning. *ArXiv preprint*, abs/2202.07206.

Regier, T. and Kay, P. (2009). Language, thought, and color: Whorf was half right. *Trends in cognitive sciences*, 13(10):439–446.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In Krishnapuram, B., Shah, M., Smola, A. J., Aggarwal, C. C., Shen, D., and Rastogi, R., editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM.

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471.

Rohde, D. L. and Plaut, D. C. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, 72(1):67–109.

Rosenblatt, F. et al. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*, volume 55. Spartan books Washington, DC.

Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278.

Rosenstein, M. T., Marx, Z., Kaelbling, L. P., and Dietterich, T. G. (2005). To transfer or not to transfer. In *In NIPS'05 Workshop, Inductive Transfer: 10 Years Later*.

Rowe, M. L. (2012). A longitudinal investigation of the role of quantity and quality of child-directed speech in vocabulary development. *Child development*, 83(5):1762–1774.

Ruder, S. (2017). An Overview of Multi-Task Learning in Deep Neural Networks. *ArXiv preprint*, abs/1706.05098.

Rudin, W. (1953). *Principles of mathematical analysis*. McGraw Hill.

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

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Rust, P., Lotz, J. F., Bugliarello, E., Salesky, E., de Lhoneux, M., and Elliott, D. (2022). Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*.

Salakhutdinov, R. (2014). Deep learning. In Macskassy, S. A., Perlich, C., Leskovec, J., Wang, W., and Ghani, R., editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, page 1973. ACM.

Sapir, E. (1929). The status of linguistics as a science. *Language*, pages 207–214.

Sapolsky, R. and Balt, S. (1996). Reductionism and variability in data: a meta-analysis. *Perspectives in biology and medicine*, 39(2):193–203.

Savitzky, A. and Golay, M. J. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8):1627–1639.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. (2018). On the information bottleneck theory of deep learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Scherer, K. R. (2003). Vocal communication of emotion: A review of research paradigms. *Speech communication*, 40(1-2):227–256.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *ArXiv preprint*, abs/2302.04761.

Schmidhuber, J. (1990). Towards compositional learning in dynamic networks. *Technical University of Munich (Technical Report FKI-129-90)*.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. M. (2012). On causal and anticausal learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress.

Scholte, H. S. (2017). Fantastic dnimals and where to find them. *Neuroimage*, 180(Pt A):112–113.

Schwartz, D. L., Bransford, J. D., Sears, D., et al. (2005). Efficiency and innovation in transfer. *Transfer of learning from a modern multidisciplinary perspective*, 3:1–51.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. (2020). The pitfalls of simplicity bias in neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Shwartz-Ziv, R. and Tishby, N. (2017). Opening the black box of deep neural networks via information. *ArXiv preprint*, abs/1703.00810.

Si, C., Gan, Z., Yang, Z., Wang, S., Wang, J., Boyd-Graber, J., and Wang, L. (2023). Prompting gpt-3 to be reliable.

Sigman, M., Peña, M., Goldin, A. P., and Ribeiro, S. (2014). Neuroscience and education: prime time to build the bridge. *Nature neuroscience*, 17(4):497–502.

Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American philosophical society*, 106(6):467–482.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Søgaard, A. and Goldberg, Y. (2016). Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany. Association for Computational Linguistics.

Solomonoff, R. J. (1964). A formal theory of inductive inference. part i. *Information and control*, 7(1):1–22.

Spelke, E. S. and Tsivkin, S. (2001). Language and number: A bilingual training study. *Cognition*, 78(1):45–88.

Sperber, D. (1996). *Explaining culture: A naturalistic approach*. Oxford Blackwell.

Sperber, D. and Wilson, D. (1986). *Relevance: Communication and cognition*, volume 142. Citeseer.

Spitkovsky, V. I., Alshawi, H., and Jurafsky, D. (2010a). From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of*

the North American Chapter of the Association for Computational Linguistics, pages 751–759, Los Angeles, California. Association for Computational Linguistics.

Spitkovsky, V. I., Alshawi, H., and Jurafsky, D. (2010b). From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In Kaplan, R., Burstein, J., Harper, M., and Penn, G., editors, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 751–759, Los Angeles, California. Association for Computational Linguistics.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv preprint*, abs/2206.04615.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2023). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Standley, T., Zamir, A. R., Chen, D., Guibas, L. J., Malik, J., and Savarese, S. (2020a). Which tasks should be learned together in multi-task learning? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9120–9132. PMLR.

Standley, T., Zamir, A. R., Chen, D., Guibas, L. J., Malik, J., and Savarese, S. (2020b). Which tasks should be learned together in multi-task learning? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9120–9132. PMLR.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020). Learning to

summarize with human feedback. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Surkov, M., Mosin, V., and Yamshchikov, I. (2022). Do data-based curricula work? In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 119–128, Dublin, Ireland. Association for Computational Linguistics.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Stanford alpaca: An instruction-following llama model.

Tay, Y., Wang, S., Luu, A. T., Fu, J., Phan, M. C., Yuan, X., Rao, J., Hui, S. C., and Zhang, A. (2019). Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931, Florence, Italy. Association for Computational Linguistics.

Teney, D., Lin, Y., Oh, S. J., and Abbasnejad, E. (2023). Id and ood performance are sometimes inversely correlated on real-world datasets.

Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Thierry, G., Athanasopoulos, P., Wiggett, A., Dering, B., and Kuipers, J.-R. (2009). Unconscious effects of language-specific terminology on preattentive color perception. *Proceedings of the National Academy of Sciences*, 106(11):4567–4570.

Thrun, S. and O'Sullivan, J. (1996). Discovering structure in multiple learning tasks: The tc algorithm. In *Proceedings of the Thirteenth*

International Conference on Machine Learning, pages 489–497. Morgan Kaufmann.

Tishby, N., Pereira, F. C., and Bialek, W. (1999). The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*.

Tishby, N. and Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE.

Titchener, E. B. (1912). The schema of introspection. *The American Journal of Psychology*, 23(4):485–508.

Tomasello, M. (2009). *The cultural origins of human cognition*. Harvard University Press.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Trudgill, P. (2019). *Sociolinguistic variation and change*. Edinburgh University Press.

Tu, L., Lalwani, G., Gella, S., and He, H. (2020). An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Tucker, M. and Hirsh-Pasek, K. (1993). Systems and language: implications for acquisition.

Vapnik, V. (1982). Estimation of dependences based on empirical data: Springer series in statistics (springer series in statistics).

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999.

Varela, F. J., Thompson, E., and Rosch, E. (1991). *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. (2023). Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Wang, A., Hula, J., Xia, P., Pappagari, R., McCoy, R. T., Patel, R., Kim, N., Tenney, I., Huang, Y., Yu, K., Jin, S., Chen, B., Van Durme, B., Grave, E., Pavlick, E., and Bowman, S. R. (2019a). Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4465–4476, Florence, Italy. Association for Computational Linguistics.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019b). Superglue: A stickier benchmark for general-purpose language understanding systems. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019c). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on*

Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Wang, H., Zhang, Y., Yu, X., et al. (2020). An overview of image caption generation methods. *Computational intelligence and neuroscience*, 2020.

Wang, X., Girshick, R. B., Gupta, A., and He, K. (2018). Non-local neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7794–7803. IEEE Computer Society.

Wang, X., Wang, H., and Yang, D. (2022). Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. (2023). Self-instruct: Aligning language models with self-generated instructions. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Wang, Z., Dai, Z., Póczos, B., and Carbonell, J. G. (2019d). Characterizing and avoiding negative transfer. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11293–11302. Computer Vision Foundation / IEEE.

Wang, Z., Hamza, W., and Florian, R. (2017). Bilateral multi-perspective matching for natural language sentences. In Sierra, C., editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4144–4150. ijcai.org.

Warstadt, A. and Bowman, S. R. (2020). Can neural networks acquire a structural bias from raw linguistic data? *ArXiv preprint*, abs/2007.06761.

Warstadt, A., Cao, Y., Grosu, I., Peng, W., Blix, H., Nie, Y., Alsop, A., Bordia, S., Liu, H., Parrish, A., Wang, S.-F., Phang, J., Mohananey, A., Htut, P. M., Jeretic, P., and Bowman, S. R. (2019). Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., and Bowman, S. R. (2020). BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Watson, J. B. (1913). Psychology as the behaviorist views it. *Psychological review*, 20(2):158.

Weaver, W. (1952). Translation. In *Proceedings of the Conference on Mechanical Translation*, Massachusetts Institute of Technology.

Weber, L., Bruni, E., and Hupkes, D. (2023). The icl consistency test. In *Proceedings of GenBench: The first workshop on generalisation (benchmarking) in NLP*, Online.

Weber, L., Jumelet, J., Bruni, E., and Hupkes, D. (2021). Language modelling as a multi-task problem. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2049–2060, Online. Association for Computational Linguistics.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022a). Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning*

Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022b). Emergent abilities of large language models. *ArXiv preprint*, abs/2206.07682.

Wei, J., Wei, J., Tay, Y., Tran, D., Webson, A., Lu, Y., Chen, X., Liu, H., Huang, D., Zhou, D., et al. (2023). Larger language models do in-context learning differently. *ArXiv preprint*, abs/2303.03846.

Weisleder, A. and Fernald, A. (2013). Talking to children matters: Early language experience strengthens processing and builds vocabulary. *Psychological science*, 24(11):2143–2152.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Whorf, B. L. (1940). *Science and linguistics*. Bobbs-Merrill Indianapolis, IN, USA:.

Wilcox, E., Qian, P., Futrell, R., Ballesteros, M., and Levy, R. (2019). Structural supervision improves learning of non-local grammatical dependencies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3302–3312, Minneapolis, Minnesota. Association for Computational Linguistics.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wittgenstein, L. (1953). *Philosophical Investigations*. Basil Blackwell, Oxford.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint*, abs/1910.03771.

Xu, B., Zhang, L., Mao, Z., Wang, Q., Xie, H., and Zhang, Y. (2020). Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Yamins, D. L. and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365.

Yoo, K. M., Kim, J., Kim, H. J., Cho, H., Jo, H., Lee, S.-W., Lee, S.-g., and Kim, T. (2022). Ground-truth labels matter: A deeper look into input-label demonstrations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2437, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. (2020). Gradient surgery for multi-task learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zamir, A. R., Sax, A., Shen, W. B., Guibas, L. J., Malik, J., and Savarese, S. (2019a). Taskonomy: Disentangling task transfer learning. In Kraus, S., editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 6241–6245. ijcai.org.

Zamir, A. R., Sax, A., Shen, W. B., Guibas, L. J., Malik, J., and Savarese, S. (2019b). Taskonomy: Disentangling task transfer learning. In Kraus, S., editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 6241–6245. ijcai.org.

Zhang, Y., Baldridge, J., and He, L. (2019). PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhang, Y. and Yang, Q. (2017). A Survey on Multi-Task Learning. *ArXiv preprint*, abs/1707.08114.

Zhang, Z., Luo, P., Loy, C. C., and Tang, X. (2014). Facial landmark detection by deep multi-task learning. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 94–108, Cham. Springer International Publishing.

Zhao, Z., Wallace, E., Feng, S., Klein, D., and Singh, S. (2021). Calibrate before use: Improving few-shot performance of language models. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Zhong, R., Lee, K., Zhang, Z., and Klein, D. (2021). Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics:*

EMNLP 2021, pages 2856–2878, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., and Levy, O. (2023). Lima: Less is more for alignment.

Zwaan, R. A. (2004). The immersed experiencer: Toward an embodied theory of language comprehension. *Psychology of learning and motivation*, 44:35–62.

Appendices

A.1 Chapter 1

The supplementary material to Chapter 3 contains additional information on the NPIs used in our experiments.

A.1.1 List of NPIs

We here present the full list of 160 NPIs that has been used for modifying the corpora:

- a bed of roses
- a care in the world
- a chance in hell
- a damn
- a damn thing
- a day goes by
- a day over
- a ghost of a
- a hair out of place
- a living soul
- a moment of your time
- a moment too soon
- a shadow of a doubt
- a single soul
- all that much
- all that many
- any
- any longer
- any old
- any time soon
- anybody
- anymore
- anyone
- anything
- anything like
- anytime soon
- anywhere
- anywhere close
- anywhere near
- as of yet
- as yet
- at all
- avail
- bat an eye
- be any time
- be anything like

- beat around the bush
- by a long sho
- by any chance
- by any means
- by any stretch
- by miles
- by much
- can be bothered
- can compare to
- can hold a candle to
- can make of
- can possibly
- chance in hell
- come at a worse time
- come cheap
- could care less
- could possibly
- cut the mustard
- even once
- ever
- far wrong
- for much longer
- for shit
- for the life of
- for the soul of
- give a crap
- give a damn
- give a fuck
- give a shit
- half a chance
- half bad
- have a clue
- have any of
- hold a candle to
- hold water
- in a blue moon
- in a hundred years
- in a long time
- in a million years
- in ages
- in all of history
- in any
- in any manner
- in any way
- in centuries
- in days
- in decades
- in his right mind
- in hours
- in living memory
- in minutes
- in months
- in recent memory
- in the least
- in the least bit
- in the slightest
- in weeks
- in years
- just any
- just yet
- know the first thing
- know the first thing about
- know the half of it
- least of all
- let alone
- lift a finger
- make a sound
- make head or tail of
- make much difference
- mean a thing
- mean feat
- miss a beat
- much care
- much help
- much of a
- much of anything
- much to look at
- much to lose
- nor
- on speaking terms
- on your life
- one single thing
- or anything
- rhyme or reason

- say much
- see eye to eye
- set foot
- set foot in
- set foot on
- sit right with
- sit well
- sit well with
- small feat
- so much as
- square with
- squat
- stand a chance
- strong suit
- such thing
- sweat it
- take his eyes off
- take kindly to
- take lightly
- take no for an answer
- that many
- that much
- that often
- the ghost of
- the half of
- the half of it
- the least bit
- the like of which
- the likes of which
- the slightest
- the slightest bit
- think much of
- to be taken lightly
- whatever
- whatsoever
- with a barge pole
- worth a damn
- worth his salt
- worth its salt
- yet

A.2 Chapter 2

The supplementary material to Chapter 4 contains additional information about the control hypothesis spaces that we employ to verify the meaningfulness of our linguistic spaces (Appendix A.2.1). Further, we document the development of the LMs' performance on BLiMP in different scenarios (Appendix A.2.2). We also provide additional information on the development of subspace sizes throughout training (Appendix A.2.3). Ultimately, we show all heatmaps for all transfer and gradient spaces for all models throughout the whole training process (Appendix A.2.4).

A.2.1 Controls

We include control conditions and baselines for our experiments. This appendix section provides the necessary details.

A.2.1.1 Vocabulary baselines

We calculate two baselines to estimate the amount of transfer that is due to mere vocabulary overlap between different paradigms: (1) *Absolute token overlap* between the vocabularies V_A and V_B of different paradigms – calculated simply as the size of their intersection $|V_A \cap V_B|$ – and (2) the *Wasserstein distance* $\mathcal{W}$ (Kantorovich, 1960) between the vocabularies distributions. These vocabulary controls can be correlated with correlated with the transfer or gradient spaces. The degree of correlation indicates how much of the transfer between different paradigms can be attributed to the vocabulary overlap between phenomena alone.

A.2.2 BLiMP performance

Throughout our experiments, we pre-train and fine-tune our LMs. We here document the performance of the models in different scenarios: first, we show how the models perform on the whole benchmark throughout the pre-training process. Second, we show how different pre-training checkpoints

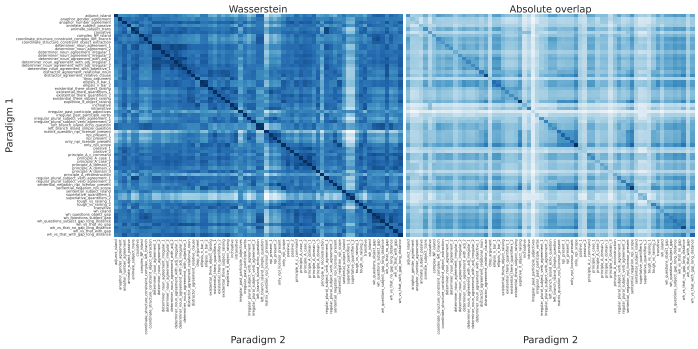


Figure A.1: (1 - $\mathcal{W}$) in the left heatmap and the normalised absolute vocabulary overlap on the right.

adapt In the following, we detail the performance of our models on the BLiMP dataset throughout the pre-training process.

A.2.2.1 BLiMP learning curves

During the pre-training process, we evaluate each saved checkpoint on all paradigms of the BLiMP benchmark and average the results. The following plot shows the respective learning curves for the different models. While none of the models achieve very good performance, the largest model achieves their final performance much faster than the smaller ones.

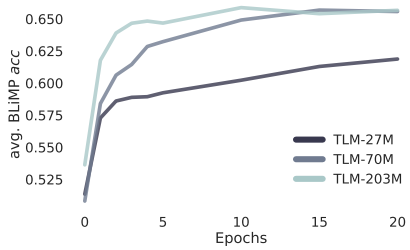


Figure A.2: Learning curves achieved on the different paradigms of the BLiMP dataset by our generative transformer LM.

A.2.2.2 BLiMP probe tuning

The final performance after fine-tuning a specific phenomenon changes with the amount of pre-training. The final performance of that model for that specific phenomenon is shown in Figure A.3. With more pre-training, models adapt better during the fine-tuning. Larger models generally adapt better than smaller models.

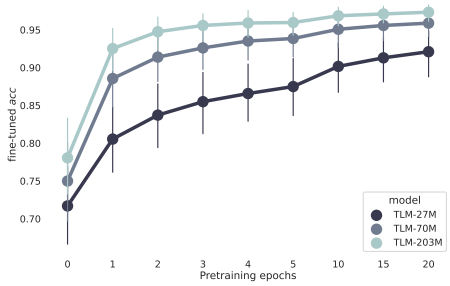


Figure A.3: Average final performance after fine-tuning a linguistic task.

A.2.3 Subspace sizes

Throughout the training process, the average size of the subspaces with which the LMs learn the different linguistic phenomena increases. The subspace sizes within the larger models increase to a higher percentage of their overall parameters and continue to increase for longer. Interestingly, we have seen in the main body of the thesis (see Section 4.4.2) that this increase does not happen at random but rather is directed to increase the overlap between related phenomena throughout the training process.

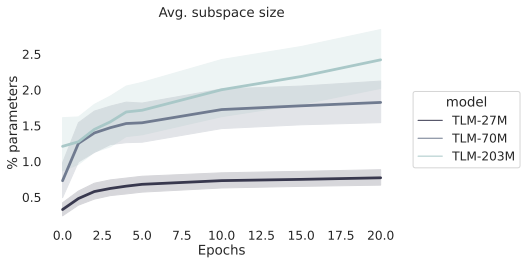
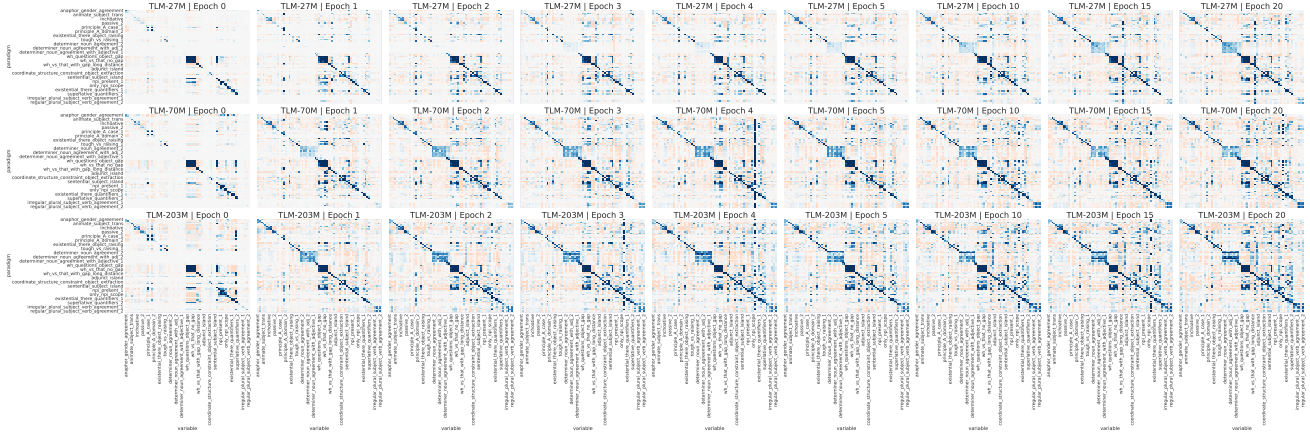


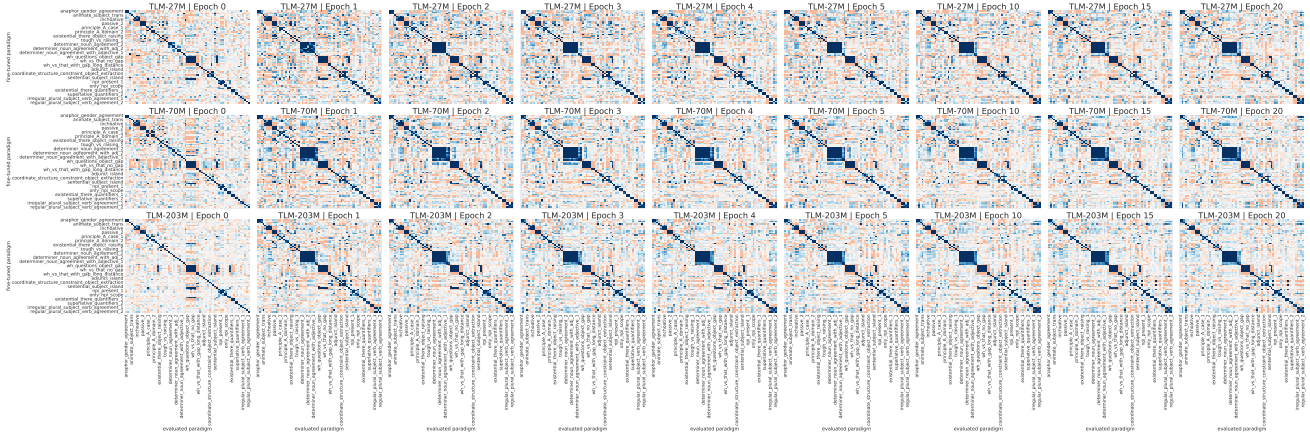
Figure A.4: Average subspace sizes of different linguistic phenomena throughout the training process.

A.2.4 Similarity spaces

We calculate all similarity spaces throughout the training process. Figure A.5 on the following page illustrates the transfer and gradient matrices for each saved model checkpoint.



(a) Transfer spaces for all models throughout training.



(b) Gradient spaces for all models throughout training.

Figure A.5

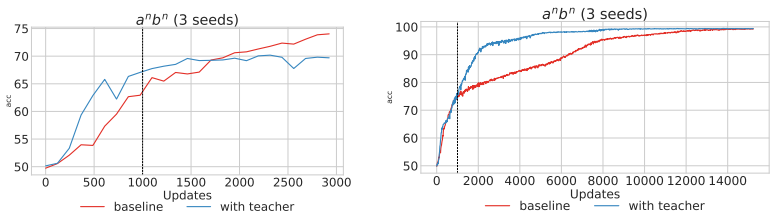
A.3 Chapter 3

The supplementary material to Chapter 5 contains additional information about exact hyperparameter settings for all experiments (Appendix A.3.2), additional learning curves for all replications of Raghu et al. (2021)'s experiments (Appendix A.3.3) and results for our extension to GLUE-data (Appendix A.3.5). Further, we illustrate the weighting policies of the toy-teachers from Section 5.3.2 (Appendix A.3.4), give empirical proof for relation of difficulty measures with their associated gradient norms $|g|$ (Appendix A.3.6) and provide the learning-curves of our finetuning experiments using hand-crafted curricula in Section 5.4.2 which are summarised in Table 5.3. Ultimately, we disclose the hardware infrastructure that we used to conduct all experiments (Appendix A.3.8).

A.3.1 Additional Experiment: Impact of batch-size differences

We found that differences in hyperparameter settings can have a large impact on teaching outcomes. For computational limitations, we use practice students with small batch sizes during teacher optimisation as this allows for more steps I_p in the inner loop. Depending on our choice of batch sizes during the following longer training, the effectiveness of the teacher differs substantially: Batch sizes of similar size to the ones used by practice students produce performance improvements for is that the teacher has seen during training (up to I_p; see Figure A.6 (a)).

On the other hand, with an increased batch size during longer training compared to teacher optimisation, the performance improvements project towards later stages of training (beyond I_p; see Figure A.6 (b)). Curious is that at these later stages of training in which we observe the learning speed improvements, variance in teacher weights is already close to 0 (see Figure 5.4 (a)) and the teacher therefore has no influence on the student training anymore.



(a) (b)

Figure A.6: (a) Student trained with batch size of 8 and a teacher optimised with a practice student trained with batch-size of 8; (b) Student trained with batch size of 64 with the same teacher

A.3.2 Hyperparameter details

Throughout our experiments, we employed different sets of hyperparameters. In the following tables, we summarise the hyperparameter settings for every experiment, separated by hyperparameters for training and fine-tuning, for model architectures (if not given by Raghu et al., 2021) and for the schedule functions of our hand-crafted curricula:

A.3.2.1 Hyperparameters training

Here, 'variable' values are set depending on the specific subset of GLUE we train on. RoBERTa-models that were trained with hand-crafted (HC) curricula were trained using suboptimal (LOW) and optimal (OPT) learning rates.

Table A.1: Hyperparameters training.

EXPERIMENT		γ (LR)	LR-DECAY	BATCH SIZE	WARM-UP	EPOCHS	$I_{practice}$	$I_{teacher}$
§ 5.3.2: COMMENTARIES	CIFAR (T)	INNER: 10^{-4} ; OUTER: 10^{-3}	NONE	8	-	-	1500	100
	GLUE (T)	INNER: 10^{-4}; OUTER: 10^{-3}	NONE	8	-	-	VARIABLE	100
	CIFAR (S)	2L-CNN: 10^{-3}; RESNET: 10^{-5}	NONE	64	-	25	-	-
	GLUE (S)	ROBERTA: 4×10^{-6}	SQUARE-ROOT	8	100	VARIABLE	-	-
§ 5.4.1: HC CURRICULA	ALL	LOW: 4×10^{-6}; OPT.: 2×10^{-5}	SQUARE-ROOT	8	100	VARIABLE	-	-

A.3.2.2 Hyperparameters model architecture

For all replications of Raghu et al. (2021)'s experiments, we used their exact same model architectures. To transfer commentaries to NLP, we conducted a small hyperparameter search to find the smallest possible model architecture for the practice student S_{p} and teacher (T) model that maintains the capacity to substantially reduce the empirical error on all GLUE-benchmark-tasks. The best model follows the transformer-encoder architecture and is implemented using the fairseq library (Vaswani et al., 2017; Ott et al., 2019). S_{p} and T are using the same base architecture.

Table A.2: Hyperparameters models.

EXPERIMENT		N-LAYERS	EMB-DIMS	FFN-EMB	ATTENTION-HEADS
§ 5.3.2: COMMENTARIES	GLUE (T AND S_{p})	2	64	64	8

A.3.2.3 Hyperparameters schedule functions

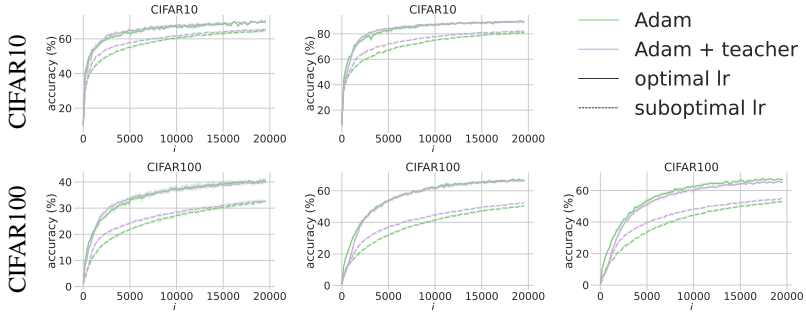
We obtain the exact shape of the manual schedule functions from Section 5.4.2 through a hyperparameter grid search and selected the triples in Table A.3 as best performing schedule functions for our two hand-crafted curricula. 'Start portion' describes percentage of initially used data, 'step size' how much data is added to the portion of used data at every increment and 'increment' is the number of updates after which additional data is added to the pool of used data.

Table A.3: Hyperparameters schedule functions.

EXPERIMENT		START PORTION	STEP SIZE	INCREMENT
§ 5.4.2: HAND-CRAFTED CURRICULA	SEQUENCE LENGTH CURRICULUM	30%	10%	300
	LOSS CURRICULUM	30%	10%	50

A.3.3 Replication commentaries curriculum CIFAR10/100

The following (Figure A.7) shows the performance of different models on CIFAR10 and CIFAR100 when trained with and without the commentaries curriculum. We replicate Raghu et al. (2021)'s results. However, we also find that their hyperparameter setting is suboptimal and that with properly tuned hyperparameters, vanilla Adam outperforms the curriculum.



(a) 2-layer CNN (b) ResNet18 (c) ResNet34

Figure A.7: All replication results from the original paper, with suboptimal hyperparameters that show the effect from the original paper and optimised hyperparameters.

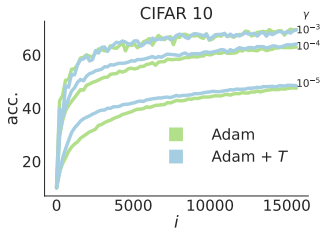
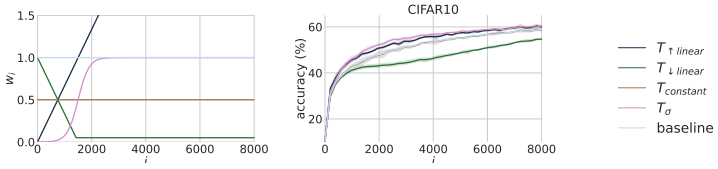


Figure A.8: Learning curves for the 2-layer CNN trained on CIFAR10 with and without teacher at different learning rates γ. We see how lowering γ helps commentaries improve over the vanilla Adam. At the overall best γ, however, vanilla Adam performs on par.

A.3.4 Toy curricula CIFAR10

In this section, we exemplify the simple loss-weighting policies that we described in Section 5.3.2. When applied to the 2-layer CNN model while training on the CIFAR10 dataset, the toy-teacher show how a simple shift of loss-reweighting from low- to high weight values can improve learning speed above no weighting (baseline with $w_i = 1$). We can also see, how decreasing weights have the opposite effect (see $T_{\downarrow \text{linear}}$) and that the absolute value of the weight has no influence (compare T_{constant} and baseline).



(a) (b)

Figure A.9: The left side (a) shows the weights applied to the loss by the different toy curricula. The right side (b) shows the performance of a 2-layer CNN trained on CIFAR10 with the different toy curricula.

A.3.5 GLUE with Commentaries

In this section, we document the learning speed improvements that we observe with commentaries when we finetune RoBERTa on different GLUE-tasks. Either axis shows the steps that the models requires to converge to 98% of its final performance when it is trained with and without a commentaries teacher. We can see how with a suboptimal learning-rate (lr), RoBERTa generally converges faster when it is trained with commentaries (dots land above the diagonal). As soon as we use the optimal learning rate, Adam without a teacher converges faster or just as fast as with teacher (crosses land below the diagonal or on it).

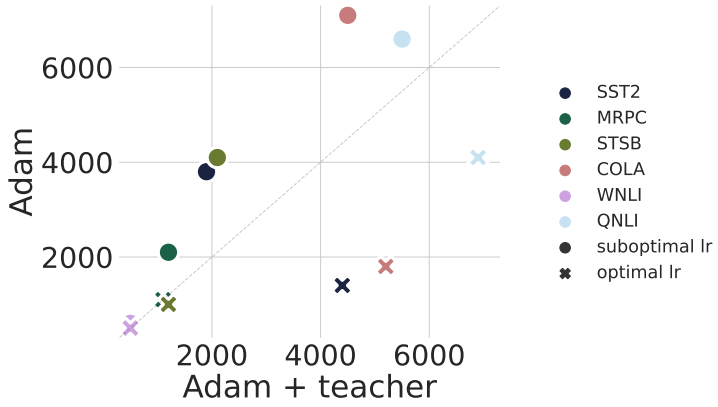
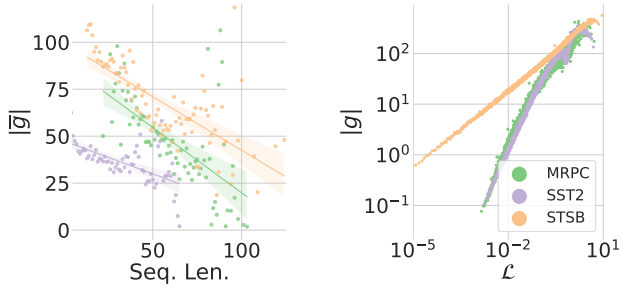


Figure A.10: Updates RoBERTa$_{\text{BASE}}$ needs to converge when finetuned on different GLUE tasks, with and without teacher. Dots above the line mean that the model with teacher learns faster; dots below the line mean the model without teacher is faster. We see how an optimal learning rate eliminates the effects of the teacher. Convergence is defined as 98% of final validation performance.

A.3.6 Correlations of difficulty measures with $|g|$

We stated in Section 5.4.2 that difficulty measures are correlated with the size of the gradient that they evoke in a model. We here show empirically that this is the case for the two difficulty measures that we are considering in our experiments (sequence length and loss).



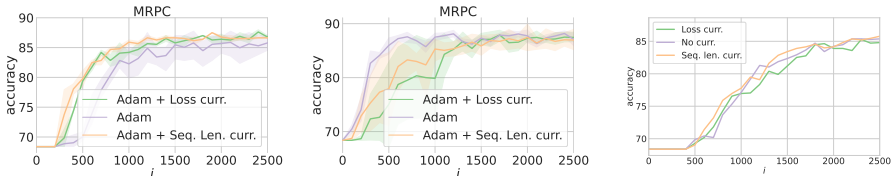
(a) (b)

Figure A.11: Covariance of common difficulty measures (Sequence length and Loss) with the size of gradients that they produce when fine-tuning $\text{RoBERTa}_{\text{BASE}}$ for a selection of GLUE-tasks. Both, sequence lengths (a) and by cross-entropy-loss (b) are highly correlated with the average gradient norms. We chose a representative subset of GLUE and binned data points to improve the presentability of the results.

A.3.7 Learning curves hand-crafted curricula

In this section, we present the learning-curves that correspond to the training-runs summarised in Table 5.3. We can see that our hand-crafted curricula only provide an advantage when γ is set low. As soon as we use an optimal learning rate, plain Adam outperforms the curricula. Moreover, learning with the curricula becomes highly unstable (see by variance across runs), something that is generally known to happen when parameter updates are too large. Ultimately, we can also see how the benefit in hand-

crafted curricula can also be eliminated by setting beta-values to equal values, just like we previously observed it for commentaries before.



(a) default βs + low γ (b) default βs + high γ (c) $\beta_1 = \beta_2 = 0.99$

Figure A.12: Learning curves of $\text{RoBERTa}_{\text{BASE}}$ when finetuned on MRPC trained with the hand-crafted curricula. (a) shows the performance when Adam's β-parameters allow for interaction. The learning rate γ = 4e-6 lets our hand-crafted curricula outperform the baseline using vanilla Adam. (b) Shows what happens with optimal γ = 2e-5: vanilla Adam outperforms any curriculum condition. (c) shows the performance when interactions are prevented. Here, the curricula do not yield any learning advantage.

A.3.8 Computational resources

In this very last section, we disclose the computational infrastructure that was necessary to conduct our experiments. As commentaries require to save the whole computational graph of the practice student's training to be saved, GPUs with larger vRAM are desirable.

Table A.4: Computational resources used for conducting our experiments.

Resources	Type	Quantity	Capacity
GPUs	NVIDIA A30	5	24GB HBM2
CPUs	Intel Xeon Silver	25	2.4GHz x 10
RAM	–	1	256GB

A.4 Chapter 4

The supplementary material to Chapter 6 contains additional information about the fine-tuning of LMs in the first section of the experiments. Further, more details about the datasets and instruction templates used in all experiments are given. For experiment 2, we add detailed descriptions of `factors` as well as our motivations to include them. We provide results to supplementary analyses in the second set of experiments.

A.4.1 List of finetuned models

		Models	
		RoBERTa$_{\text{BASE}}$	RoBERTa$_{\text{LARGE}}$
Base datasets	MNLI	textattack/roberta-base-MNLI	roberta-large-mnli
	SQuAD	deepset/roberta-base-squad2	deepset/roberta-large-squad2
	QQP	own	own
Adv. datasets	HANS	own	own
	ANLI	own	own
	PAWS	own	own
	SQuAD adversarial	own	own
	adversarial QA	own	own
	SQuAD shifts	own	own

A.4.2 Finetuning details of own models

We finetuned all RoBERTa models using the same set of hyperparameters based on the literature and experience.

Hyperparameters We train using the ADAM Optimizer with γ = 1e-05, inverse square root decay and $\beta_{1/2}$ = (0.9, 0.999), no weight decay, 250 warmup steps and a batch size of 8. We stop training if the model does not show improvement on the validation set for 1 epoch of training.

Data For adversarially tuned models, we mixed the training set of the base data with 70% of the adversarial data (30% retained for evaluation). We ensured a mixing ratio of 20%/80% adversarial/base data.

A.4.3 Experiment 1: Datasets details

We here provide additional information about the datasets we use in Experiment 1:

A.4.3.1 Base datasets

MNLI (Multi Natural Language inference; Williams et al. 2018)
A large-scale natural language inference dataset. It contains sentence pairs annotated with three categories: entailment, contradiction, and neutral. The dataset is sourced from a variety of genres, like fiction, government documents, and telephone conversations, thus encouraging models to learn domain-agnostic representations.

QQP (Quora Question Pairs; Wang et al. 2017)
A collection of question pairs from the Quora platform, labelled as either duplicates or non-duplicates. The aim is to identify semantically equivalent questions, addressing challenges such as paraphrasing and varying levels of detail.

SQuAD (Stanford Question Answering Dataset; Rajpurkar et al. 2016)
A reading comprehension dataset consisting of questions about passages from Wikipedia. The questions are human-annotated, and the answer to each question is a segment (or span) of the passage. The goal of models is to identify and extract the correct span from the passage that answers the question.

A.4.3.2 Adversarial datasets

HANS (Heuristic Analysis for NLI Systems; McCoy et al. 2019)
Constructed to evaluate models on non-entailment cases that appear entailed due to spurious biases. Built upon common NLI datasets like SNLI and MultiNLI, it dissects three heuristic strategies that a model might utilise: lexical overlap, subsequence, and syntactic structure.

ANLI (Adversarial Natural Language Inference; Nie et al. 2020)

Generated by first training models on existing datasets (e.g., SNLI and MultiNLI) and then having human annotators produce examples that the models predict incorrectly. Generation of additional examples was done in multiple rounds with respectively improved models, accordingly each round increases the adversarial difficulty.

PAWS (Paraphrase Adversaries from Word Scrambling; Zhang et al. 2019)

Comprises sentence pairs with high lexical overlap but differing semantics, challenging models that heavily weigh word overlap. An adversarial expansion to datasets like the Quora Question Pairs dataset (QQP).

SQuAD Adversarial (Jia and Liang, 2017)

A derivative of the Stanford Question Answering Dataset (SQuAD) where adversarial sentences are introduced into the context paragraphs, aiming to mislead models into selecting incorrect answers while the correct answers remain unchanged.

Adversarial QA (Bartolo et al., 2020)

A reading comprehension dataset, where each question is tied to a Wikipedia passage. Distinctively, answer annotations are freeform human responses rather than extracts from the passage, testing the extractive capability boundaries of SQuAD-inspired models.

SQuAD Shifts (Miller et al., 2020)

Formed by perturbing the original SQuAD distribution in terms of linguistic and stylistic attributes. This dataset gauges model robustness against unseen data distributions, such as domain shifts or synthetic noise.

A.4.4 Experiment 1: Impact of spurious correlations in ICL

We conducted an additional analysis of the results in Section 6.3.2. The goal of this additional analysis is to understand the impact of the type of adaptation data (adversarial vs. base) on the prediction outcomes in comparison with *other* factors that we varied in our experiments (such as the type of `instruction template`, whether the model was `instruction tuned` or the `size` of the model). `Type data` is a binary factor indicating whether the model was adapted on base or adversarial data; `Size` is a quarternary factor indicating model size; `Type instructions` is a binary factor indicating the type of template that was used; `Instruction tuned` is a binary factor indicating whether the tested model was instruction tuned or not.

Table A.5 shows the summary statistics of an ANOVA that we apply to these factors and their impact on the model accuracy. We can see from Table A.5 that adaptation data is the only factor that does not significantly impact prediction outcomes.

	df	**sum_sq**	**mean_sq**	**F**	**P(>F)**
Type data	1.0	8.67	8.67	0.12	0.72
Size	3.0	6626.73	2208.91	31.26	5.71e-18
Type instruction	1.0	95.32	95.32	1.34	0.024
Instruction tuned	1.0	900.55	900.55	12.74	4.05e-04
Residual	357.0	25220.11	70.64	NaN	NaN

Table A.5: Results of ANOVA

A.4.5 Prompt template examples

A.4.5.1 FLAN instructions

Input:

Does the Hypothesis in the input entail (True) or contradict (False) the Premise or is it independent (Neither)?
Premise: Kirklees Stadium (known as the John Smith's Stadium due to sponsorship), is a multi-use sports stadium in Huddersfield in West Yorkshire, England. Since 1994, it has been the home ground of football club Huddersfield Town and rugby league side Huddersfield Giants, both of whom moved from Leeds Road.
Hypothesis: Kirklees Stadium is in Scotland.

OPTIONS:
- True
- Neither
- False

ANSWER: False.

[...]

Does the Hypothesis in the input entail (True) or contradict (False) the Premise or is it independent (Neither)?
Premise: Jonathan Smith (born January 17, 1971), better known by his stage name Lil Jon, is an American rapper, record producer, and DJ. He was the frontman of the group Lil Jon & The East Side Boyz, which he formed in 1997, and they released several albums until 2004.
Hypothesis: Jonathan Smith spent much of his time in China.

OPTIONS:
- True
- Neither
- False

ANSWER:

Target:

Neither.

A.4.5.2 P3 - details

In the following, we provide more details on the instruction templates (Bach et al., 2022), as used in Experiments II.

P3 details – names Names of all available P3-instructions, ordered as in Figure 6.2

1. 'MNLI Crowdsource'
2. 'Guaranteed Possible Impossible'
3. 'Always Sometimes Never'
4. 'Consider Always Sometimes Never'
5. 'Does This Imply'
6. 'Guaranteed True'
7. 'GPT 3 Style'
8. 'Take the Following as Truth'
9. 'Must Be True'
10. 'Based on the Previous Passage'
11. 'Should Assume'
12. 'Can We Infer'
13. 'Justified in Saying'
14. 'Does It Follow That'
15. 'Claim True False Inconclusive'

P3 details – examples

High-performing templates 'Claim true false inconclusive'

[...]

Jonathan Smith (born January 17, 1971), better known by his stage name Lil Jon, is an American rapper, record producer, and DJ. He was the frontman of the group Lil Jon & The East Side Boyz, which he formed in 1997, and they released several albums until 2004. Based on that information, is the claim: "Jonathan Smith spent much of his time in China." true, false, or inconclusive?

ANSWER:

High-performing templates ‘Does it follow that’

[...]
Given that Jonathan Smith (born January 17, 1971), better known by his stage name Lil Jon, is an American rapper, record producer, and DJ. He was the frontman of the group Lil Jon & The East Side Boyz, which he formed in 1997, and they released several albums until 2004. Does it follow that Jonathan Smith spent much of his time in China. Yes, no, or maybe?

ANSWER:

Low-performing templates ‘MNLI crowdsource’

[...]
Jonathan Smith (born January 17, 1971), better known by his stage name Lil Jon, is an American rapper, record producer, and DJ. He was the frontman of the group Lil Jon & The East Side Boyz, which he formed in 1997, and they released several albums until 2004. Using only the above description and what you know about the world, ”Jonathan Smith spent much of his time in China.” is definitely correct, incorrect, or inconclusive?
ANSWER:

Low-performing templates ‘Guaranteed possible impossible’

[...]
Assume it is true that Jonathan Smith (born January 17, 1971), better known by his stage name Lil Jon, is an American rapper, record producer, and DJ. He was the frontman of the group Lil Jon & The East Side Boyz, which he formed in 1997, and they released several albums until 2004.

Therefore, ”Jonathan Smith spent much of his time in China.” is guaranteed, possible, or impossible?

ANSWER:

A.4.6 Introducing custom factors

The ICL consistency test allows the addition of additional user-defined factors. This is useful if factors should be evaluated that are related to modifications of the model (e.g. whether it was instruction-tuned Wei et al., 2022a, or not) or when the model was evaluated in a different way (e.g. whether we calibrate our output probabilities Zhao et al., 2021, or

not). Note that adding a factor in this way will change the overall results of the analysis (see Section 6.4.1 for more details). Alternatively, the task can be evaluated separately for either of the user-defined factors.

A.4.7 Factors details

In the following, we provide a more detailed description of the factors used in Section 6.4 and also provide our motivation to include these factors.

A.4.7.1 Variance factors

Size We consider models of different sizes. Model size has been shown to be an important moderating factor in probably all previous studies on in-context learning.

Instruction tuning We have seen previously that instruction tuning improves the consistency of a model across templates (see Section 6.4.1). We introduce it as a factor to show which other invariance factors it may affect.

Calibration Previous research has shown how small models are especially biased towards single labels when prompted. We find similar tendencies for our model: We exploratively calculate the entropy of a model's predictions across all data points in a dataset. This allows us to estimate whether a model is biased toward predicting a single label (low entropy). Optimally, a model's prediction should be close to the entropy of the target distribution $\mathcal{H}(Y)$. We find that smaller models have a larger bias towards predicting a single label (lower prediction entropy), while larger and IT models get closer to $\mathcal{H}(Y)$ (see Figure A.13).
Zhao et al. (2021) suggests solving this issue by calibrating the model probabilities using 'content-free' prompts. We add the factor of calibration to assess its effects systematically.

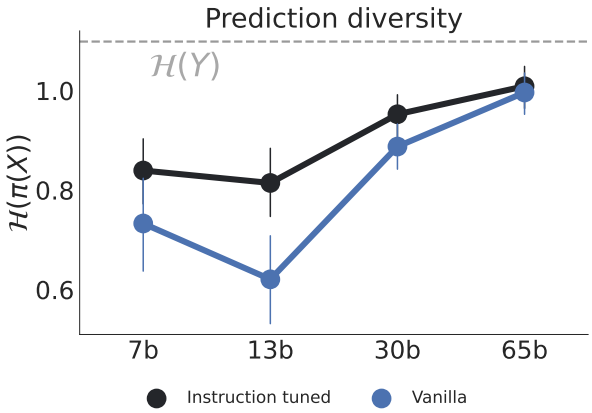


Figure A.13

n-shots The number of in-context examples has been shown to interact with other factors (e.g. according to Zhao et al., 2021, `calibration` has a more significant effect for fewer in-context examples). We would also expect that `n-shots` interacts with many other in-context factors such as `one label`, in which we show the model just examples with the same label in-context, is modulated by the number of in-context examples. We introduce 'few' ($k = 2$) and 'many' ($k = 5$) examples as a factor.

Instruction quality Ultimately, we have seen how some instructions produce consistent and relatively well-performing responses across different models while others do not (see Section 6.4.1. We add this last factor to see which other types of factors help the in-context learner cope with varying `instruction quality`. We chose the two best and two worst-performing templates[1] from our previous analysis.

A.4.7.2 Invariance factors

The following briefly describes each of the tested λ_{inv}.

[1] See Appendix A.4.5 for an example of the instructions

Balanced labels Zhao et al. (2021) additionally showed how a majority label among the in-context example can influence the distribution of model outputs. Therefore, we compare contexts with balanced in-context label distribution with randomly sampled labels and an extreme case with only a single in-context label.

Cross-instruction We include `cross-templates` as a factor to assess model robustness to shifts in label space and surface form of instruction formulation. Previous research has shown how in-context learners are sensitive to the instructions (Mishra et al., 2022) as well as the label distribution $\mathcal{C}$ (Min et al., 2022). The experiments of Min et al. (2022) represent an extreme case in which $\mathcal{C}$ is resampled to be random tokens. While these edge cases are theoretically attractive, we here change this scenario to a practically common one, where instructions and labels are semantically equivalent but have different surface forms by randomly sampling from the available p3 instructions for the in-context examples. We test the impact of in-context instructions in a single setting with results shown in Figure A.14 Surprisingly, almost all models are robust to semantic-invariant changes to instructions of the in-context examples despite changes in the label space and substantial changes in surface form and format across different instructions.

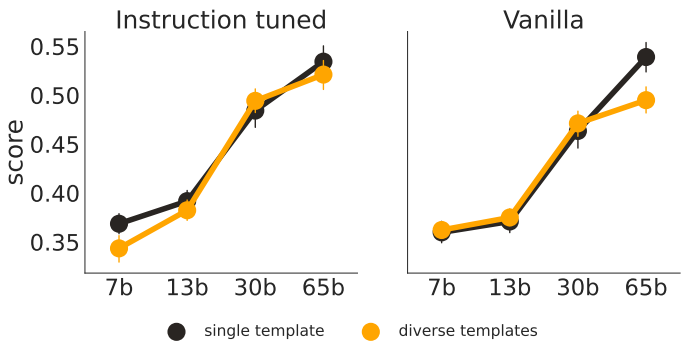


Figure A.14

Cross-task In `cross-task`, we exchange the task of the in-context examples such that the only consistency between in-context and target examples is the general format (x followed by y) and the truthfulness of the x to y mapping. To see whether conditioning on a fixed label space matters, we add tasks with a discriminative (QQP) and a generative (SQuAD) objective as different factors. Compared to a zero-shot baseline, we can see that large models can benefit from conditioning on other tasks (Figure A.15).

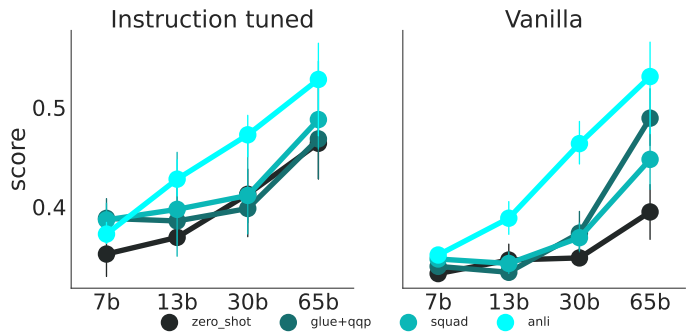


Figure A.15

For our principal analysis, we only include QQP as an in-context task, as SQuAD is incompatible with many other factors (such as `balanced labels`, `one label` aso...)

Instructions Besides the quality of the instructions, we are also interested in how consistent model behaviour is across `instructions` that are of similar quality. To get an insight into this, we bin the high-quality instructions respectively into a new factor.

A.4.8 Experiment 2: Accuracy distribution

We here show the distribution of accuracy scores for all setups in experiment 2, separated by model size (hue) and whether the model is instruction tuned or not (i.e. vanilla).

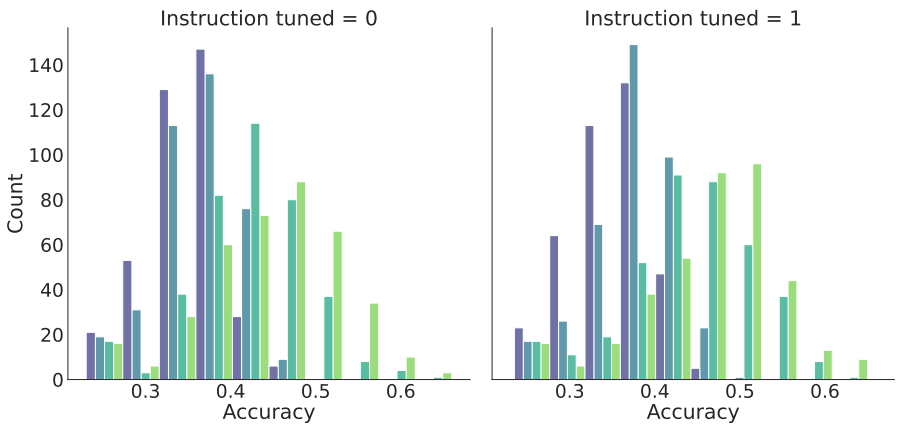


Figure A.16

A.4.9 Experiment 2: Interactions details

A.4.9.1 ANOVA using `instructions` factor

We fit an ANOVA using the factor `instructions` instead of `instruction quality`. In that case, we find a similar pattern of interactions, showing that the size of the main effect can not merely explain the number of interactions.

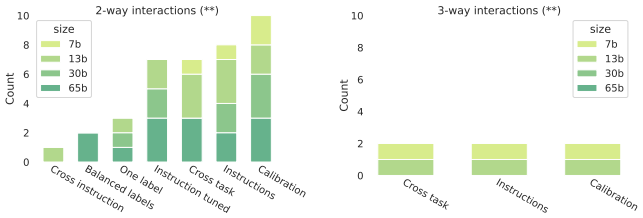


Figure A.17: Interactions when excluding `Instruction quality` and keeping `Instructions` instead. We find similar patterns.

A.4.9.2 Interaction mappings and effect sizes

The following shows the exact mapping of the interacting factors as well as the size of the corresponding effect size, measured by $\beta_{\lambda_1 \times \lambda_2}$ values from a post hoc regression analysis.

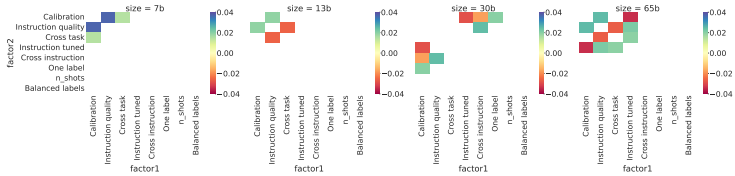


Figure A.18: The exact mappings of all two-way interactions in our experiments.

Additionally, the following table lists the exact mapping of the significant three-way interactions between different factors, as measured by $\beta_{\lambda_1 \times \lambda_2 \times \lambda_3}$ of the post hoc regression analysis.

Model	λ_1	λ_2	λ_3	$\beta_{\lambda_1 \times \lambda_2 \times \lambda_3}$
7B	Instruction quality	Calibration	Cross task	0.037106
13B	Instruction tuned	Calibration	Instruction quality	0.002102
13B	Instruction quality	Cross task	Calibration	-0.013176

Table A.6: The exact mappings of all three-way interactions in our experiments.