Exploring the limits of Systematicity of Natural Language Understanding Models

Koustuv Sinha

School of Computer Science
McGill University
Montreal, Canada

November 2022

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Acknowledgements

First and foremost, I would like to thank my supervisor, Joelle Pineau, for her endless support, motivation and guidance; being an incredibly patient mentor and role model for conducting scientific research; involving me in the quest to achieve reproducibility in machine learning; and continually providing me opportunities to learn and grow during my PhD. I consider myself incredibly lucky to have such a kind and enthusiastic mentor in my life.

I would like to thank my close collaborators Adina Williams, Shagun Sodhani and Prasanna Parthasarthi for their guidance, endless support and motivation throughout many projects that are included in this thesis or otherwise. Furthermore, I would like to thank my incredible mentors with whom I have been fortunate enough to collaborate and learn how to conduct thorough scientific research during my thesis: William L. Hamilton, Dieuwke Hupkes, Douwe Kiela, Robin Jia, Timothy O'Donnell and Adi Renduchintala. Furthermore, I would like to thank my professors in the RL Lab and beyond, who have helped me foster my research interests in natural language processing: Siva Reddy, Jackie Cheung, Dima Bahdanau, Derek Ruths and Andrew Piper.

I'd also like to thank my collaborators and co-authors for their help and support for the research materials involved in this thesis and related works: Amirhossein Kazemnejad, Jin Dong, Emily Goodwin, Nicolas Gontier, Ryan Lowe and Jasmine Wang.

My life in Montreal started back in 2016 with my Masters, then followed by my PhD. In the last six years, I am lucky and grateful to all my friends who made my life

away from home feel like home, including: Sumana Basu, Upasana Dasgupta, Haji Mohammad Saleem, Shagun Sodhani, Sayantan Datta, Arna Ghosh, Haque Ishfaq, Attia Amin Oni, Matt Gittings from McGill University, Nicolas Gontier, Harsh Satija, Jad Kabbara, Malik Altrakrori, Kushal Arora, Khimya Khetarpal, Charles Onu, Lucas Caccia, Joey Bose, Arushi Jain, Ayush Jain, Jonathan Lebensold, Maxime Wabartha, Emmanuel Bengio, Yue Dong, Audrey Durand, Nadeem Ward, Riashat Islam and the entire RL Lab/Mila, and all my friends from my Masters, including Peter Henderson, Caitrin Armstrong, Deven Patel, Ramchalam K. Ramakrishnan and Jaspal Singh, all of whom who have a special place in my heart.

I fondly recall my time at Samsung Advanced Institute of Technology, Korea where I did my first internship of my PhD. A special thank you to all at SAIT for making my time at Korea something to cherish and fondly remember for years to come, including Young Sang Choi, Sanghyun Yoo, Jehun Jeon, Ki Soo, Park Jong Hun and the entire Language Technologies team.

During my PhD I spent significant time being an intern at Meta AI Montreal. I'd like to thank all my colleagues who made those days enjoyable and fruitful in both research and life experiences: Shagun Sodhani, Vinayak Tantia, Mike Rabbat, Daniellea Kafla, Adriana Romero, Michal Drozdzal and many others. A special shoutout to the security and culinary teams at Meta Montreal for keeping me company during late night sessions and providing an unforgettable culinary experience. I'd also like to thank all my colleagues in Meta AI who have provided technical guidance and support throughout my internship, including Jack Urbaneck, Emily Dinan, Shruti Bhosale, Shubho Sengupta and many others.

This thesis consists of experiments which required a significant amount of compute. Thus, I would like to thank the teams at McGill, Mila, Meta and Compute Canada responsible for maintaining the compute clusters and providing us a seamless experience to run countless experiments on demand. In particular, I would like to thank

Ron Simpson, Andrew Bogecho, Corey Barton Antoniuk from McGill; Olexa Bilaniuk, Bruno Travouillon, Ahmed Mamlouk, Frederic Osterrath and the entire Mila IDT team; and the Penguin Compute Team from Meta for providing extensive technical support and assistance during the experiments conducted in this thesis.

Over the course of my PhD I was actively involved in leading the annual Machine Learning Reproducibility Challenge, which would not have been possible without the constant support, motivation and guidance of my co-organizers: Jessica Forde, Jesse Dodge, Sasha Luccioni, Robert Stojnic, Sharath Raparthy and Joelle Pineau; Nicolas Rougier from ReScience and the OpenReview team for providing the technical support.

The Covid pandemic occurred during my PhD, which led to an exciting opportunity to contribute towards understanding the disease in a data-driven way, thanks to my supervisor Joelle Pineau. I would like to thank my colleagues from Meta AI, Matthew Muckley and Anuroop Sriram, with whom I worked closely during this period to develop chest X-ray imaging pipelines, and who helped me develop an alternate research interest in medical imaging and taught me to conduct rigorous scientific studies.

I would like also to extend by sincere thanks and gratitude to the incredible professors and mentors I had during my pre-Masters days in India, who helped nurture my interest in computer science research, including Ee-Kian Wong, Saptarsi Goswami, Sukalyan Goswami, Tamal Chakraborty, Nilanjana Dutta Roy from Institute of Engineering and Management, Kolkata; Koumudi Patil and Arnab Bhattacharya from Indian Institute of Technology, Kanpur; and Debjit Chakraborty from Ramakrishna Mission Vidyalaya, Narendrapur.

I am grateful to have long standing relationships with my childhood friends from school and undergrad, who despite being physically located miles apart check on me from time-to-time to ensure my mental health is okay: Anikendu Bose, Pamela Roy, Sounak Mohanta, Debojyoti Roy Malakar, Subhodeep Santra, Anindya Chatterjee, Aritra Chatterjee, Soumalyo Sarkar, Abhishek Rudra and many wonderful people at Ramakrishna Mission Vidyalaya, Narendrapur and Institute of Engineering & Management, Kolkata.

Finally, I grateful to my family for being there through thick and thin, cheering for me and supporting me even though multiple continents separate us, including my parents, *Ma* (Supriya Sinha), *Baba* (Kanchan Kumar Sinha), my sister *June* (Adrija Sinha), my in-laws *Maa* (Sanchita Basu), *Bapi* (Manas Basu), *Didibhai* (Arati Basu), brother-in-law *Jerry* (Ananda Basu) and especially my wife, Atrayee Basu. My wife gets a special acknowledgement, for loving me more than I deserve, for constantly supporting, encouraging and motivating me, and for being my constant source of inspiration, when we are together, and even when we are apart. Love you all!

Abstract

In this thesis, we investigate several approaches to evaluate modern neural language models through the lens of *systematicity*, in order to assess their human-level reasoning and comprehension of natural language. First, we investigate the model's limits in encoding the natural language semantics by proposing a diagnostic challenge dataset known as CLUTRR. Drawing inspiration from first-order logic, this dataset specifically tests for systematicity in length generalization in natural language understanding models, in the form of a question-answering task. We observe most major models fail in generalizing to longer chain of reasoning, with the main limitation arising from their rudimentary understanding of syntax. Next, we apply the principles of systematicity to evaluate the syntax encoding strategy of large language models by applying permutations to the word order seen during inference and training. We observe a surprising fact that a trained neural language model can still perform optimally when subjected to sentences of shuffled word orders, devoid of their original meaning, and furthermore they can even improve their performance significantly on specific permutations. Next, we investigate the reasons of such behavior by pre-training large language models on meaningless, word-order shuffled corpora, to find they too behave optimally on downstream semantic and syntactic tasks. These results highlight the potential distributional nature of large language models, such that they only focus on n-grams during computation. Finally, we attempt to investigate the root cause of these effects, to find the component of the model most responsible. We observe that certain classes of position embeddings lead the models to overfit on the token positions, subjecting models to exhibit un-systematic behavior on out-of-position sentences. In summary, this thesis attempts to shed more light to the black box nature of the state-of-the-art neural language models, and introduces mechanisms to test and ensure systematic behaviors in their understanding of natural language.

Abstract in French

Dans cette thèse, nous étudions plusieurs approches pour évaluer les modèles neuronaux de langage modernes à travers le prisme de la systématicité, afin d'évaluer leur raisonnement et leur compréhension du langage naturel à l'échelle humaine. Tout d'abord, nous étudions les limites du modèle dans l'encodage de la sémantique du langage naturel en proposant un jeu de données de défi de diagnostic connu sous le nom de CLUTRR. S'inspirant de la logique du premier ordre, ce jeu de données teste spécifiquement la systématicité de la généralisation des longueurs dans les modèles de compréhension du langage naturel, sous la forme d'une tâche de question-réponse. Nous observons que la plupart des principaux modèles ne parviennent pas à généraliser à des chaînes de raisonnement plus longues, la principale limitation provenant de leur compréhension rudimentaire de la syntaxe. Ensuite, nous appliquons les principes de la systématicité pour évaluer la stratégie d'encodage syntaxique des grands modèles de langage en appliquant des permutations à l'ordre des mots vu pendant l'inférence et l'entraînement. Nous observons un fait surprenant : un modèle de langage neuronal entraîné peut encore obtenir des performances optimales lorsqu'il est soumis à des phrases dont l'ordre des mots est mélangé, dépourvues de leur signification originale, et, de plus, il peut même améliorer ses performances de manière significative sur des permutations spécifiques. Ensuite, nous étudions les raisons d'un tel comportement en pré-entraînant de grands modèles de langage sur des corpus de mots mélangés sans signification, pour constater qu'ils se comportent également de manière optimale sur des tâches sémantiques et syntaxiques en aval. Ces résultats mettent en évidence la nature distributive potentielle des grands modèles de langage, tels qu'ils ne se concentrent que sur les n-grammes pendant le calcul. Enfin, nous essayons d'étudier la cause profonde de ces effets, afin de trouver le composant du modèle le plus responsable. Nous observons que certaines classes d'encastrements de position conduisent les modèles à s'adapter de manière excessive aux positions des tokens, ce qui les amène à présenter un comportement non systématique sur les phrases hors position. En résumé, cette thèse tente de mettre en lumière la nature de boîte noire des modèles neuronaux de langage de pointe, et introduit des mécanismes pour tester et assurer des comportements systématiques dans leur compréhension du langage naturel.

Contributions to Original Knowledge

This thesis consists of original contributions towards understanding the reasoning process employed by natural language understanding models, including pre-trained large language models. Specifically, I make the following contributions:

1. A highly configurable benchmark dataset to diagnostically evaluate implicit reasoning in length generalization settings, along with benchmarks on popular neural language models (Chapter §3).

2. An original analysis and mechanisms to evaluate the sensitivity to word order for modern neural language models (Chapter §4).

3. A comprehensive analysis on pre-training large language models with shuffled word orders to understand their nature of sentence processing, along with publicly released code and model checkpoints (Chapter §5).

4. An investigation on a specific component of modern neural language models which allows the model to understand the relative position of words in a sentence, to understand the reasoning behind their insensitivity to word order (Chapter §6).

For all the above contributions, the code and data used are publicly released [1] [2] [3] [4].

[1] https://github.com/facebookresearch/clutrr

[2] https://github.com/koustuvsinha/clutrr-baselines

[3] https://github.com/facebookresearch/unlu

[4] https://github.com/facebookresearch/fairseq/tree/main/examples/shuffled_word_order

Contributions of Authors

- Chapters §1 and §2 consists of introduction and technical background material for this thesis which are written by me, with inspiration from various sources including the theses of Ryan Lowe and Prasanna Parthasarathi.

- Chapter §3 is based on Sinha et al. [2019], which is a published conference paper (Oral presentation) at *Empirical Methods of Natural Language Processing* (EMNLP) 2019. In this work, I led the project: I developed the dataset, ran experiments with baselines and contributed in writing the paper. Shagun Sodhani helped in developing the dataset, planning the experiments and contributed in paper writing. Jin Dong contributed in development of some baselines and verification of the code. Joelle Pineau and William L. Hamilton provided technical guidance, leadership, and contributed in paper writing, with William leading the effort in providing guidance on the direction, experimental design and writing of the paper.

- Chapter §4 is based on Sinha et al. [2021b], which is a published conference paper (Oral Presentation) at *Association for Computational Linguistics* (ACL) 2021. My paper was awarded the *Outstanding Paper Award* at this conference. In this work, I led the project by developing the experiments and analysis of the paper and contributed towards writing. Prasanna Parthasarathi contributed towards implementing the non-Transformer baselines, and overall provided technical guidance for devising the experiments. Joelle Pineau and Adina Williams supervised the project and provided leadership, with Adina Williams leading the supervision effort by helping to brainstorm experiments and analysis and writing of the paper.

- Chapter §5 is based on Sinha et al. [2021a], which is a published conference paper (Poster Presentation) at *Empirical Methods of Natural Language Processing* (EMNLP) 2021. This work was led by me, where I developed and executed the experiments consisting of pre-training large language models and their subsequent analysis, and I wrote the paper. Robin Jia, Dieuwke Hupkes and Joelle Pineau provided technical guidance in devising the experiments of the project, and helped in writing of the paper. Adina Williams and Douwe Kiela supervised the project and contributed in writing, with Douwe Kiela providing the leadership by helping to devise the experimental protocol and analysis.

- Chapter §6 is based on Sinha et al. [2022], which is a conference paper published at the *Empricial Methods of Natural Language Processing* (EMNLP) Findings 2022 (Poster Presentation). In this work, I led the project along with help from my equal co-author Amirhossein Kazemnejad. Amirhossein and I wrote the code for the position shift experiments, where I focused on the grammatical acceptability and downstream task experiments and Amirhossein ran the experiments on phase shifting with prompting large language models. Both me and Amirhossein contributed in the writing of the paper. Siva Reddy and Joelle Pineau contributed in providing technical guidance and devising direction. Dieuwke Hupkes and Adina Williams supervised the project, and both contributed towards the writing of the paper.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Motivation

A principled focus in computer science research is to develop models which mimic human-like reasoning. Towards that objective, one of the key goals is to develop models which understand and reason on natural language. Natural language allows humans to express and comprehend a vast variety of novel thoughts and ideas, and it serves as a medium to convey higher-order reasoning and comprehension. Thus, endowing the reasoning and communication capabilities using natural language to machines has long been the goal of computer scientists. Early approaches involved formal rule-based symbolic approaches to represent language, in order to have an *interpretable* function to compose the form and meaning of written and spoken text [Chomsky, 1957, Van Benthem, 1995, Van Benthem et al., 2008, MacCartney and Manning, 2009]. However, this formal approach was unable to model the *ambiguity* of language, where the meaning of the text depends on its context. Thus, approaches based on distributional semantics evolved, which tracked the contextual meaning of word representations using continuous vectors [Mikolov et al., 2013a, Pennington et al., 2014]. Following the success of distributional processing, the modern *deep-learning* era of language representations evolved, where the entirety of language acquisition is data-driven powered by a neural network, leading to highly successful and robust Natural Language Understanding (NLU) models [Rosenblatt, 1958, Rumelhart et al., 1986, Hinton et al., 2006, Bengio et al., 2006, Hochreiter and Schmidhuber, 1997, Cho et al., 2014, Kim, 2014, Bahdanau et al., 2014]. However, these neural models forgo interpretability in favor of performance, leading to the black-box nature of their internal processing.

Over the last five years, a class of the deep neural models rose to the pinnacle of success: the Transformers [Vaswani et al., 2017] and its derivatives. Since its introduction in 2017, Transformer-based models have achieved impressive results on numerous benchmarks and datasets, with BERT [Devlin et al., 2019b] being one of the most popular instantiation of the same. Using a technique known as "pre-training" [Mikolov et al., 2013a, Peters et al., 2018], Transformer-based models are first trained to learn a powerful language representation by ingesting massive corpus of text in an *unsupervised* fashion. Through this kind of unsupervised training, the models learn and tune their millions and billions of parameters, and using which they solve downstream tasks with surprising, near-human efficiency [Devlin et al., 2019b, Liu et al., 2019b, Lewis et al., 2020b]. Due to the improvement in massive compute resources and availability of large text corpus, the community has witnessed a steady increase in the variants of the Transformer family of models, each having more number of parameters than their predecessors [Kaplan et al., 2020]. At the time of the writing of this thesis, the largest Transformer family of models, GPT3 [Brown et al., 2020] and OPT [Zhang et al., 2022b], consist of 175 billion parameters, and are able to not only solve many downstream tasks with surprising ease, but learn new tasks with fewer instructions.

While Transformer-based models excel in the current datasets and benchmarks, it is less clear why do they work so well, as the neural models themselves lack interpretability. Due to the sheer amount of over-parameterization, direct inspection of the inner workings of these models are limited. Thus, various research have been conducted by using auxilliary tasks and probing functions to understand the reasoning processes employed by these models [Rogers et al., 2020]. Transformer embeddings are also reported to contain syntactic information about a given input, to the extent that the model may internally perform several natural language processing pipeline steps, involving parts-of-speech tagging, entity recognition etc [Tenney et al., 2019]. The fact that Transformer-family of models are credited for acquiring some level syntactic [Hewitt and Manning, 2019, Jawahar et al., 2019a], semantic [Ettinger, 2020], and world knowledge [Petroni et al., 2019, Rogers et al., 2020], purely by self-supervision, suggests that pre-training with massive overparameterized models and large corpora might just be the perfect roadmap to achieve "human-like" reasoning capabilities.

However, practitioners in natural language processing routinely concur that even with these models we are still far from the goal of developing a model mimicking human-level intelligence [Kiela et al., 2021]. There is growing evidence in literature which highlight the brittleness of NLU systems to adversarial examples [Jia and Liang, 2017, Ettinger et al., 2017, Ettinger, 2020, Ribeiro et al., 2020], where a non-standard input is shown to drastically reduce model performance. Models have been demonstrated to be highly susceptible to character-level perturbations and changes [Ebrahimi et al., 2018a, Minervini and Riedel, 2018, Ebrahimi et al., 2018b, Wallace et al., 2019]. More so, there is strong evidence that state-of-the-art models tend to exploit statistical artifacts and heuristics in datasets, rather than exhibiting true reasoning and generalization capabilities [Gururangan et al., 2018, Poliak et al., 2018, Tsuchiya, 2018, Naik et al., 2018, McCoy et al., 2019a].

In view of the positive and negative evidences towards Transformers acquiring "human-like" natural language understanding capacity, it is very important that we take a step back and carefully examine the reasoning processes of these models, to make them more interpretable. In this thesis, I use the principles of *systematicity* to re-investigate the reasoning processes employed by the state-of-the-art models. The phenomenon of *systematicity*, widely studied in the cognitive sciences [Gentner and Toupin, 1986], refers to the fact that lexical units such as words make consistent contributions to the meaning of the sentences in which they appear [Fodor and Pylyshyn, 1988]. As an illustration, they provide an example that all English speakers who understand the sentence "John loves the girl" should also understand the phrase "the girl loves John". In case of NLU tasks, this accounts to model being consistent in understanding novel compositions of existing, learned words or phrases. Systematicity also defines a way to evaluate models by *composition*, in a way to compare the inner workings of a model to a formal, symbolic rule-based system. While rule-based systems were poor in representing natural language, they were consistent, interpretable and compositionally generalizable. In this thesis, I present my work to evaluate the modern NLU models in the lens of systematicity, so that we can assess their limits of reasoning. This will allow us to implement better safeguards to build more robust and generalizable models. Since Transformer-based models are now being deployed in production and decision making systems, it is even more prudent to test the models towards systematic understanding in order to avoid catastrophic scenarios.

1.2 Research Questions

Therefore, in this section I list the main research questions I investigate throughout this thesis:

- Are state-of-the-art NLU models systematic in their understanding of semantics? (Chapter §3)
- Are these models systematic in their understanding of the basic form syntax, such as word order? (Chapter §4)
- Do Transformer models learn language during pre-training in a systematic way?

(Chapter §5)

- Are the Transformer models systematic in their understanding of relative positions of words in a given input? (Chapter §6)

In the next section, I provide a brief overview of the results, and discuss the full results and experimental setups in detail throughout the thesis. Finally, I end the thesis with a conclusion and discussion of future work in Chapter §7.

1.3 Preview of the Results

Throughout my thesis, we find overwhelming evidence of systematicity issues of NLU models, including the state-of-the-art Transformer family of models. In Chapter §3, we find models are unable to *extrapolate* to longer sequences than the one they are trained on, and also they are unable to reason in a noisy setup where the input text contains irrelevant information [Sinha et al., 2019]. In Chapter §3, we also find evidence that deep neural models can do better in extrapolation only if they are subjected to rudimentary syntax, which is not the case for natural language.

Chapter §4 confirms our hypothesis of weak syntax encoding capabilities of modern neural networks [Sinha et al., 2021b]. Specifically, we find all models to be *insensitive* to perturbations in word order - so much so that they can operate on shuffled, meaningless input hallucinating it as if it is a natural sentence. Through various analysis we observe that the syntax encoding capabilities of these large neural models are rudimentary, in that they can only understand a shallow hierarchy of collection of words.

In Chapter §5, we get a glimpse of the true nature of natural language processing pipelines employed by the state-of-the-art Transformer models [Sinha et al., 2021a]. Instead of internally processing a given text through a classical NLP pipeline as claimed by several papers, we find Transformer models learn language by exploiting the distributional properties of words and their neighbors. We observe that pre-training a large language model on the same exact conditions by just changing the order of the words in the input text surprisingly *does not* change the downstream task performance significantly. Our results indicate that these Transformer models only need to learn n-gram statistics of tokens in order to arrive at an optimal representation.

Finally in Chapter §6, we gain a probable reason to why Transformer models are insensitive to syntax perturbations [Sinha et al., 2022]. Our results indicate certain class of Transformer models overfit on the position information, so much so that their sentence processing capabilities are significantly affected if the starting position of the sentence is shifted. Our results highlight a key component in the Transformer-family of models which requires more attention in order to induce systematicity in their reasoning : the position encodings.

Chapter 2

Background

2.1 Early methods for text representation

Natural Language is the primary medium to conduct reasoning, and correspondingly, most use cases of natural language processing involve reasoning in some form. The process of reasoning has been widely studied throughout the history of mankind, from Aristotle's *Prior Analytics*. During 19th and 20th centuries, theoreticians such as Boole and Fredge transformed reasoning into a rigorous mathematical science Van Benthem et al. [2008]. Historically, natural language meaning and logical inference have been studied under the notion of *formal semantics* Van Benthem [1995], where the natural language meaning is represented in a symbolic compositional structure and sentences are reduced to a logical formula MacCartney and Manning [2009]. In this notion, the grammatical syntax of a sentence along with the semantic interpretations are represented as a combination of expressions in some higher order logic. Formal semantic representations of language are extremely powerful and expressive as they can be fed to automatic theorem provers to perform various natural language processing tasks. However, designing such sophisticated theorem proving systems introduces a major challenge: the rules associated with the formal grammar have to be provided *a priori*, or learned. But learning such combinatorially expressive rules is hard and computationally expensive.

2.1.1 Formal semantics and logical representations of natural language

Formal semantics provides a powerful way to represent natural language in some form of higher order *logic*. Logic is usually represented in various formulas such as Propositional Logic, Predicate Logic, and Higher order Logic Kowalski [1979]. Each of these form of formulas are built on top of the other, which result in gradually more and more expressive powers of the reasoning logic. For instance, the central component of propositional or first-order logic is represented by a set of if-then *rules*, or clauses. A clause is a rule of the form:

$$\alpha \leftarrow \alpha_1, ..., \alpha_m \tag{2.1}$$

where, α is referred as an atom, and the left-hand side (LHS) of the above equation is known as *head* and the right-hand side (RHS) is referred as *body*. The rules are composed by chaining through the *and* relation ($\wedge$).

Formal semantics offers a mechanism to encode natural language by the usage of a collection of sound *rules* or clauses, using which all possible combinations (conjuction and disjunction) of the formula can be derived using in first-order predicate calculus Hofstadter [1980]. This field in formal semantics is known as *deductive logic*. In the early 1970's, Colmeraurer and Kowalski [1979] were instrumental in development of a logic based programming language called Prolog. Prolog performs restricted first-order logical resolution using lazy backward-chaining. Using Prolog, one can specify a set of rules such that it can verify or deduce any given logical formulae.

However, deductive logic requires a pre-defined set of *rules* governing natural language, which is an untractable problem. Therefore, formal semantics also provides a mechanism to *learn* the rules from the data, using *inductive reasoning*. This kind of learning method belongs to the field of Inductive Logic Programming (ILP) Džeroski [2009]. In this line of approach, one starts with a background knowledge or Knowledge Base (KB) of predicates, and then learns rules or clauses from the given data. KB are often highly incomplete Min et al. [2013], and thus ILP can be used to induce rules and infer missing facts by reasoning over the existing relations and predicates. ILP systems can be broadly classified into *forward-chaining* and *backward-chaining* systems. Forward-chaining, also known as *top-down* approaches use *generate-and-test*: they generate clauses from a language definition, and test the generated programs against the positive and negative examples. Backward chaining or *bottom-up* approaches start by examining features of the examples, extract specific clauses from those examples, and then generalize from those specific clauses.

Formal semantics represent sentences in logical form which can be fed to an automated theorem prover to perform textual inference, question answering, machine translation and a host of other tasks. These systems are highly interpretable, as the rule deduction or inductive decisions are backed by formal proofs in the form of built clauses. However, the sophistication of such semantic models comes at a cost: the complex set of rules allowing for logical interpretation of text must either be provided or learned *a-priori*. For language, this boils down to maintaining an inventory of semantic relations MacCartney and Manning [2009] which is built by domain experts and thus incorporates bias and which also cannot be exhaustive.

Natural language itself is symbolic and discrete, however the discrete symbols evoke different interpretations based on the context. Thus, symbolic methods are unable to handle noisy, erroneous, ambiguous or polysemous nature of text data. Language is compositional, but the way in which words (discrete symbols) are composed (combined) to form meanings is practically infinite. In the next section, we therefore discuss a class of models which account for this ambiguity in data through the use of vectors and differentiable functions.

2.1.2 N-gram Language Models

Natural language is highly *ambiguous* (syntactic ambiguity, word sense ambiguity, semantic ambiguity) and *contextual*, i.e., it has multiple interpretations based on usage. Hence, to account for the flexibility of natural language use, *probabilistic* models of natural language reasoning were built, which attempt to learn the associated rules of reasoning through empirical data. This led to the era of *statistical relational learning* [Glickman et al., 2005, Han et al., 2006, Domingos and Richardson, 2006] which modelled the uncertainty of reasoning by starting with a *prior* belief and updating it to a *posterior* belief based on the *evidence* of the data. One such popular probabilistic model to represent natural language is an n-gram *language model*. In its core, n-gram language model are a family of Markov models that estimate the probability of a word given a fixed window of prior words. These estimations can be computed by counting the probability of occurrence of such words, using the maximum likelihood estimate. For example, the probability of a sentence S containing n words can be decomposed based on the Markov assumption that the probability of a word only depends on the previous words as follows.

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_{<i}) \tag{2.2}$$

In n-gram language models, the assumption is relaxed (independence assumption) such that a word can only depend on the probability of n previous words. Thus, for a bi-gram model, the probability of a word only depends on the previous word, and thus can be computed by counting the bi-grams in the corpus involving the word divided by all possible bi-grams.

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=2}^{n} P(w_i | w_{i-1}) \tag{2.3}$$

Using the probabilistic interpretation, natural language sequences can be modelled by learning the probability estimates from a large corpora. The probability estimate (or likelihood, after application of log) is evaluated as a measure for the models reasoning abilities, such that syntactically and semantically well-formed sentences are expected to estimate a higher likelihood.

2.1.3 Distributional semantics using vector representations

The n-gram language modelling approach still uses distinct symbols and a count-based approach. Better generalization can be obtained in natural language reasoning by operating on embedded vector representations of discrete symbols, in which vector similarity can be interpreted as semantic similarity [Joos, 1950, Harris, 1954, Firth, 1957]. Distributional semantics, thus, capture the vector representations of symbols, such as words, to compose a meaning representation of a sentence in context of other words. These methods are also called sub-symbolic methods as they operate on the vector representations of the symbols themselves [Neelakantan et al., 2015].

For instance, in vector semantics a word is not represented as a discrete token as in formal semantics, but is represented as a point in high dimensional space, or a vector embedding. These embeddings can be either *sparse* or *dense*. Sparse embeddings of words are usually computed from a given corpora, which capture the co-occurrence counts of neighbors of the given word in the document, using term-frequency [Luhn, 1957], inverse document frequency [Jones, 1972] (tf-idf) and pointwise positive mutual information (PPMI) [Fano, 1949] measures. Dense vectors represent a word in the order of high dimensions, and are typically computed using the Word2Vec algorithm, such as skip-gram, which trains a logistic regression classifier to compute the likelihood of two words to occur nearby in the corpora. Another method, Glove [Pennington et al., 2014] is also used to represent dense vectors for words, which is based on learning a global word co-occurrence probabilities. The use of vectors enables us to compute *similarity* between two words, sentences or documents, by computing the dot product between the vector representations. This similarity metric is further used to train the vector representations of words using the Word2vec [Mikolov et al., 2013a] or Glove [Pennington et al., 2014] algorithms, using efficient optimization algorithms such as stochastic gradient descent (SGD), allowing learning from large training corpora. These features highlights the strength of vector representations to model the ambiguity in language.

2.2 Neural Inductive bias of text representation

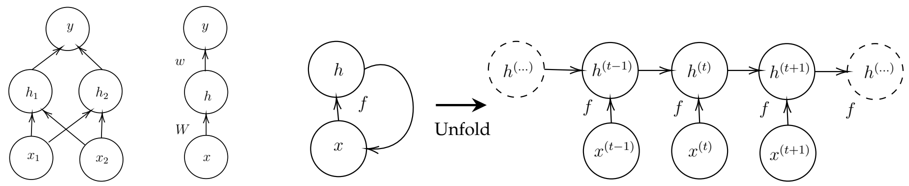
Neural Networks are a class of biologically inspired connectionist models which operates on continuous vector representations and thus they are also more robust to noise and ambiguity of natural language [Wang and Jiang, 2016, Goodfellow et al., 2016]. Neural networks offer significant representational capacity to learn rich vector representations of words, sentences and documents, and also does not require feature engineering. Due to its representational strength, most modern natural language understanding (NLU) models have thus built on top of neural models having distributional vector representation of the lexical items. In its simplest form, a feed-forward neural network consists of fully-connected networks in multiple layers, where each layer consists of an affine combination of vectors and non-linear activation function (tanh, sigmoid or rectified linear units) [Rosenblatt, 1958, Rumelhart et al., 1986, Hinton et al., 2006, Bengio et al., 2006]. Further improvements were made to handle the polysemous and context-dependent nature of words by using contextual embeddings through leveraging the sequences of sentences, typically using Recurrent (Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997], Gated Recurrent Units (GRU) [Cho et al., 2014]) or Convolutional models (CNN) [Kim, 2014]. Models based on Long Short Term Memory (LSTM) became the de-facto standard in NLU, as they can effectively capture the contextual information with locality bias present in natural language. However, its performance is often affected by the long-term dependency problem in several NLU tasks. Recently, the field of NLP has witnessed a paradigm shift in research with the advent of Transformers [Vaswani et al., 2017]. Being a fully-connected multi-head self-attention model, Transformers can directly model the dependency between any two words in a sequence, thus proving to be more powerful and suitable to model long range dependencies of natural language. However, Transformers come with massively increased model parameters, thus requiring significantly large corpus to train and resulting in overfitting on small and medium sized datasets [Radford et al., 2018].

2.2.1 Feed Forward Neural Networks

A Feed-Forward Neural Network (FNN) consists of a collection of neurons which are connected to other neurons via synapses or weights Rosenblatt [1958]. A FNN is further modelled into layers represented as input, hidden or output vectors and weights connecting input and hidden layers, and hidden and output layers. There can also be weights connecting hidden and hidden layers when the number of hidden layers is more than one. A simple formulation of an FNN can be given as:

$$f^{(i)} = g(W_i x + b_i) \tag{2.4}$$

Activation for a node i in any layer l, $f_l^{(i)}$, is computed as a function of the weights connecting that node from the previous layer and the input from previous layer. Every hidden and output layer has a non-linear function to project the activations, g, which are usually *sigmoid*, *tanh*, *rectilinear* or *softmax* functions. Learning happens by estimating the gradients for the weights using backpropagation [Rumelhart et al., 1986] and updating using stochastic gradient descent with a learning rate, ϵ, that reflects the magnitude of descent in the direction of the gradient.



(a) FNN (b) RNN

Figure 2.1 A Feed Forward Network (FFN) and a Recurrent Neural Network (RNN)

2.2.2 Recurrent Neural Networks and LSTMs

RNN employ a feedback connection by repeatedly applying FNN (Figure 2.1). The addition of the feedback loops to the hidden units of an MLP can be thought of forming a hidden state of the network $h(x)$, corresponding to the vector of activations (value after the non-linearity) of the hidden neurons, that evolves over time as we present inputs to the network. The hidden state is updated at each time step according to some function f:

$$h_t = f(h_{t-1}, x_t) \tag{2.5}$$

RNNs are learned by computing gradients through time, which is also known as Back Propagation Through Time (BPTT) [Rumelhart et al., 1986].

While repeatedly applying the FNNs to create an RNN, the architecture faces a critical “vanishing and exploding gradient" problem [Hochreiter et al., 2001], i.e., the gradients either become too insignificant or too big while backpropagating through multiple non-linear layers. To overcome this problem, Long Short-Term Memory [Hochreiter and Schmidhuber, 1997] was proposed which consists of a specific gating architecture to overcome the problem to a great extent. The standard LSTM architecture is described in Figure 2.2.

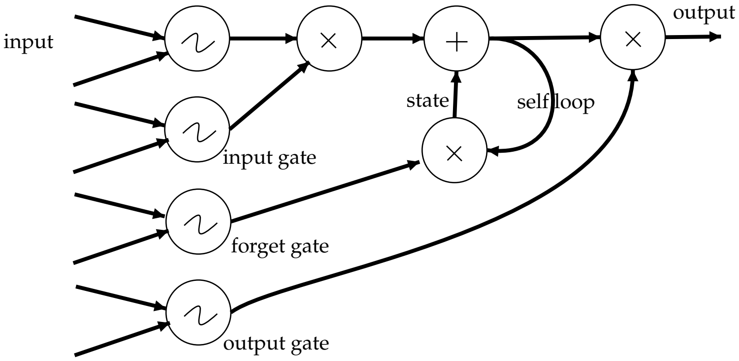


Figure 2.2 Block diagram of LSTM cell.

Given an input sequence, LSTM can also be run in both directions to get two final representations $\overrightarrow{y_t}$ in the forward direction and $\overleftarrow{y_t}$ in the backward direction. This class of modified LSTM is known as Bidirectional LSTM and it has been used as a standard feature extractor in various natural language processing tasks Rocktäschel et al. [2015], Wang and Jiang [2016], Serban et al. [2016]. In language modelling & machine translation tasks [Serban et al., 2016, 2017, Bahdanau et al., 2014], BiLSTM is used to construct an *encoded* representation of the input text, conditioned on which the next word in the output text is *decoded*. For textual entailment tasks [Bowman et al., 2015], two individual sentence representations of the hypothesis and premise are calculated using BiLSTM, and the entailment is predicted over the concatenated representation [Rocktäschel et al., 2015]. For QA tasks, the natural language question is encoded through a BiLSTM and the sentence representation is used in downstream reasoning [Wang and Jiang, 2016]. Given the gated memory structure of LSTM and bidirectional read, they provide a powerful mechanism to convert text into a fixed vector representation which captures various syntactic and semantic features of natural language [Conneau et al., 2018b].

2.2.3 Transformer Models

In contrast to RNNs and LSTMs, Transformer [Vaswani et al., 2017] architectures eliminate the need of recurrent connections and instead use fully connected self-attention layers. Attention mechanisms [Bahdanau et al., 2014] were previously developed to improve RNNs and LSTMs to allow the model to condition on longer contexts than what the default recurrence provides. Transformer architecture leverages this attention mechanism to attend on the same input (hence the name "self-attention") in multiple layers, this allowing the model to bypass the need of recurrence and directly utilize the information from any context.

Self-attention

Transformer architecture maps an input sequence $(x_1, x_2, \ldots, x_n)$ to an output sequence $(y_1, y_2, \ldots, y_n)$ of the same length, hence being a transductive model. Self-attention consists of computing vector similarities, using the dot product, between a pair of inputs:

$$\text{score}(x_i, x_j) = x_i . x_j \tag{2.6}$$

This dot product yields scalar values in $\mathbb{R}$. Attention from each input to any other input is computed as the softmax over the dot product value in order to capture the proportional probability of similarity between tokens. This attention value is then multiplied by the inputs to compute the output at a sequence element. Every input is restricted to attend to all inputs beyond its position, thereby mimicing the canonical language modeling setup (although Markov assumptions no longer hold here due to the linear combination).

$$\alpha_{i,j} = \frac{\exp(\text{score}(x_i, x_j))}{\sum_{k=1}^{i} \exp(\text{score}(x_i, x_k))} \tag{2.7}$$

$$y_i = \sum_{j \leq i} \alpha_{i,j} x_j \tag{2.8}$$

Using this basic formulation, Transformers first map the inputs into three distinct vectors, *query* (q), *key* (k) and *value* (v). When computing the attention from the current input to a preceding input, the model chooses to compute the score between the query mapping of the current input against the key mapping of the preceding input. This score is then using to linearly combine the value mappings of preceding inputs to compute the output for the current token. In practice, this score can become unbounded and lead to numerical issues. Therefore, the score is normalized by dividing with the square root of the dimensionality of the query and key vectors.

$$q_i = \theta_q x_i, k_i = \theta_k x_i, v_i = \theta_v x_i \tag{2.9}$$

$$\text{score}(x_i, x_j) = \frac{q_i . k_j}{\sqrt{d_k}} \tag{2.10}$$

$$y_i = \sum_{j \leq i} \alpha_{i,j} v_j \tag{2.11}$$

This attention mechanism is now stacked in two dimensions. First, the same attention mechanism is applied multiple times using unique set of weights. This is known as *multi-head* attention, to emulate multiple heads (or weights) being utilized to compute unique sets of attentions for the same current and preceding input pair. The resulting attention mechanism is referred to as an *attention-layer*. Second, the attention layers can be stacked on top of each other to emulate a multi-layer neural network. Typically, a feed-forward layer, layer normalizations [Ba et al., 2016] and residual connections are applied after one attention layer to convert it to a *transformer block*, which is then stacked vertically in multiple layers (Figure 2.3).

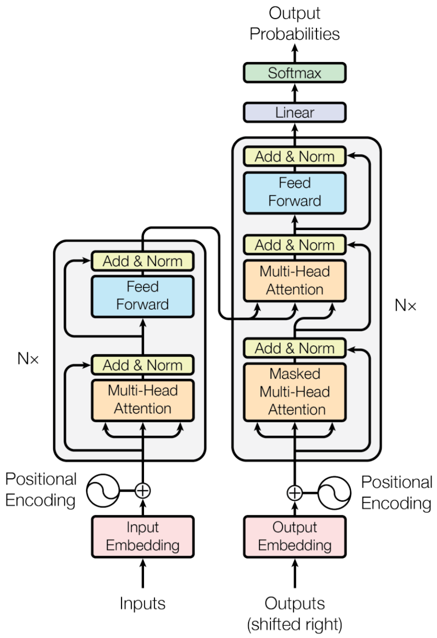


Figure 2.3 Transformer encoder block as described in Vaswani et al. [2017].

Position embeddings

One drawback of the vanilla Transformer architecture is that word order cannot be represented, as due to linear combination of attention scores the individual inputs can be re-arranged to get the exact same output. Thus, position information is fed to the model by representing the position of each token using a feature vector. Vaswani et al. [2017] use absolute word position information as a feature vector, which is derived from an alternating sine and cosine functions:

$$\begin{aligned} \mathrm{PE}(pos, 2i) &= \sin(pos/10000^{2i/d}) \\ \mathrm{PE}(pos, 2i+1) &= \cos(pos/10000^{2i/d}) \end{aligned} \tag{2.12}$$

PE is the position embedding matrix of size $m \times d$, where m is the total number of positions supported by the model, and d is the dimensionality of the positional embedding. d is kept as same as the embedding dimensionality of the input x_i such that the input representation is updated with the positional features by element-wise sum *before* applying the self-attention mechanism.

$$x_i = x_i + \text{PE}[i] \tag{2.13}$$

Subsequently, a unique learned position feature is used to define the absolute position encoding [Devlin et al., 2019b]. Instead of using a fixed function, the model can use a learned positional matrix $\theta_{PE} \in \mathcal{R}^{m \times d}$ of the same dimension (commonly referred to as learned positional embedding), where the features are learned by backpropagation during training.

While absolute position encodings, especially learned position embeddings are still used by many variants of the Transformer models, it lacks the capacity to model relative position distances between tokens [1]. Thus, relative positional encodings were proposed [Shaw et al., 2018] which directly encode the relative distance between the words. Specifically, instead of maintaining a large embedding matrix (θ_{PE}), a smaller embedding matrix is now maintained based on a window size r, and dynamically compute the position encoding of a token based on the relative distances between the current token and its neighbors within the window of size r. Please refer to Wang et al. [2021], Dufter et al. [2021] for a comprehensive overview of different position encoding strategies.

2.2.4 Pre-training and the advent of large language models

Transformer architectures introduce mechanisms to drastically increase the amount of trainable model parameters, compared to RNN and LSTM counterparts. With increasing number of model parameters, the requirement to train on large datasets be-

[1] In Chapter 6, we investigate in detail several shortcomings of the learned position embedding scheme

came inevitable - as otherwise it is not possible to fully train the model parameters. However, it is challenging to design tasks of such scale due to extremely expensive annotation costs. In contrast, it is relatively easy to obtain large-scale unlabeled corpora, typically through scraping the internet Raffel et al. [2020] or collecting all literary works Zhu et al. [2015]. Thus emerged the idea of *pre-training* [Mikolov et al., 2013a, Peters et al., 2018], where a massive, unlabeled corpus is leveraged to train a *language model*, i.e. predicting the probability of tokens given the context (§2.1.2). Given an input context, the aim of a given model is to compute a representation of the same, in order to predict the next token probability distribution, typically in the form of a softmax function over the full vocabulary. Learning such a language model on top of the massive corpora yields better contextual token representation, which are then utilized to perform various downstream tasks. The large, unlabeled data can be leveraged to learn useful word embeddings either using pairwise ranking [Collobert and Weston, 2008], or through the use of shallow architectures [Mikolov et al., 2013a], or by computing global word-word co-occurence statistics [Pennington et al., 2014]. However, these pre-trained word embeddings are still context-independent, and requires learning context-dependent parameters from scratch on the downstream task. Thus, efforts have been made to pre-train the entire model parameters instead of generating word embeddings by training in a language modeling objective, using bidirectional LSTM based architectures [Peters et al., 2018]. These models are then used to fine-tune on downstream tasks such as text classification [Howard and Ruder, 2018].

More recently, Transformer-based architectures became widely popular with the introduction of BERT [Devlin et al., 2019b]. BERT is fundamentally a stack of Transformer layers, having multiple self-attention heads. To learn its massive amount of parameters, BERT is pre-trained on unlabeled corpora using two self-supervised training objectives: masked language modeling (MLM) and next sentence prediction (NSP) (predicting if two sentences are adjacent to each other). In MLM, an input sentence is corrupted by randomly masking some percentage of tokens, and the objective of the model is to leverage the context to predict the correct masked tokens. In NSP, BERT is provided an additional objective to predict if two given sentences are neighbors [2]. BERT significantly improved the state-of-the art in many downstream tasks - ranging from text classication, question answering, natural language inference, machine translation, summarizatio, to name a few. Subsequently, several flavor of Transformer models have been proposed, which can be broadly classified into three distinct classes:

- **Encoder-only models** This model family only consists of a n layered Transformer model as the encoder, such as BERT [Devlin et al., 2019b]. This class of models is utilized to extract generalized representations of sentences, which can be further used in downstream tasks. RoBERTa [Liu et al., 2019b], DistilBERT [Sanh et al., 2020], ALBERT [Lan et al., 2019], ELECTRA [Clark et al., 2020a] are other examples of this class of model.

- **Encoder-Decoder models** This is a two-step model, which was originally defined in Vaswani et al. [2017]. The model architecture contains two separate stacked Transformer layers, where the first stack is used to *encode* the sentence representation, and the second stack is used to generate the target text conditioned on the encoded representation (i.e., to *decode*). BART [Lewis et al., 2020b], mBART [Liu et al., 2020], T5 [Raffel et al., 2020] are more examples of this class of models.

- **Decoder-only models** In this family of models, they consist of only the decoder element to predict the text from a given input, without having a dedicated encoder block. The goal of these models if the predict the next token given a sentence or paragraph. GPT2 [Radford et al., 2019a], GPT-3 [Brown et al., 2020], Transformer-XL Dai et al. [2019] are models of this type.

[2] In the followup works, such as RoBERTa, the NSP objective has been subsequently discarded in favor of pure MLM objective on larger batch sizes.

Transformer models can also be classified based on the type of training objective used while pre-training. Broadly, they can be grouped into two classes:

- **Masked Language Models** Popularized by BERT [Devlin et al., 2019b], this family of models in its core uses the technique of predicting the masked tokens in a given corrupted input. Thus, these type of models are typically *bidirectional*, as in they have the ability to leverage contextual information in the input sentence from both directions. RoBERTa, BART, ALBERT, ELECTRA etc are examples of this class of Transformer models.

- **Autoregressive Language Models** On the other hand, the vanilla Transformer model was proposed as an autoregressive model, where the model does not have the contextual information from both directions of the input. These models mimic the classical language model setup (§2.1.2), where given a sequence of tokens the model only uses the previously seen tokens as context to generate the next set of tokens, hence they are also known as *uni-directional* models. GPT-2, GPT-3, T5, Transformer-XL are examples of this class of Transformer models.

2.3 Evaluation of text representation capacity

2.3.1 Standard tasks for measuring progress in NLU

With the advent of better and higher capacity models to represent text, the need for evaluating the progress in natural language understanding (NLU) gave rise to development of various tasks and datasets. Popular instances of the datasets include SQUAD [Rajpurkar et al., 2016] for Question Answering; SNLI [Bowman et al., 2015] and MNLI [Williams et al., 2018c] for Natural Language Inference; DecaNLP [McCann et al., 2018] and the venerable GLUE [Wang et al., 2018] and SuperGLUE [Wang et al., 2019] benchmarks for general language understanding. The GLUE benchmark was proposed to unify the measurement of progress in language understanding by using a collection of publicly released datasets:

- Corpus of Linguistic Acceptability [CoLA, Warstadt et al., 2019c]
- Stanford Sentiment Treebank [SST, Socher et al., 2013]
- Microsoft Research Paragraph Corpus [MRPC, Dolan and Brockett, 2005]
- Quora Question Pairs (QQP)[3]
- Multi-Genre Natural Language Inference [MNLI, Williams et al., 2018c]
- Question NLI [QNLI, Rajpurkar et al., 2016, Demszky et al., 2018]
- Recognizing Textual Entailment [RTE, Giampiccolo et al., 2007a, Haim et al., 2006, Giampiccolo et al., 2007b, Bentivogli et al., 2009]
- NLI formulation of the Winograd Schema Challenge [WNLI, Levesque et al., 2011].

The SuperGLUE benchmark [Wang et al., 2019] was proposed following the success of GLUE as a much harder benchmark. This benchmark consist of the following datasets:

- Boolean Questions [BoolQ, Clark et al., 2019a]
- Commitment Bank [CB, De Marneffe et al., 2019]
- Choice of Plausible Alternatives [COPA, Roemmele et al., 2011]
- Multi-Sentence Reading Comprehension [MultiRC, Khashabi et al., 2018]
- Reading Comprehension with Commonsense Reasoning Dataset [ReCoRD, Zhang et al., 2018]

[3] http://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

- Recognizing Textual Entailment [RTE, Giampiccolo et al., 2007a, Haim et al., 2006, Giampiccolo et al., 2007b, Bentivogli et al., 2009]
- Word-in-Context [WiC, Pilehvar and Camacho-Collados, 2019]
- Winograd Schema Challenge [WSC, Levesque et al., 2011]

Using these benchmarks, text understanding systems have been compared against each other to measure progress in the field. While RNNs and LSTMs exhibited modest performance in these tasks, the advent of attention mechanisms [Bahdanau et al., 2014] increased the performance to considerable extent. However, we witnessed meteoric improvements on these benchmarks with every release of a new variant of the Transformer model, which typically consists of increasingly large amount of parameters and have been progressively trained on greater amount of unlabeled data [Kaplan et al., 2020]. The performance rise has even skyrocketed beyond human performance on these tasks [Kiela et al., 2021, Figure 1]. This stunning progress with the pre-trained Transformer family of models on NLU taks have attracted greater investments in pre-training even larger models [Brown et al., 2020, Zhang et al., 2022b]; invited researchers to dig deep into the reasoning process employed by these models [Rogers et al., 2020]; and have invited a higher level of scrutiny of the model outputs [Jia and Liang, 2017, Gururangan et al., 2018].

2.3.2 Analysis and Interpretability

The sheer performance of pre-trained Transformer-based models following BERT created a watershed moment in NLU, as these models outperform previous LSTM-based models by a large extent, even surpassing human performance. Due to this success, interest peaked in the community to investigate the inner workings of this largely black box model [Rogers et al., 2020]. The investigation is primarily done using the tool of *probing*, which consists of learning a function on top of pre-trained representations to perform targeted assesment of linguistic information [Hupkes et al., 2018]. How well this probe learns a given signal can be seen as a proxy for linguistic knowledge encoded in the representations. Using these probes, it has been shown BERT representations contain adequate syntactic [Hewitt and Manning, 2019, Jawahar et al., 2019a], semantic [Tenney et al., 2019, Ettinger, 2020], and world knowledge [Petroni et al., 2019, Rogers et al., 2020].

2.3.3 Brittleness issues in NLU models

Despite the large performance gain and representations containing the necessary linguistic information to process natural language, state-of-the-art NLU models are often subject to scrutiny due to their unsystematic behaviors on specifically crafted test suites. NLU models are repeatedly shown to be brittle when subject to adversarial attacks Jia and Liang [2017], Ettinger et al. [2017], Ettinger [2020] to the input sentence forms Kaushik and Lipton [2018]. NLU models tend to exploit the statistical irregularities and annotation artefacts Gururangan et al. [2018], Poliak et al. [2018], Tsuchiya [2018] of a given datasets, resulting in failure cases on carefully crafted examples Naik et al. [2018], McCoy et al. [2019a]. NLU models are also subjected to tests of systematic generalization in context of semantic parsing in novel compositions Lake and Baroni [2018], shuffled argument structure in natural language inference Dasgupta et al. [2018], etc. On these tests, typically the NLU models fail to behave in the expected ways, unless trained with the same objective as the test sets.

2.3.4 The need for better, challenging benchmarks

How well have our best, state-of-the-art models have progressed in *human-level* language representation? We get a conflicting answer on the question. On one hand, we witness an ever increasing performance improvement on standard tasks with larger amount of parameters and longer pre-training with more data. On the other hand, we witness a stream of results showcasing the brittleness and inconsistencies of the model outputs, so much so that they fail on real world examples, unlike humans. Thus, attempts have been made to develop better benchmarks, such as Dynabench [Kiela et al., 2021]. In Dynabench, models are constantly evaluated in a loop where human annotators are tasked to find harder examples to elicit an undesirable result. Dynabench is a great step in the right direction, however it poses some caveats: continuous data collection from annotators is an expensive endeavour, and the challenge nature of data collection could lead to unnatural distribution shifts [Kiela et al., 2021, Section 4, pp. 7]. The community has also invested in building newer dataset benchmarks, and have progressed to measuring zero-shot and few-shot *in-context* learning evaluations to pose a significant challenge to the model. However, by nature the Transformer language models require pre-training on massive corpora, which could possibly lead to high distribution overlap between the model representation and the test dataset, resulting in memorization effects on the datasets constructed from public corpus [Carlini et al., 2021, 2022]. Thus, we need a better mechanisms and datasets to measure the real progress of text understanding using neural models.

2.4 Measuring language understanding progress using Systematicity

An alternative to measuring *human-level* language understanding progress is by employing the notions of *systematicity* in designing datasets and experiments. The term is first introduced by Fodor and Pylyshyn [1988], which is defined as *"the ability to produce/understand some sentences is intrinsically connected to the ability to produce/understand certain others"*. This definition comes with varying interpretations in the literature [Lake and Baroni, 2018, Bahdanau et al., 2019, Goodwin et al., 2020] and is heavily debated based on its interpretation [Szabó, 2012, Herbelot, 2020]. Systematicity is also closely tied to the concept of *compositionality*, which requires models to have the ability to compose and combine distinct functions to construct meaning representation [Kratzer and Heim, 1998, Szabó, 2012]. Another term synonymous to systematicity in context is *productivity*, which construes the ability of a model to generate infinite combinations of known constituents [Pullum et al., 2010]. For the purposes of this thesis, systematicity can be formulated into two categories:

- The ability of the model to understand the *recombination* of known parts and rules. In the classic example, a learner understanding the phrase "brown dog" and "black cat" also understands "brown cat" [Hupkes et al., 2020].
- The ability of the model to extract *consistent* contributions from words. For example, a learner understanding the phrase "John loves Mary" must also understand the contributions of the words "Mary loves John" [Goodwin et al., 2020].

It is heavily debated in the literature whether *connectionist* architectures, such as the neural models, could account for the notions of systematicity [Fodor and Pylyshyn, 1988, Smolensky, 1991, Marcus, 1998]. Nevertheless, the above definition of systematicity allows us to evaluate the *human-ness* of the language understanding models. Systematic models employ the notions of compositionality in their representations to understand the contributions of words in the given text. Thus, a model following the principles of systematicity should exhibit the following properties:

- A systematic model should be generalizable to out-of-distribution examples: any out-of-distribution example can be decomposed into known rules of syntax and meaning.
- It should be robust to outliers and adversarial inputs: any adversarial input will not conform with the application of known rules and thus can allow the model to circumvent processing on those inputs.

- It should be consistent in its understanding of words: if the rules required to understand a phrase is learned then they should yield consistent representations of the same phrase (or vice-versa, if the rule is not learned then a systematic model should not yield any useful representation for the given input).

In this thesis, we utilize the above principles of systematicity to evaluate the limits of natural language understanding of state-of-the-art NLU models.

Chapter 3

Understanding semantic generalization through systematicity

Natural Language Understanding (NLU) systems have been extremely successful at reading comprehension tasks, such as question answering (QA) and natural language inference (NLI). These tasks typically test for semantic generalization, where a model has to understand the meaning of the input sentence / passage in order to perform the given task. An array of existing datasets are available for these tasks. This includes datasets that test a system's ability to extract factual answers from text [Rajpurkar et al., 2016, Nguyen et al., 2016, Trischler et al., 2016, Mostafazadeh et al., 2016, Su et al., 2016], as well as datasets that emphasize commonsense inference, such as entailment between sentences [Bowman et al., 2015, Williams et al., 2018c].

However, there are growing concerns regarding the ability of NLU systems—and neural networks more generally—to generalize in a systematic and robust way [Bahdanau et al., 2019, Lake and Baroni, 2018, Johnson et al., 2017]. For instance, recent work has highlighted the brittleness of NLU systems to adversarial examples [Jia and Liang, 2017], as well as the fact that NLU models tend to exploit statistical artifacts in datasets, rather than exhibiting true reasoning and generalization capabilities [Gururangan et al., 2018, Kaushik and Lipton, 2018]. These findings have also dovetailed with the recent dominance of large pre-trained language models, such as BERT, on NLU benchmarks [Devlin et al., 2019b, Peters et al., 2018], which suggest that the primary difficulty in these datasets is incorporating the statistics of the natural language, rather than reasoning.

An important challenge is thus to develop NLU benchmarks that can precisely test a model's capability for robust and systematic generalization. Ideally, we want language understanding systems that can not only answer questions and draw inferences from text, but that can also do so in a systematic, logical, and robust way. While such reasoning capabilities are certainly required for many existing NLU tasks, most datasets combine several challenges of language understanding into one, such as coreference/entity resolution, incorporating world knowledge, and semantic parsing—making it difficult to isolate and diagnose a model's capabilities for systematic generalization and robustness.

In this work, we propose to use the properties of *systematicity* to test the limits of semantic generalization of modern neural networks (§2.4). As defined by Fodor and Pylyshyn [1988], systematicity test the ability of a system to understand the recombination of known parts and rules. Thus, inspired by the classic AI challenge of inductive logic programming [Quinlan, 1990], in this chapter I discuss my work on developing semi-synthetic benchmark designed to explicitly test an NLU model's ability for systematic and robust logical generalization [Sinha et al., 2019]. Our benchmark suite—termed **CLUTRR** (Compositional Language Understanding and Text-based Relational Reasoning)—contains a large set of semi-synthetic stories involving hypothetical families. Given a story, the goal is to infer the relationship between two family members, whose relationship is not explicitly mentioned. To solve this task, a learning agent must extract the relationships mentioned in the text, induce the logical rules governing the kinship relationships (e.g., the transitivity of the sibling relation), and use a combination of these rules to infer the relationship between a given pair of entities. Crucially, the CLUTRR benchmark allows us to test a learning agent's ability for *systematic generalization* by testing on stories that contain unseen combinations of logical rules. CLUTRR also allows us to precisely test for the various forms of *model robustness* by adding different kinds of superfluous *noise facts* to the stories.

3.1 Technical Background

3.1.1 Notations and Terminology

Following standard practice in formal semantics, we use the term *atom* to refer to a *predicate* symbol and a list of terms, such as $[\texttt{grandfatherOf}, X, Y]$, where the predicate `grandfatherOf` denotes the *relation* between the two *variables*, X and Y. We restrict the predicates to have an arity of 2, i.e., binary predicates. A logical *rule* in this setting is of the form $\mathcal{H} \vdash \mathcal{B}$, where $\mathcal{B}$ is the *body* of the rule, i.e., a conjunction of two *atoms* ($[\alpha_1, \alpha_2]$) and $\mathcal{H}$ is the *head*, i.e., a single atom (α) that can be viewed as the goal or query. For instance, given a knowledge base (KB) R that contains the single rule

$$[\texttt{grandfatherOf}, X, Y] \vdash [[\texttt{fatherOf}, X, Z], [\texttt{fatherOf}, Z, Y]], \tag{3.1}$$

the query $[\texttt{grandfatherOf}, X, Y]$ evaluates to true if and only if the body

$$\mathcal{B} = [[\texttt{fatherOf}, X, Z], [\texttt{fatherOf}, Z, Y]] \tag{3.2}$$

is also true in a given world. A rule is called a *grounded* rule if all atoms in the rule are themselves *grounded*, i.e., all variables are replaced with *constants* or entities in a world. A *fact* is a grounded binary predicate. A *clause* is a conjunction of two or more atoms ($\mathcal{C} = (\mathcal{H}_\mathcal{C} \vdash \mathcal{B}_\mathcal{C} = ([\alpha_1, ..., \alpha_n]))$) which can be built using a set of rules.

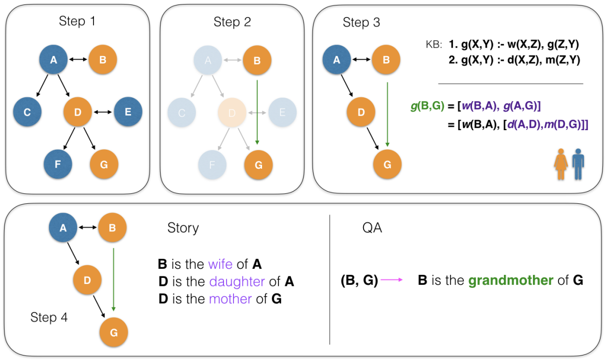

Step 1
A
B
C
D
E
F
G
Step 2
Step 3
KB: 1. g(X,Y) :- w(X,Z), g(Z,Y)
2. g(X,Y) :- d(X,Z), m(Z,Y)
g(B,G) = [w(B,A), g(A,G)]
= [w(B,A), [d(A,D),m(D,G)]]
Step 4
Story
B is the wife of A
D is the daughter of A
D is the mother of G
QA
(B, G) → B is the grandmother of G

Figure 3.1 Data generation pipeline. Step 1: generate a kinship graph. Step 2: sample a target fact. Step 3: Use backward chaining to sample a set of facts. Step 4: Convert sampled facts to a natural language story.

3.2 Overview and construction of CLUTRR

The core idea behind the CLUTRR benchmark suite is the following: Given a natural language story describing a set of kinship relations, the goal is to infer the relationship between two entities, whose relationship is *not* explicitly stated in the story. To generate these stories, we first design a knowledge base (KB) with rules specifying how kinship relations resolve, and we use the following steps to create semi-synthetic stories based on this knowledge base:

Step 1. **Generate** a random kinship graph that satisfies the rules in our KB.

Step 2. **Sample a target fact** (i.e., relation) to predict from the kinship graph.

Step 3. **Apply backward chaining** to sample a set of facts that can prove the target relation (and optionally sample a set of "distracting" or "irrelevant" noise facts).

Step 4. **Convert the sampled facts into a natural language story** through pre-specified text templates and crowd-sourced paraphrasing.

Figure 3.1 provides a high-level overview of this idea, and the following subsections describe the data generation process in detail, as well as the diagnostic flexibility afforded by CLUTRR.

The short stories in CLUTRR are essentially narrativized renderings of a set of logical facts. In the following sections, we describe how we sample the logical facts that make up a story by generating random kinship graphs and using backward chaining to produce logical reasoning chains.

3.2.1 Graph generation

To generate a kinship graph (say, G) underlying a particular story, we first sample a set of gendered[1] entities and kinship relations using a stochastic generation process. This generation process contains a number of tunable parameters—such as the maximum number of children at each node, the probability of an entity being married to another entity, etc.—and is designed to produce a valid, but possibly incomplete "backbone graph". For instance, this backbone graph generation process will specify "parent"/"child" relations between entities but does not add "grandparent" relations. After this initial generation process, we recursively apply the logical rules in R to the backbone graph to produce a final graph G that contains the full set of kinship relations between all the entities. [2]

In the CLUTRR Benchmark, the following kinship relations are used: *son, father, husband, brother, grandson, grandfather, son-in-law, father-in-law, brother-in-law, uncle, nephew, daughter, mother, wife, sister, granddaughter, grandmother, daughter-in-law, mother-in-law, sister-in-law, aunt, niece*.

[1] Kinship and gender roles are oversimplified in our data (compared to the real world) to maintain tractability.

[2] In the context of our data generation process, we distinguish between the knowledge base, R, which contains a finite number of predicates and rules specifying how kinship relations in a family resolve, and a particular kinship graph G, which contains a grounded set of atoms specifying the particular kinship relations that underlie a single story. In other words, R contains the logical rules that govern all the generated stories in CLUTRR, while G contains the grounded facts that underlie a specific story.

$$
\begin{aligned}
[\texttt{grand}, X, Y] &\vdash [[\texttt{child}, X, Z], [\texttt{child}, Z, Y]],\\
[\texttt{grand}, X, Y] &\vdash [[\texttt{SO}, X, Z], [\texttt{grand}, Z, Y]],\\
[\texttt{grand}, X, Y] &\vdash [[\texttt{grand}, X, Z], [\texttt{sibling}, Z, Y]],\\
[\texttt{inv-grand}, X, Y] &\vdash [[\texttt{inv-child}, X, Z], [\texttt{inv-child}, Z, Y]],\\
[\texttt{inv-grand}, X, Y] &\vdash [[\texttt{sibling}, X, Z], [\texttt{inv-grand}, Z, Y]],\\
[\texttt{child}, X, Y] &\vdash [[\texttt{child}, X, Z], [\texttt{sibling}, Z, Y]],\\
[\texttt{child}, X, Y] &\vdash [[\texttt{SO}, X, Z], [\texttt{child}, Z, Y]],\\
[\texttt{inv-child}, X, Y] &\vdash [[\texttt{sibling}, X, Z], [\texttt{inv-child}, Z, Y]],\\
[\texttt{inv-child}, X, Y] &\vdash [[\texttt{child}, X, Z], [\texttt{inv-grand}, Z, Y]],\\
[\texttt{sibling}, X, Y] &\vdash [[\texttt{child}, X, Z], [\texttt{inv-un}, Z, Y]],\\
[\texttt{sibling}, X, Y] &\vdash [[\texttt{inv-child}, X, Z], [\texttt{child}, Z, Y]]\\
[\texttt{sibling}, X, Y] &\vdash [[\texttt{sibling}, X, Z], [\texttt{sibling}, Z, Y]],\\
[\texttt{in-law}, X, Y] &\vdash [[\texttt{child}, X, Z], [\texttt{SO}, Z, Y]],\\
[\texttt{inv-in-law}, X, Y] &\vdash [[\texttt{SO}, X, Z], [\texttt{inv-child}, Z, Y]],\\
[\texttt{un}, X, Y] &\vdash [[\texttt{sibling}, X, Z], [\texttt{child}, Z, Y]],\\
[\texttt{inv-un}, X, Y] &\vdash [[\texttt{inv-child}, X, Z], [\texttt{sibling}, Z, Y]],
\end{aligned}
$$

We used a small, tractable, and logically sound KB of rules as mentioned above. We carefully select this set of deterministic rules to avoid ambiguity in the resolution. We use gender-neutral predicates and resolve the gender of the predicate in the head $\mathcal{H}$ of a clause $\mathcal{C}$ by deducing the gender of the second constant. We have two types of predicates, *vertical* predicates (parent-child relations) and *horizontal* predicates (sibling or significant other). We denote all the vertical predicates by its *child-to-parent* relation and append the prefix `inv-` to the predicates for the corresponding *parent-to-child* relation. For example, `grandfatherOf` is denoted by the gender-neutral predicate $[\texttt{inv-grand}, X, Y]$, where the gender is determined by the gender of Y.

3.2.2 Backward chaining

The resulting graph G provides the *background knowledge* for a specific story, as each edge in this graph can be treated as a grounded predicate (i.e., fact) between two entities. From this graph G, we sample the facts that make up the story, as well as the target fact that we seek to predict: First, we (uniformly) sample a target relation $\mathcal{H}_\mathcal{C}$, which is the fact that we want to predict from the story. Then, from this target relation $\mathcal{H}_\mathcal{C}$, we run a simple variation of the backward chaining [Gallaire and Minker, 1978] algorithm for k iterations starting from $\mathcal{H}_\mathcal{C}$, where at each iteration we uniformly sample a subgoal to resolve and then uniformly sample a KB rule that resolves this subgoal. Crucially, unlike traditional backward chaining, we do not stop the algorithm when a proof is obtained; instead, we run for a fixed number of iterations k in order to sample a set of k facts $\mathcal{B}_\mathcal{C}$ that imply the target relation $\mathcal{H}_\mathcal{C}$.

3.2.3 Adding natural language

So far, we have described the process of generating a conjunctive logical clause $\mathcal{C} = (\mathcal{H}_\mathcal{C} \vdash \mathcal{B}_\mathcal{C})$, where $\mathcal{H}_\mathcal{C} = [\alpha^*]$ is the target fact (i.e., relation) we seek to predict and $\mathcal{B}_\mathcal{C} = [\alpha_1, ..., \alpha_k]$ is the set of supporting facts that imply the target relation. We now describe how we convert this logical representation to natural language through crowdsourcing.

Paraphrasing using Amazon Mechanical Turk

We use Amazon Mechanical Turk (AMT), an online platform for collecting annotations from crowd-workers [3]. The platform supports a mechanism to quickly annotate large amounts of data by paying anonymous workers for their effort. In our work, the crowd-workers are shown a set of facts $\mathcal{B}_\mathcal{C}$ corresponding to a story and then they

[3] https://www.mturk.com/

are asked to paraphrase these facts into a narrative. Since workers are given a set of facts $\mathcal{B}_C$ to work from, they are able to combine and split multiple facts across separate sentences and construct diverse narratives (Figure 3.3).

We use ParlAI [Miller et al., 2017] Mturk interface to collect paraphrases from the users. Specifically, given a set of facts, we ask the users to paraphrase the facts into a story. The users (*turkers*) are free to construct any story they like as long as they mention all the entities and all the relations among them. We also provide the head $\mathcal{H}$ of the clause as an *inferred* relation and specifically instruct the users to *not* mention it in the paraphrased story. In order to evaluate the paraphrased stories, we ask the turkers to peer review a story paraphrased by a different turker. Since there are two tasks - paraphrasing a story and rating a story - we choose to pay 0.5$ for each annotation. A sample task description in our MTurk interface is as follows:

> In this task, you will need to write a short, simple story based on a few facts. **It is crucial that the story mentions each of the given facts at least once.** The story does not need to be complicated! It just needs to be grammatical and mention the required facts.
>
> After writing the story, you will be asked to evaluate the quality of a generated story (based on a different set of facts). **It is crucial that you check whether the generated story mentions each of the required facts.**
>
> *Example of good and bad stories: Good Example*
>
> **Facts to Mention**
>
> - John is the father of Sylvia.
> - Sylvia has a brother Patrick.
>
> **Implied Fact**: John is the father of Patrick.
>
> **Written story**

John is the proud father of the lovely Sylvia. Sylvia has a love-hate relationship with her brother Patrick.

Bad Example

Facts to Mention

- Vincent is the son of Tim.
- Martha is the wife of Tim.

Implied Fact : Martha is Vincent's mother.

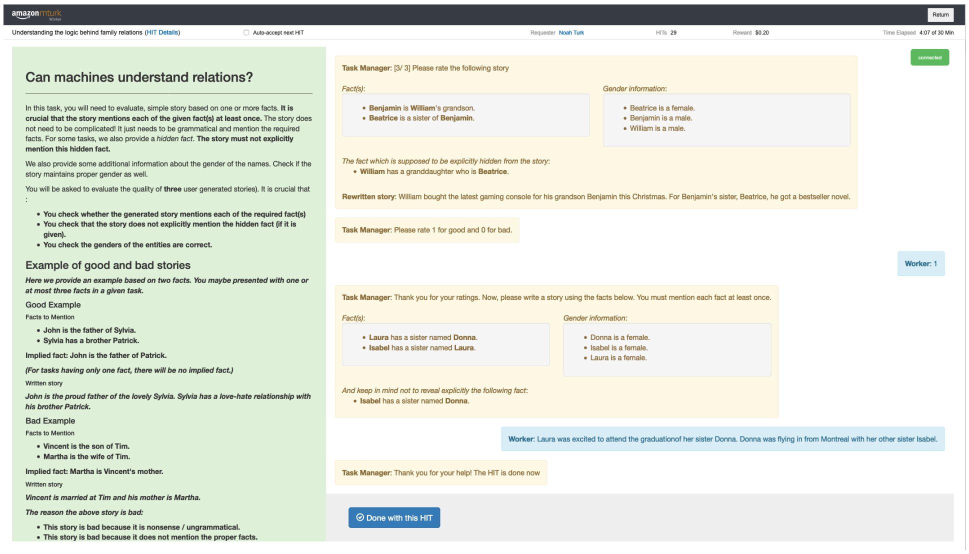
Written story

Vincent is married at Tim and his mother is Martha.

The reason the above story is bad:

- This story is bad because it is nonsense / ungrammatical.
- This story is bad because it does not mention the proper facts.
- This story is bad because it reveals the implied fact.

A sample of the AMT interface is shown in Figure 3.2. To ensure that the turkers are providing high-quality annotations without revealing the inferred fact, we also launch another task to ask the turkers to rate three annotations to be either good or bad which are provided by a set of *different* turkers. We pay 0.2$ for each HIT consisting of three reviews. This helped to remove logical and grammatical inconsistencies to a large extent. Based on the reviews, 79% of the collected paraphrases passed the peer-review sanity check where all the reviewers agree on the quality. This subset of the placeholders is used in the benchmark. A sample of programmatically generated dataset for clause length of $k = 2$ to $k = 6$ is provided in the tables 3.3 to 3.7.



Understanding the logic behind family relations (HIT Details) □ Auto-accept next HIT Requester Noah Turk HITs 29 Reward $0.20 Time Elapsed 4:07 of 30 Min

Can machines understand relations?

In this task, you will need to evaluate, simple story based on one or more facts. **It is crucial that the story mentions each of the given fact(s) at least once.** The story does not need to be complicated! It just needs to be grammatical and mention the required facts. For some tasks, we also provide a *hidden fact*. **The story must not explicitly mention this hidden fact.**

We also provide some additional information about the gender of the names. Check if the story maintains proper gender as well.

You will be asked to evaluate the quality of **three** user generated stories). It is crucial that :

- **You check whether the generated story mentions each of the required fact(s)**
- **You check that the story does not explicitly mention the hidden fact (if it is given).**
- **You check the genders of the entities are correct.**

Example of good and bad stories

Here we provide an example based on two facts. You maybe presented with one or at most three facts in a given task.

Good Example

Facts to Mention

- **John is the father of Sylvia.**
- **Sylvia has a brother Patrick.**

Implied fact: John is the father of Patrick.

(For tasks having only one fact, there will be no implied fact.)

Written story

John is the proud father of the lovely Sylvia. Sylvia has a love-hate relationship with his brother Patrick.

Bad Example

Facts to Mention

- **Vincent is the son of Tim.**
- **Martha is the wife of Tim.**

Implied fact: Martha is Vincent's mother.

Written story

Vincent is married at Tim and his mother is Martha.

The reason the above story is bad:

- **This story is bad because it is nonsense / ungrammatical.**
- **This story is bad because it does not mention the proper facts.**

Task Manager: [3/ 3] Please rate the following story

Fact(s):

- **Benjamin** is **William**'s grandson.
- **Beatrice** is a sister of **Benjamin**.

Gender information:

- Beatrice is a female.
- Benjamin is a male.
- William is a male.

The fact which is supposed to be explicitly hidden from the story:

- **William** has a granddaughter who is **Beatrice**.

Rewritten story: William bought the latest gaming console for his grandson Benjamin this Christmas. For Benjamin's sister, Beatrice, he got a bestseller novel.

Task Manager: Please rate 1 for good and 0 for bad.

Worker: 1

Task Manager: Thank you for your ratings. Now, please write a story using the facts below. You must mention each fact at least once.

Fact(s):

- **Laura** has a sister named **Donna**.
- **Isabel** has a sister named **Laura**.

Gender information:

- Donna is a female.
- Isabel is a female.
- Laura is a female.

And keep in mind not to reveal explicitly the following fact:

- **Isabel** has a sister named **Donna**.

Worker: Laura was excited to attend the graduationof her sister Donna. Donna was flying in from Montreal with her other sister Isabel.

Task Manager: Thank you for your help! The HIT is done now

Figure 3.2 Amazon Mechanical Turker Interface built using ParlAI which was used to collect data as well as peer reviews.

Reusability and composition

One challenge for data collection via AMT is that the number of possible stories generated by CLUTRR grows combinatorially as the number of supporting facts increases, i.e., as $k = |\mathcal{B}_C|$ grows. This combinatorial explosion for large k—combined with the difficulty of maintaining the quality of the crowd-sourced paraphrasing for long stories—makes it infeasible to obtain a large number of paraphrased examples for $k > 3$. To circumvent this issue and increase the flexibility of our benchmark, we reuse and compose AMT paraphrases to generate longer stories. In particular, we collected paraphrases for stories containing $k = 1, 2, 3$ supporting facts and then replaced the entities from these collected stories with placeholders in order to re-use them to generate longer semi-synthetic stories. An example of a story generated by stitching together two shorter paraphrases is provided below:

> [Frank] went to the park with his father, [Brett]. [Frank] called his brother [Boyd] on the phone. He wanted to go out for some beers. [Boyd] went to the baseball game with his son [Jim].
>
> Q: What is [Brett] and [Jim]'s relationship?

Thus, instead of simply collecting paraphrases for a fixed number of stories, we instead obtain a diverse collection of natural language templates that can be programmatically recombined to generate stories with various properties.

3.2.4 AMT Template statistics

Number of Paraphrases			# clauses
	k = 1	1,868	20
	k = 2	1,890	58
	k = 3	2,258	236
	Total	6,016	
Unique Word Count		3,797	
Jaccard Word Overlap	Unigrams	0.201	
	Bigrams	0.0385	

Table 3.1 Statistics of the AMT paraphrases. Jaccard word overlap is calculated within the templates of each individual clause of length k.

At the time of submission, we have collected 6,016 unique paraphrases with an average of 19 paraphrases for every possible logical clause of length $k = 1, 2, 3$. Table 3.1 contains summary statistics of the collected paraphrases. Overall, we found high linguistic diversity in the collected paraphrases. For instance, the average Jaccard overlap in unigrams between pairs paraphrases corresponding to the same logical clause was only 0.201 and only 0.0385 for bigrams.

3.2.5 Human performance

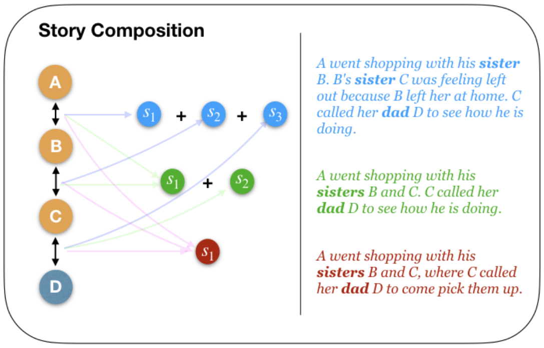
To get a sense of the data quality and difficulty involved in CLUTRR, we asked human annotators to solve the task for random examples of length $k = 2, 3, ..., 6$. (Table 3.2)

Relation Length	Human Performance		Reported Difficulty
	Time Limited	Unlimited Time	
2	0.848	1	1.488 +- 1.25
3	0.773	1	2.41 +- 1.33
4	0.477	1	3.81 +- 1.46
5	0.424	1	3.78 +- 0.96
6	0.406	1	4.46 +- 0.87

Table 3.2 Human performance accuracies on CLUTRR dataset. Humans are provided the Clean-Generalization version of the dataset, and we test on two scenarios: when a human is given limited time to solve the task, and when a human is given unlimited time to solve the task. Regardless of time, our evaluators provide a score of difficulty of individual puzzles.

We perform the evaluation in two scenarios: first a time-limited scenario where we ask AMT Turkers to solve the puzzle in a fixed time. Turkers were provided a maximum time of 30 mins, but they solved the puzzles in an average of 1 minute 23 seconds. Secondly, we use another set of expert evaluators who are given ample time to solve the tasks. Not surprisingly, if a human being is given ample time (experts took an average of 6 minutes per puzzle) and a pen and a paper to aid in the reasoning, they get all the relations correct. However, if an evaluator is short of time, they might miss important details on the relations and perform poorly. Thus, our tasks require *active attention*.

We found that time-constrained AMT annotators performed well (i.e., $> 70\%$) accuracy for $k \leq 3$ but struggled with examples involving longer stories, achieving 40-50% accuracy for $k > 3$. However, trained annotators with unlimited time were able to solve 100% of the examples, highlighting the fact that this task requires attention and involved reasoning, even for humans.



A went shopping with his ***sister*** *B. B's* ***sister*** *C was feeling left out because B left her at home. C called her* ***dad*** *D to see how he is doing.*

A went shopping with his ***sisters*** *B and C. C called her* ***dad*** *D to see how he is doing.*

A went shopping with his ***sisters*** *B and C, where C called her* ***dad*** *D to come pick them up.*

Figure 3.3 Illustration of how a set of facts can split and combined in various ways across sentences.

3.2.6 Representing the question and entities

The AMT paraphrasing approach described above allows us to convert the set of supporting facts $\mathcal{B}_{\mathcal{C}}$ to a natural language story, which can be used to predict the target relation/query $\mathcal{H}_{\mathcal{C}}$. However, instead of converting the target query, $\mathcal{H}_{\mathcal{C}} = [\alpha^*]$, to a natural language question, we instead opt to represent the target query as a K-way classification task, where the two entities in the target relation are provided as input and the goal is to classify the relation that holds between these two entities. This representation avoids the pitfall of revealing information about the answer in the question [Kaushik and Lipton, 2018].

When generating stories, entity names are randomly drawn from a set of 300 common gendered English names. Thus, depending on each run, the entities are never the same. This ensures that the entity names are simply placeholders and uncorrelated from the task.

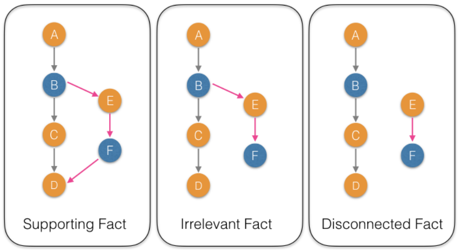


Figure 3.4 Noise generation procedures of CLUTRR.

3.3 Experimental Setups

The modular nature of CLUTRR provides rich diagnostic capabilities for evaluating the robustness and generalization abilities of neural language understanding systems. We highlight some key diagnostic capabilities available via different variations of CLUTRR below. These diagnostic variations correspond to the concrete datasets that we generated in this work, and we describe the results on these datasets in §3.5.

3.3.1 Systematic generalization

Most prominently, CLUTRR allows us to explicitly evaluate a model's ability for generalizing with the property of systematicity. In particular, we rely on the following hold-out procedures to test systematic generalization:

- During training, we hold out a subset of the collected paraphrases, and we only use this held-out subset of paraphrases when generating the test set. Thus, to succeed on CLUTRR, an NLU system must exhibit *linguistic generalization* and be robust to linguistic variation at test time.
- We also hold out a subset of the logical clauses during training (for clauses of length $k > 2$).[4] In other words, during training, the model sees all logical rules but does not

[4] One should not holdout clauses from length $k = 2$ in order to allow models to learn the compositionality of all possible binary predicates.

see all *combinations* of these logical rules. Thus, in addition to linguistic generalization, success on this task also requires *logical generalization*.

- Lastly, as a more extreme form of both logical and linguistic generalization, we consider the setting where the models are trained on stories generated from clauses of length $\leq k$ and evaluated on stories generated from larger clauses of length $> k$. Thus, we explicitly test the ability for models to generalize on examples that require more steps of reasoning that any example they encountered during training.

3.3.2 Robust Reasoning

In addition to evaluating systematic generalization, the modular setup of CLUTRR also allows us to diagnose model robustness by adding *noise facts* to the generated narratives. Due to the controlled semi-synthetic nature of CLUTRR, we are able to provide a precise taxonomy of the kinds of noise facts that can be added (Figure 3.4). In order to structure this taxonomy, it is important to recall that any set of supporting facts $\mathcal{B}_{\mathcal{C}}$ generated by CLUTRR can be interpreted as a path, $p_{\mathcal{C}}$, in the corresponding kinship graph G (Figure 3.1). Based on this interpretation, we view adding noise facts from the perspective of sampling three different types of noise paths, p_n, from the kinship graph G:

- *Irrelevant facts*: We add a path p_n, which has exactly one shared end-point with p_c. In this way, this is a *distractor* path, which contains facts that are connected to one of the entities in the target relation, $\mathcal{H}_{\mathcal{C}}$, but do not provide any information that could be used to help answer the query.
- *Supporting facts*: We add a path p_n, whose two end-points are on the path $p_{\mathcal{C}}$. The facts on this path p_n are noise because they are not needed to answer the query, but they are supporting facts because they can, in principle, be used to construct alternative (longer) reasoning paths that connect the two target entities.
- *Disconnected facts*: We add paths which neither originate nor end in any entity on p_c.

These disconnected facts involve entities and relations that are completely unrelated to the target query.

3.3.3 Generated Datasets

For all experiments, we generated datasets with 10-15k training examples. In many experiments, we report training and testing results on stories with different clause lengths k. (For brevity, we use the phrase "clause length" throughout this section to refer to the value $k = |\mathcal{B}_{\mathcal{C}}|$, i.e., the number of steps of reasoning that are required to predict the target query.) In all cases, the training set contains 5000 train stories per k value, and, during testing, all experiments use 100 test stories per k value. All experiments were run 10 times with different randomly generated stories, and means and standard errors over these 10 runs are reported. As discussed above, during training we holdout 20% of the paraphrases, as well as 10% of the possible logical clauses.

Table 3.3 Snapshot of puzzles in the dataset for k=2

Puzzle	Question	Gender	Answer
Charles's son *Christopher* entered rehab for the ninth time at the age of thirty. *Randolph* had a nephew called *Christopher* who had n't seen for a number of years.	Randolph is the _____ of Charles	Charles:male, Christopher:male, Randolph:male	brother
Randolph and his sister *Sharon* went to the park. *Arthur* went to the baseball. game with his son *Randolph*	Sharon is the _____ of Arthur	Arthur:male, Randolph:male, Sharon:female	daughter
Frank went to the park with his father, *Brett*. *Frank* called his brother *Boyd* on the phone. He wanted to go out for some beers.	Brett is the _____ of Boyd	Boyd:male, Frank:male, Brett:male	father

3.4 Evaluated Models

Our primary baselines are neural language understanding models that take unstructured text as input. We consider bidirectional LSTMs [Hochreiter and Schmidhuber, 1997, Cho et al., 2014] (with and without attention), as well as models that aim to incorporate inductive biases towards relational reasoning: Relation Networks (RN) [Santoro

Table 3.4 Snapshot of puzzles in the dataset for k=3

Puzzle	Question	Gender	Answer
Roger was playing baseball with his sons *Sam* and *Leon*. *Sam* had to take a break though because he needed to call his sister *Robin*.	Leon is the _____ of Robin	Robin:female, Sam:male, Roger:male, Leon:male	brother
Elvira and her daughter *Nancy* went shopping together last Monday and they bought new shoes for *Elvira*'s kids. *Pedro* and his sister *Allison* went to the fair. *Pedro*'s mother, *Nancy*, was out with friends for the day.	Elvira is the _____ of Allison	Allison:female, Pedro:male, Nancy:female, Elvira:female	grandmother
Roger met up with his sister *Nancy* and her daughter *Cynthia* at the mall to go shopping together. *Cynthia*'s brother *Pedro* was going to be the star in the new show.	Pedro is the _____ of Roger	Roger:male, Nancy:female, Cynthia:female, Pedro:male	nephew

Table 3.5 Snapshot of puzzles in the dataset for k=4

Puzzle	Question	Gender	Answer
Celina has been visiting her sister, *Fran* all week. *Fran* is also the daughter of *Bethany*. *Ronald* loves visiting his aunt *Bethany* over the weekends. *Samuel*'s son *Ronald* entered rehab for the ninth time at the age of thirty.	Celina is the _____ of Samuel	Samuel:male, Ronald:male, Bethany:female, Fran:female, Celina:female	niece
Celina adores her daughter *Bethany*. *Bethany* loves her very much, too. *Jackie* called her mother *Bethany* to let her know she will be back home soon. *Thomas* was helping his daughter *Fran* with her homework at home. Afterwards, *Fran* and her sister *Jackie* played Xbox together.	Celina is the _____ of Thomas	Thomas:male, Fran:female, Jackie:female, Bethany:female, Celina:female	daughter
Raquel is *Samuel* 'daughter and they go shopping at least twice a week together. *Kenneth* and her mom, *Theresa*, had a big fight. *Theresa*'s son, *Ronald*, refused to get involved. *Ronald* was having an argument with her sister, *Raquel*.	Samuel is the _____ of Kenneth	Kenneth:male, Theresa:female, Ronald:male, Raquel:female, Samuel:male	father

Table 3.6 Snapshot of puzzles in the dataset for k=5

Puzzle	Question	Gender	Answer
Steven's son is *Bradford*. *Bradford* and his father always go fishing together on Sundays and have a great time together. *Diane* is taking her brother *Brad* out for a late dinner. *Kristin*, *Brad*'s mother, is home with a cold. *Diane*'s father *Elmer*, and his brother *Steven*, all got into the rental car to start the long cross-country roadtrip they had been planning.	Bradford is the _____ of Kristin	Kristin:female, Brad:male, Diane:female, Elmer:male, Steven:male, Bradford:male	nephew
Elmer went on a roadtrip with his youngest child, *Brad*. *Lena* and her sister *Diane* are going to a restaurant for lunch. *Lena*'s brother *Brad* is going to meet them there with his father *Elmer* *Brad* ca n't stand his unfriendly aunt *Lizzie*.	Lizzie is the _____ of Diane	Diane:female, Lena:female, Brad:male, Elmer:male, Lizzie:female	aunt
Ira took his niece *April* fishing Saturday. They caught a couple small fish. *Ronald* was enjoying spending time with his parents, *Damion* and *Claudine*. *Damion*'s other son, *Dennis*, wanted to come visit too. *Dennis* often goes out for lunch with his sister, *April*.	Ira is the _____ of Claudine	Claudine:female, Ronald:male, Damion:male, Dennis:male, April:female, Ira:male	brother

Table 3.7 Snapshot of puzzles in the dataset for k=6

Puzzle	**Question**	**Gender**	**Answer**
Mario wanted to get a good gift for his sister, *Marianne*. *Jean* and her sister *Darlene* were going to a party held by *Jean*'s mom, *Marianne*. *Darlene* invited her brother *Roy* to come, too, but he was too busy. *Teri* and her father, *Mario*, had an argument over the weekend. However, they made up by Monday. *Agnes* wants to make a special meal for her daughter *Teri*'s birthday.	Roy is the _____ of Agnes	Agnes:female, Teri:female, Mario:male, Marianne:female, Jean:female, Darlene:female, Roy:male	nephew
Robert's aunt, *Marianne*, asked *Robert* to mow the lawn for her. *Robert* said he could n't because he had a bad back. *William*'s parents, *Brian* and *Marianne*, threw him a surprise party for his birthday. *Brian*'s daughter *Jean* made a mental note to be out of town for her birthday! *Agnes*'s biggest accomplishment is raising her son *Robert*. *Jean* is looking for a good gift for her sister *Darlene*.	Darlene is the _____ of Agnes	Agnes:female, Robert:male, Marianne:female, William:male, Brian:male, Jean:female, Darlene:female	niece
Sharon and her brother *Mario* went shopping. *Teri*, *Mario*'s daughter, came too. *Agnes*, *Annie*'s mother, is unhappy with *Robert*. She feels her son is cruel to *Annie*'s sister *Teri*, and she wants *Robert* to be nicer. *Robert*'s sister, *Nicole*, participated in the dance contest.	Nicole is the _____ of Sharon	Sharon:female, Mario:male, Teri:female, Annie:female, Agnes:female, Robert:male, Nicole:female	niece

et al., 2017], Relational Recurrent Networks (RMC) [Santoro et al., 2018] and Compositional Memory Attention Network (MAC) [Hudson and Manning, 2018]. We also use the large pre-trained language model, BERT [Devlin et al., 2019b], as well as a modified version of BERT having a trainable LSTM encoder on top of the pretrained BERT embeddings. All of these models (except BERT) were re-implemented in PyTorch 1.0 [Paszke et al., 2017] and adapted to work with the CLUTRR benchmark.

Since the underlying relations in the stories generated by CLUTRR inherently form a graph, we also experiment with a Graph Attention Network (GAT) [Veličković et al., 2018]. Rather than taking the textual stories as input, the GAT baseline receives a structured graph representation of the facts that underlie the story.

Entity and query representations. We use the various baseline models to encode the natural language story (or graph) into a fixed-dimensional embedding. With the exception of the BERT models, we do not use pre-trained word embeddings and learn the word embeddings from scratch using end-to-end backpropagation. An important note, however, is that we perform Cloze-style anonymization [Hermann et al., 2015] of the entities (i.e., names) in the stories, where each entity name is replaced by a *@entity-k* placeholder, which is randomly sampled from a small, fixed pool of placeholder tokens. The embeddings for these placeholders are randomly initialized and fixed during training.

To make a prediction about a target query given a story, we concatenate the embedding of the story (generated by the baseline model) with the embeddings of the two target entities and we feed this concatenated embedding to a 2-layer feed-forward neural network with a softmax prediction layer.

3.4.1 Hyperparameters

For all models, the common hyperparameters used are: Embedding dimension: 100 (except BERT based models), Optimizer: Adam, Learning rate: 0.001, Number of epochs: 100, Number of runs: 10. Specific model-based hyperparameters are given as follows:

- **Bidirectional LSTM** [Hochreiter and Schmidhuber, 1997, Cho et al., 2014]: LSTM hidden dimension: 100, # layers: 2, Classifier MLP hidden dimension: 200
- **Relation Networks** [Santoro et al., 2017]: f_{θ_1} : 256, f_{θ_2}: 64, g_θ : 64
- **Compositional Memory Attention Network (MAC)** [Hudson and Manning, 2018]: # Iterations: 6, `shareQuestion`: True, Dropout - Memory, Read and Write: 0.2
- **Relational Recurrent Networks** [Santoro et al., 2018]: Memory slots: 2, Head size: 192, Number of heads: 4, Number of blocks : 1, forget bias : 1, input bias: 0, gate style: unit, key size: 64, # Attention layers: 3, Dropout: 0
- **BERT** [Devlin et al., 2019b]: Layers : 12, Fixed pretrained embeddings from `bert-base-uncased` using Pytorch HuggingFace BERT repository [5], Word dimension: 768, appended with a two-layer MLP for final prediction.
- **BERT-LSTM**: Same parameters as above, with a two-layer unidirectional LSTM encoder on top of BERT word embeddings.
- **GAT** [Veličković et al., 2018]: Node dimension: 100, Message dimension: 100, Edge dimension: 20, number of rounds: 3

3.5 Results

We evaluate several NLU systems on the proposed CLUTRR benchmark to surface the relative strengths and shortcomings of these models in the context of inductive reasoning and combinatorial generalization.[6] We aim to answer the following key questions:

[5] https://github.com/huggingface/pytorch-pretrained-BERT

[6] Code to reproduce all the results in this section are available at https://github.com/facebookresearch/clutrr/.

(**Q1**) How do state-of-the-art NLU models compare in terms of systematic generalization? Can these models generalize to stories with unseen combinations of logical rules?

(**Q2**) How does the performance of neural language understanding models compare to a graph neural network that has full access to graph structure underlying the stories?

(**Q3**) How robust are these models to the addition of noise facts to a given story?

3.5.1 Systematic Generalization

We begin by using CLUTRR to evaluate the ability of the baseline models to perform systematic generalization (**Q1**). In this setting, we consider two training regimes: in the first regime, we train all models with clauses of length $k = 2, 3$, and in the second regime, we train with clauses of length $k = 2, 3, 4$. We then test the generalization of these models on test clauses of length $k = 2, ..., 10$.

Figure 3.5 illustrates the performance of different models on this generalization task. We observe that the GAT model is able to perform near-perfectly on the held-out logical clauses of length $k = 3$, with the BERT-LSTM being the top-performer among the text-based models but still significantly below the GAT. Not surprisingly, the performance of all models degrades monotonically as we increase the length of the test clauses, which highlights the challenge of "zero-shot" systematic generalization Lake and Baroni [2018], Sodhani et al. [2018]. However, as expected, all models improve on their generalization performance when trained on $k = 2, 3, 4$ rather than just $k = 2, 3$ (Figure 3.5, right). The GAT, in particular, achieves the biggest gain by this expanded training.

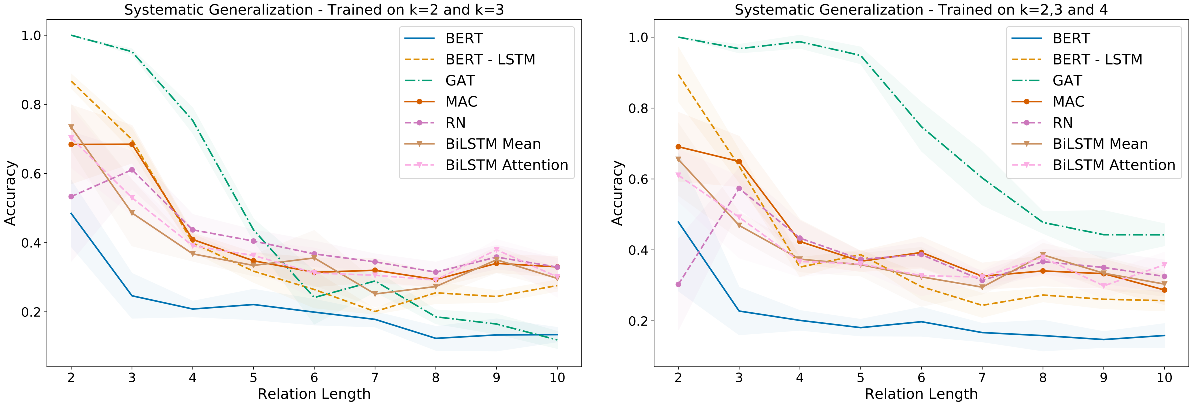


Figure 3.5 Systematic generalization performance of different models when trained on clauses of length $k = 2, 3$ (Left) and $k = 2, 3, 4$ (Right).

3.5.2 The benefit of structure

The empirical results on systematic generalization also provide insight into how the text-based NLU systems compare against the graph-based GAT model that has full access to the logical graph structure underlying the stories (**Q2**). Indeed, the relatively strong performance of the GAT model (Figure 3.5) suggests that the language-based models fail to learn a robust mapping from the natural language narratives to the underlying logical facts.

To further confirm this trend, we ran experiments with modified train and test splits for the text-based models, where the same set of natural language paraphrases were used to construct the narratives in both the train and test splits (Figure 3.6). In this simplified setting, the text-based models must still learn to reason about held-out logical patterns, but the difficulty of parsing the natural language is essentially removed, as the same natural language paraphrases are used during testing and training. We found that the text-based models were competitive with the GAT model in this simplified setting, confirming that the poor performance of the text-based models on the main task is driven by the difficulty of parsing the unseen natural language narratives.

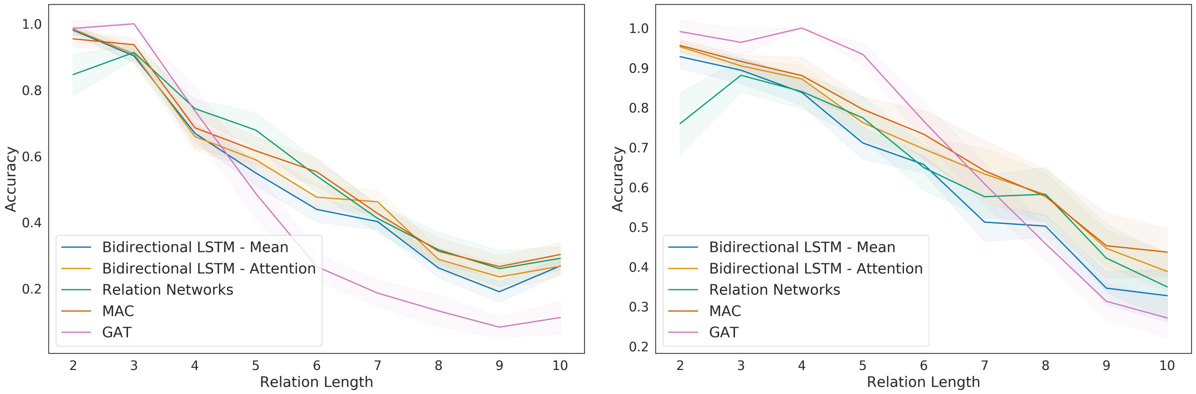


Figure 3.6 Systematic Generalizability of different models on `CLUTRR-Gen` task (having 20% less placeholders and without training and testing placeholder split), when **Left:** trained with $k = 2$ and $k = 3$ and **Right:** trained with $k = 2, 3$ and 4

Models		Unstructured models (no graph)						Structured model (with graph)
Training	Testing	BiLSTM - Attention	BiLSTM - Mean	RN	MAC	BERT	BERT-LSTM	GAT
Clean	Clean	0.58 $_{\pm 0.05}$	0.53 $_{\pm 0.05}$	0.49 $_{\pm 0.06}$	0.63 $_{\pm 0.08}$	0.37 $_{\pm 0.06}$	0.67 $_{\pm 0.03}$	**1.0** $_{\pm 0.0}$
	Supporting	**0.76** $_{\pm 0.02}$	0.64 $_{\pm 0.22}$	0.58 $_{\pm 0.06}$	0.71 $_{\pm 0.07}$	0.28 $_{\pm 0.1}$	0.66 $_{\pm 0.06}$	0.24 $_{\pm 0.2}$
	Irrelevant	0.7 $_{\pm 0.15}$	**0.76** $_{\pm 0.02}$	0.59 $_{\pm 0.06}$	0.69 $_{\pm 0.05}$	0.24 $_{\pm 0.08}$	0.55 $_{\pm 0.03}$	0.51 $_{\pm 0.15}$
	Disconnected	0.49 $_{\pm 0.05}$	0.45 $_{\pm 0.05}$	0.5 $_{\pm 0.06}$	0.59 $_{\pm 0.05}$	0.24 $_{\pm 0.08}$	0.5 $_{\pm 0.06}$	**0.8** $_{\pm 0.17}$
Supporting	Supporting	0.67 $_{\pm 0.06}$	0.66 $_{\pm 0.07}$	0.68 $_{\pm 0.05}$	0.65 $_{\pm 0.04}$	0.32 $_{\pm 0.09}$	0.57 $_{\pm 0.04}$	**0.98** $_{\pm 0.01}$
Irrelevant	Irrelevant	0.51 $_{\pm 0.06}$	0.52 $_{\pm 0.06}$	0.5 $_{\pm 0.04}$	0.56 $_{\pm 0.04}$	0.25 $_{\pm 0.06}$	0.53 $_{\pm 0.06}$	**0.93** $_{\pm 0.01}$
Disconnected	Disconnected	0.57 $_{\pm 0.07}$	0.57 $_{\pm 0.06}$	0.45 $_{\pm 0.11}$	0.4 $_{\pm 0.1}$	0.17 $_{\pm 0.05}$	0.47 $_{\pm 0.06}$	**0.96** $_{\pm 0.01}$
Average		**0.61** $_{\pm 0.08}$	0.59 $_{\pm 0.08}$	0.54 $_{\pm 0.07}$	**0.61** $_{\pm 0.06}$	0.30 $_{\pm 0.07}$	0.56 $_{\pm 0.05}$	**0.77** $_{\pm 0.09}$

Table 3.8 Testing the robustness of the various models when training and testing on stories containing various types of noise facts. The types of noise facts (supporting, irrelevant, and disconnected) are defined in Section .

3.5.3 Robust Reasoning

Finally, we use CLUTRR to systematically evaluate how various baseline neural language understanding systems cope with noise (**Q3**). In all the experiments we provide a combination of $k = 2$ and $k = 3$ length clauses in training and testing, with noise facts being added to the train and/or test set depending on the setting (Table 3.8). We use the different types of noise facts defined in Section 3.3.2..

Overall, we find that the GAT baseline outperforms the unstructured text-based models across most testing scenarios (Table 3.8), which showcases the benefit of a structured feature space for robust reasoning. When training on clean data and testing on noisy data, we observe two interesting trends that highlight the benefits and shortcomings of the various model classes:

1. All the text-based models excluding BERT actually perform better when testing on examples that have *supporting* or *irrelevant* facts added. This suggests that these models actually benefit from having more content related to the entities in the story. Even though this content is not strictly useful or needed for the reasoning task, it may provide some linguistic cues (e.g., about entity genders) that the models exploit. In contrast, the BERT-based models do not benefit from the inclusion of this extra content, which is perhaps due to the fact that they are already built upon a strong language model (e.g., that already adequately captures entity genders.)
2. The GAT model performs poorly when *supporting* facts are added but has no performance drop when *disconnected* facts are added. This suggests that the GAT model is sensitive to changes that introduce cycles in the underlying graph structure but is robust to the addition of noise that is disconnected from the target entities.

Learning from noisy data

Moreover, when we trained on noisy examples, we found that only the GAT model was able to consistently improve its performance (Table 3.8). We notice that the GAT model, having access to the true underlying graph of the puzzles, perform better across different testing scenarios when trained with the noisy data. As the *Supporting facts* contains cycles, it is difficult for GAT to generalize for a dataset with cycles when it is trained on a dataset without cycles. However, when trained with cycles, GAT learns to attend to *all* the paths leading to the correct answer. This effect is disastrous when GAT is tested on *Irrelevant facts* which contains dangling paths as GAT still tries to attend to all the paths. Training on *Irrelevant facts* proved to be most beneficial to GAT, as the

Models		Unstructured models (no graph)						Structured model (with graph)
Training	Testing	BiLSTM - Attention	BiLSTM - Mean	RN	MAC	BERT	BERT-LSTM	GAT
Supporting	Clean	0.38 ±0.04	0.32 ±0.04	0.4 ±0.09	0.45 ±0.03	0.19 ±0.06	0.39 ±0.06	**0.92** ±0.17
	Supporting	0.67 ±0.06	0.66 ±0.07	0.68 ±0.05	0.65 ±0.04	0.32 ±0.09	0.57 ±0.04	**0.98** ±0.01
	Irrelevant	0.44 ±0.03	0.39 ±0.03	**0.51** ±0.08	0.46 ±0.09	0.2 ±0.06	0.36 ±0.05	0.5 ±0.23
	Disconnected	0.31 ±0.21	0.25 ±0.16	0.47 ±0.08	0.41 ±0.06	0.2 ±0.08	0.32 ±0.04	**0.92** ±0.05
Irrelevant	Clean	0.57 ±0.05	0.56 ±0.05	0.46 ±0.13	0.67 ±0.05	0.24 ±0.06	0.46 ±0.08	**0.92** ±0.0
	Supporting	0.38 ±0.22	0.31 ±0.16	0.61 ±0.07	0.61 ±0.04	0.27 ±0.06	0.46 ±0.04	**0.77** ±0.12
	Irrelevant	0.51 ±0.06	0.52 ±0.06	0.5 ±0.04	0.56 ±0.04	0.25 ±0.06	0.53 ±0.06	**0.93** ±0.01
	Disconnected	0.44 ±0.26	0.54 ±0.27	0.55 ±0.05	0.61 ±0.06	0.26 ±0.03	0.45 ±0.08	**0.85** ±0.25
Disconnected	Clean	0.45 ±0.02	0.47 ±0.03	0.53 ±0.09	0.5 ±0.06	0.22 ±0.09	0.44 ±0.05	**0.75** ±0.07
	Supporting	0.47 ±0.03	0.46 ±0.05	0.54 ±0.03	0.58 ±0.06	0.22 ±0.06	0.38 ±0.08	**0.78** ±0.12
	Irrelevant	0.47 ±0.05	0.48 ±0.03	0.52 ±0.04	0.51 ±0.05	0.17 ±0.04	0.38 ±0.05	**0.56** ±0.26
	Disconnected	0.57 ±0.07	0.57 ±0.06	0.45 ±0.11	0.4 ±0.1	0.17 ±0.05	0.47 ±0.06	**0.96** ±0.01
Average		0.47 ±0.08	0.46 ±0.08	0.52 ±0.07	**0.53** ±0.06	0.23 ±0.07	0.43 ±0.05	**0.82** ±0.09

Table 3.9 Testing the robustness of the various models when trained various types of noisy facts and evaluated on other noisy / clean facts.

model now perfectly attends to *only relevant paths*. Since *Disconnected facts* contains disconnected paths, the message passing function of the graph is unable to forward any information from the disjoint cliques, thereby having superior testing scores throughout several scenarios.

Again, these results highlights the performance gap between the unstructured text-based models and GAT for solving the CLUTRR task.

Learning with synthetic placeholders

In order to further understand the effect of language placeholders on robustness, we performed another set of experiments where we use bABI Weston et al. [2015] style simple placeholders (Table 3.10). We observe a marked increase in performance of all NLU models, where they significantly decrease the gap between their performance with that of GAT, even outperforming GAT on various settings. This shows the significance of using paraphrased placeholders in devising the complexity of the dataset.

Models		Unstructured models (no graph)						Structured model (with graph)
Training	Testing	BiLSTM - Attention	BiLSTM - Mean	RN	MAC	BERT	BERT-LSTM	GAT
Supporting	Clean	0.96 ±0.01	**0.97** ±0.01	0.88 ±0.05	0.94 ±0.02	0.48 ±0.08	0.57 ±0.08	0.92 ±0.17
	Supporting	0.96 ±0.03	0.96 ±0.03	0.97 ±0.01	0.97 ±0.01	0.75 ±0.07	0.88 ±0.05	**0.98** ±0.01
	Irrelevant	0.92 ±0.02	**0.93** ±0.01	0.9 ±0.03	0.91 ±0.01	0.56 ±0.04	0.54 ±0.06	0.5 ±0.23
	Disconnected	0.8 ±0.04	0.83 ±0.04	0.76 ±0.08	0.86 ±0.04	0.27 ±0.06	0.42 ±0.08	**0.92** ±0.05
Irrelevant	Clean	0.63 ±0.02	0.61 ±0.07	0.85 ±0.09	0.8 ±0.07	0.53 ±0.09	0.44 ±0.06	**0.92** ±0.0
	Supporting	0.66 ±0.03	0.64 ±0.04	0.69 ±0.06	0.76 ±0.06	0.42 ±0.08	0.43 ±0.08	**0.77** ±0.12
	Irrelevant	0.89 ±0.04	0.86 ±0.1	0.74 ±0.11	0.78 ±0.06	0.61 ±0.1	0.83 ±0.06	**0.93** ±0.01
	Disconnected	0.64 ±0.02	0.62 ±0.05	0.72 ±0.05	0.73 ±0.04	0.41 ±0.04	0.61 ±0.05	**0.85** ±0.25
Disconnected	Clean	0.9 ±0.05	0.82 ±0.12	**0.94** ±0.02	0.93 ±0.04	0.68 ±0.07	0.64 ±0.02	0.75 ±0.07
	Supporting	0.87 ±0.04	0.82 ±0.05	0.85 ±0.03	**0.88** ±0.04	0.54 ±0.08	0.5 ±0.05	0.78 ±0.12
	Irrelevant	**0.87** ±0.03	0.85 ±0.03	0.83 ±0.03	0.87 ±0.02	0.59 ±0.09	0.58 ±0.09	0.56 ±0.26
	Disconnected	0.91 ±0.04	0.91 ±0.03	0.8 ±0.17	0.71 ±0.11	0.49 ±0.1	0.79 ±0.1	**0.96** ±0.01
Average		0.83 ±0.08	0.82 ±0.08	0.83 ±0.07	**0.84** ±0.06	0.58 ±0.07	0.60 ±0.05	**0.82** ±0.09

Table 3.10 Testing the robustness on toy placeholders of the various models when trained various types of noisy facts and evaluated on other noisy / clean facts.

3.6 Related Work

To design the CLUTRR dataset, we draw inspiration from the classic work on inductive logic programming (ILP), a long line of reading comprehension benchmarks in NLP, as well as work combining language and knowledge graphs.

3.6.1 Reading comprehension benchmarks

Many datasets have been proposed to test the reading comprehension ability of NLP systems. This includes the SQuAD [Rajpurkar et al., 2016], NewsQA [Trischler et al., 2016], and MCTest [Richardson et al., 2013] benchmarks that focus on factual questions; the SNLI [Bowman et al., 2015] and MultiNLI [Williams et al., 2018c] benchmarks for sentence understanding; and the bABI tasks [Weston et al., 2015], to name a few. Our primary contribution to this line of work is the development of a carefully designed *diagnostic* benchmark to evaluate model robustness and systematic generalization in the context of NLU.

3.6.2 Systematic generalization

A growing body of literature has demonstrated that NLU models tend to exploit statistical artifacts in datasets and lack true generalization capabilities [Jia and Liang, 2017, Gururangan et al., 2018, Kaushik and Lipton, 2018, Lake and Baroni, 2018]. These critical examinations have dovetailed with similar studies on visual question answering [Agrawal et al., 2016, Bahdanau et al., 2019, Johnson et al., 2017]. CLUTRR, contributes to this growing area by introducing a principled and flexible benchmark to evaluate systematic generalization in the context of language understanding—with our notion of systematic generalization being grounded in classic work on inductive logic programming (ILP) Quinlan [1990].

3.6.3 Question-answering with knowledge graphs

Our work is also related to the domain of question answering and reasoning in knowledge graphs [Das et al., 2018, Xiong et al., 2018, Hamilton et al., 2018, Wang et al., 2018, Xiong et al., 2017, Welbl et al., 2018, Kartsaklis et al., 2018], where either the model is provided with a knowledge graph to perform inference over or where the model must infer a knowledge graph from the text itself. However, unlike previous benchmarks in this domain—which are generally *transductive* and focus on leveraging and extracting knowledge graphs as a source of background knowledge about a fixed set of entities—CLUTRR requires *inductive logical reasoning*, where every example requires reasoning over a new set of previously unseen entities.

3.7 Discussion

In this paper we introduced the CLUTRR benchmark suite to test the systematic generalization and inductive reasoning capabilities of NLU systems. We demonstrated the diagnostic capabilities of CLUTRR and found that existing NLU systems exhibit relatively poor robustness and systematic generalization capabilities—especially when compared to a graph neural network that works directly with symbolic input. Concretely, using CLUTRR we were able to make the following key insights about the reasoning capability of modern neural networks:

- **Neural language models are unable to reason when tested with systematicity.** We saw in §3.5.1 that the performance of all NLU models drastically degrade when we test on instances which require systematicity - the knowledge of combination of existing parts - to solve the task. While all models had access to all possible rules (by ingesting a combination of relations in the training data), all models are notably worse when tested with longer chain of reasoning than the ones trained upon. This shortcoming could be due to overly associating to certain patterns seen during training, or learning to solve the task by taking shortcuts - associating some combination of tokens for certain relations [Gururangan et al., 2018].

- **Models are not robust in their language understanding.** When evaluated with enabling (supporting) and distractor information (noise), we observe models to display conflicting results. While supporting information is indeed useful for certain classes of models (§3.5.3), irrelevant and distracting information also seems to aide in the reasoning process, which is not a systematic behaviour. Furthermore, when trained with noise, majority of the NLU models are unable to discern between the correct and the incorrect information. These results indicate a potential surface form realization issue.

- **The key hurdle behind systematic generalization is the natural language itself.** Finally, we observe overwhelmingly that when a model which is only provided a graph, stripped of the natural language layer, the model is able to reason with surprising ability. The graph model, GAT, does not have to extract the relevant information from a given free-form text. This makes it easier for the model to generalize more effectively, even in the scenarios when the model is tasked to learn from distractor (noisy) information.

These results highlight the gap that remains between machine reasoning models that work with unstructured text and models that are given access to more structured input. It appears the key hindrance for a neural model for effective generalization and reasoning is the access to proper surface forms. These results raises questions on the syntax processing capabilities of NLU models, and call for more in-depth investigation on the same. In fact, in the following chapters of this thesis, I will discuss my works on further studying the notions of syntax encoding in NLU models using the tool of systematicity.

3.8 Follow-up findings in the community

Our work inspired the community to explore the limits of reasoning capabilities in Transformer models. Gontier et al. [2020] explore the limits of soft theorem-proving using Transformers by leveraging the CLUTRR dataset. They observe similar length generalization issues in theorem proving by generation, although they find Transformers can improve their generalization performance when trained with longer, more exhaustive theorem proofs. Clark et al. [2020b] utilize the data generation pipeline of CLUTRR to develop a mechanism to perform soft theorem-proving on explicitly provided synthetic rules (unlike our work, where we rely on *implicit* rules) expressed in natural language following a question answering setup. Similar to our work, they also find *intrapolation* and *extrapolation* issues of Transformer based models. However in in-distribution setups the models are fairly robust in their reasoning capabilities, leading Clark et al. [2020b] to conclude that Transformers are able to "*learn to reason*". Zhang et al. [2022a] attempt to clear this paradox by concluding that while Transformers show impressive in-distribution performance, the result is not sufficient to claim the model has learned to reason. They observe that Transformers learn to memorize and exploit statistical patterns rather than reasoning, thus further validating the results of our work in this chapter.

Our work also inspired the development of datasets and benchmarks to explore systematicity in language reasoning. Goodwin et al. [2020] develop a dataset grounded in first order logic to inspect the systematicity of NLU models in the domain of natural language inference, and find state-of-the-art NLU models do not generalize systematically despite projecting overall high performance. Minervini et al. [2020] develop neuro-symbolic methods to solve the base version of the CLUTRR dataset, and observe models having exposure to the symbolic rules offer much better results in systematic generalization. Tian et al. [2021] also construct a diagnostic dataset in natural language inference which is grounded in first order logic. They also discover weaknesses of popular Transformer model variants on reasoning on natural logic, especially in the similar flavor of generalization tests proposed in our work. Yanaka et al. [2021] develop datasets to test whether Transformer models can parse sentences involving novel combinations of logical expressions, such as quantifiers and negation. They find that Transformers can only generalize to unseen combinations of quantifiers, negations and modifiers in sentences having similar surface forms as in the training data, but not to unseen surface forms. Their result further validates our conclusions that poor generalization often stems from limited encoding of the surface forms of text. Leveraging the findings from our work, Tamari et al. [2022] develop a synthetic data generation framework to repurpose the bAbI dataset [Weston et al., 2015], and find compositional generalization is still a hard problem for Transformer models to solve. Fei et al. [2022] develop a similar framework following our work to generate multi-hop questions that contain key entities in multi-hop reasoning chains to ensure the complexity and quality of the task.

Chapter 4

Quantifying syntactic generalization using word order

Of late, large scale pre-trained Transformer-based [Vaswani et al., 2017] models—such as RoBERTa [Liu et al., 2019b], BART [Lewis et al., 2020b], and GPT-2 and -3 [Radford et al., 2019b, Brown et al., 2020]—have exceeded recurrent neural networks' performance on many NLU tasks [Wang et al., 2018, 2019]. Several papers have even suggested that Transformers pretrained on a language modeling (LM) objective can capture syntactic information [Hewitt and Manning, 2019, Jawahar et al., 2019a, Warstadt and Bowman, 2020, Wu et al., 2020], with their self-attention layers being capable of surprisingly effective learning Rogers et al. [2020]. In the preceeding chapter, we observed that NLU models, including BERT, are unable to reason systematicity, primarily due to their lack of understanding the surface forms of the given task. Thus, in this chapter, we question the claim that state-of-the-art NLU models "know syntax".

Since there are many ways to investigate "syntax", we must be clear on what we mean by the term. Knowing the syntax of a sentence means being sensitive to the *order of the words* in that sentence (among other things). Humans are sensitive to word order, so clearly, "language is not merely a bag of words" [Harris, 1954, p.156]. Moreover, it is easier for us to identify or recall words presented in canonical orders than in disordered, ungrammatical sentences; this phenomenon is called the *"sentence superiority effect"* (Cattell 1886, Scheerer 1981, Toyota 2001, Baddeley et al. 2009, Snell and Grainger 2017, 2019, Wen et al. 2019, i.a.). This effect also finds some neurobiological support from work showing ordered text activates portions of the temporal lobe more than unordered word lists [Bemis and Pylkkänen, 2013, Pylkkänen et al., 2014]. In our estimation then, if one wants to claim that a model "knows syntax", then they should minimally show that the model is sensitive to word order (at least for e.g. English or Mandarin Chinese).

Generally, knowing the syntax of a sentence is taken to be a prerequisite for understanding what that sentence means [Heim and Kratzer, 1998]. Models should have to know the syntax first then, if performing any particular NLU task that genuinely requires a humanlike understanding of meaning (cf. Bender and Koller 2020). Thus, if our models are as good at NLU as our current evaluation methods suggest, we should expect them to be sensitive to word order. In this chapter, I discuss our paper Sinha et al. [2021b] where we use a suite of permutation metrics to find the models are not sensitive to word order.

We focus here on textual entailment, one of the hallmark tasks used to measure how well models understand language [Condoravdi et al., 2003, Giampiccolo et al., 2007a]. This task, often also called Natural Language Inference (NLI; Bowman et al. 2015, i.a.), typically consists of two sentences: a premise and a hypothesis. The objective is to predict whether the premise entails the hypothesis, contradicts it, or is neutral with respect to it. We find rampant word order insensitivity in purportedly high performing NLI models. For nearly all premise-hypothesis pairs, **there are many permuted examples that fool the models** into providing the correct prediction. In case of MNLI, for example, the current state-of-the-art of 90.5% can be increased to **98.7**% merely by permuting the word order of test set examples. We even find drastically increased cross-dataset

Premise	**Hypothesis**	**Predicted Label**
Boats in daily use lie within feet of the fashionable bars and restaurants.	There are boats close to bars and restaurants.	E
restaurants and use feet of fashionable lie the in Boats within bars daily .	bars restaurants are There and to close boats .	E
He and his associates weren't operating at the level of metaphor.	He and his associates were operating at the level of the metaphor.	C
his at and metaphor the of were He operating associates n't level .	his the and metaphor level the were He at associates operating of .	C

Table 4.1 Examples from the MNLI Matched development set. Both the original example and the permuted one elicit the same classification label (entailment and contradiction respectively) from RoBERTa (large). A simple demo is provided in an associated Google Colab notebook.

generalization when we reorder words. This is not just a matter of chance—we show that the model output probabilities are significantly different from uniform. A sample of the model outputs with permuted examples is shown in Table 4.1.

We verify our findings with three popular English NLI datasets—SNLI [Bowman et al., 2015], MultiNLI [Williams et al., 2018c] and ANLI [Nie et al., 2020])—and one Chinese one, OCNLI Hu et al. [2020a]. It is thus less likely that our findings result from some quirk of English or a particular tokenization strategy. We also observe the effect for various transformer architectures pre-trained on language modeling (RoBERTa [Liu et al., 2019b], BART [Lewis et al., 2020b], DistilBERT [Sanh et al., 2020]), and non-transformers, including a ConvNet [Zhao et al., 2015], an InferSent model [Conneau et al., 2017], and a BiLSTM [Collobert and Weston, 2008].

Thus, in this chapter I discuss our contributions in Sinha et al. [2021b], which are as follows: (i) we propose a suite of metrics (*Permutation Acceptance*) for measuring model insensitivity to word order (§4.2), (ii) we construct multiple permuted test datasets for measuring NLI model performance at a large scale (§4.4), (iii) we show that NLI models focus on words more than word order, but can partially reconstruct syntactic information from words alone (§4.5.1), (iv) we show the problem persists on out-of-domain data, (v) we show that humans struggle with UnNatural Language Inference, underscoring the non-humanlikeness of SOTA models (§4.5.2), (vi) finally, we explore a simple maximum entropy-based method (§4.5.3) to encourage models not to accept permuted examples.

4.1 Technical Background

In this chapter, we investigate the task of Natural Language Inference (NLI), also known as Recognizing Textual Entailment (RTE). NLI task consists of inferring the logical relation between two sentences, typically known as the *premise* and the *hypothesis*. The logical relations that can exist between these two sentences can be on of three types: *entailment* if the premise *entails* the hypothesis, *contradiction* if it is the opposite, and *neutral* if the sentences have non overlapping meaning. Historically, this logical relation formulation is derived from *natural logic* [MacCartney and Manning, 2007], which consisted of seven set-theoretic relations between any given pair of sentences. We use the formulation prescribed by Bowman et al. [2015], which is the simplified formulation consisting of the three standard relations.

Linguists generally take syntactic structure to be necessary for humans to know what sentences mean. Many also find the NLI task to a very promising approximation of human natural language understanding, in part because it is rooted in the tradition of logical entailment. In the spirit of propositional logic, sentence meaning is taken to be truth-conditional [Frege, 1948, Montague, 1970, Chierchia and McConnell-Ginet, 1990, Heim and Kratzer, 1998]. That is to say that understanding a sentence is equivalent to knowing the actual conditions of the world under which the sentences would be (judged) true [Wittgenstein, 1922]. If grammatical sentences are required for sentential inference, as per a truth conditional approach [Montague, 1970], then permuted sentences should be meaningless. Put another way, the meanings of highly permuted sentences (if they exist) are not propositions, and thus those sentences don't have truth conditions. Only from their truth conditions of sentences can we tell if a sentence entails another. In short, the textual entailment task is technically undefined in our "unnatural" setting.

Since existing definitions don't immediately extend to unnatural word orders, we outline several hypothetical *systematic* ways that a model might perform, had it been sensitive to word order. We hypothesize two models that operate on the first principles of NLI, and one that doesn't. In the first case, Model A deems permuted sentences meaningless (devoid of truth values), as formal semantic theories of human language would predict. Thus, it assigns *"neutral"* to every permuted example. Next, Model B does not deem permuted sentences meaningless, and attempts to understand them. Humans find understanding permuted sentences difficult (see our human evaluations in §4.5.2). Model B could also similarly struggle to decipher the meaning, and just equally sample labels for each example (i.e., assigns equal probability mass to the outcome of each label). Finally, we hypothesize a non-systematic model, Model C, which attempts to treat permuted sentences as though they weren't permuted at all. This model could operate similarly as bag-of-words (BOW), and thus always assign the same label to the permuted examples as it would to the un-permuted examples. If the model failed to assign the original gold label to the original unpermuted examples, it will also fail to assign the original gold label to its permutations; it will never get higher accuracy on permuted examples than on unpermuted ones.

We find in our experiments that the state-of-the-art Transformer-based NLI models (as well as pre-Transformer class of models) do not perform like any of the above hypothetical models. They perform closest to Model C, but are, in some cases, actually able to achieve *higher* accuracy on permuted examples. To better quantitatively describe this behaviour, we introduce our suite of **Permutation Acceptance** metrics that enable us to quantify how accepting models are of permuted sentences.

4.2 Experimental Setup

4.2.1 Constructing the permuted dataset.

For a given dataset D having splits D_{train} and D_{test}, we first train an NLI model M on D_{train} to achieve comparable accuracy to what was reported in the original papers. We then construct a randomized version of D_{test}, which we term as $\hat{D}_{\text{test}}$ such that: for each example $(p_i, h_i, y_i) \in D_{\text{test}}$ (where p_i and h_i are the premise and hypothesis sentences of the example respectively and y_i is the gold label), we use a permutation operator $\mathcal{F}$ that returns a list $(\hat{P}_i, \hat{H}_i)$ of q permuted sentences ($\hat{p}_i$ and $\hat{h}_i$), where q is a hyperparameter. $\mathcal{F}$ essentially permutes all positions of the words in a given sentence (i.e., either in premise or hypothesis) with the restriction that *no words maintain their original position*. In our initial setting, we do not explicitly control the placement of the words relative to their original neighbors, but we analyze clumping effects in §4.4. $\hat{D}_{\text{test}}$ now consists of $|D_{\text{test}}| \times q$ examples, with q different permutations of hypothesis and premise for each original test example pair. If a sentence S (e.g., h_i) contains w words, then the total number of available permutations of S are $(w-1)!$, thus making the output of $\mathcal{F}$ a list of $\binom{(w-1)!}{q}$ permutations in this case. For us, the space of possible outputs is larger, since we permute p_i and h_i separately (and ignore examples for which any $|S| \leq 5$).

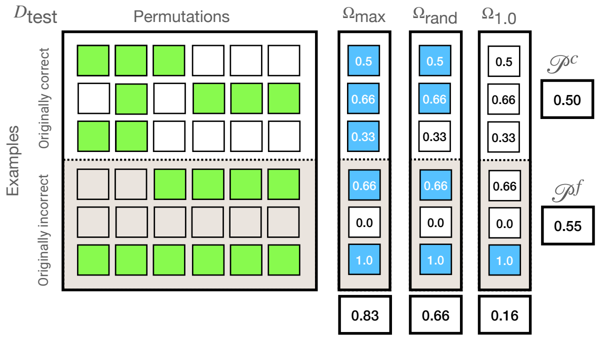


Figure 4.1 Graphical representation of the Permutation Acceptance class of metrics. Given a sample test set D_{test} with six examples, three of which originally predicted correctly (model predicts gold label), three incorrectly (model fails to predict gold label), with $n = 6$ permutations, Ω_{max},Ω_{rand}, $\Omega_{1.0}$, $\mathcal{P}^c$ and $\mathcal{P}^f$ are provided. Green boxes indicate permutations accepted by the model. Blue boxes mark examples that crossed each threshold and were used to compute the corresponding metric.

4.2.2 Defining Permutation Acceptance.

The choice of q naturally allows us to analyze a statistical view of the predictability of a model on the permuted sentences. To that end, we define the following notational conventions. Let $\mathcal{A}$ be the original accuracy of a given model M on a dataset D, and c be the number of examples in a dataset which are marked as correct according to the standard formulation of accuracy for the original dataset (i.e., they are assigned the ground truth label). Typically $\mathcal{A}$ is given by $\frac{c}{|D_{test}|}$ or $\frac{c}{|D_{dev}|}$.

Let $\Pr_M(\hat{P}_i, \hat{H}_i)_{\text{cor}}$ then be the percentage of q permutations of an example (p_i, h_i) assigned the ground truth label y_i by M:

$$\Pr_M(\hat{P}_i, \hat{H}_i)_{\text{cor}} = \frac{1}{q} \sum_{(\hat{p}_j \in \hat{P}_i, \hat{h}_j \in \hat{H}_i)} ((M(\hat{p}_j, \hat{h}_j) = y_i) \to 1) \tag{4.1}$$

To get an overall summary score, we let Ω_x be the percentage of examples $(p_i, h_i) \in D_{\text{test}}$ for which $\Pr_M(\hat{P}_i, \hat{H}_i)_{\text{cor}}$ exceeds a predetermined threshold $0 < x < 1$. Concretely, a given example will count as correct according to Ω_x if more than x percent of its permutations ($\hat{P}_i$ and $\hat{H}_i$) are assigned y_i by the model M. Mathematically,

$$\Omega_x = \frac{1}{\mid D_{test} \mid} \sum_{(p_i,h_i) \in D_{test}} ((\Pr_M(\hat{P}_i, \hat{H}_i)_{\text{cor}} > x) \rightarrow 1). \tag{4.2}$$

There are two specific cases of Ω_{x} that we are most interested in. First, we define Ω_{max} or the **Maximum Accuracy**, where $x = 1/|D_{\text{test}}|$. In short, Ω_{max} gives the percentage of examples $(p_i, h_i) \in D_{\text{test}}$ for which there is *at least one* permutation $(\hat{p_j}, \hat{h_j})$ that model M assigns the gold label y_i [1]. Second, we define Ω_{rand}, or **Random Baseline Accuracy**, where $x = 1/m$ or chance probability (for balanced m-way classification, where $m = 3$ in NLI). This metric is less stringent than Ω_{max}, as it counts an example if at least *one third* of its permutations are assigned the gold label (hence provides a lower-bound relaxation). See Figure 4.1 for a graphical representation of Ω_{x}.

We also define D^f to be the list of examples originally marked incorrect according to $\mathcal{A}$, but are now deemed correct according Ω_{max}. D^c is the list of examples originally marked correct according to $\mathcal{A}$. Thus, we should expect $D^f < D^c$ for models that have high accuracy. Additionally, we define $\mathcal{P}^c$ and $\mathcal{P}^f$, as the dataset average percentage of permutations which predicted the gold label, when the examples were originally correct (D^c) and when the examples were originally incorrect (D^f) as per $\mathcal{A}$ (hence, flipped) respectively.

$$\mathcal{P}^c = \frac{1}{|D^c|} \sum_{i=0}^{|D^c|} M(\hat{P}_i, \hat{H}_i)_{\text{cor}} \tag{4.3}$$

P^f is defined similarly by replacing D^c by D^f. Note that for a classic BOW model, $\mathcal{P}^c = 100$ and $\mathcal{P}^f = 0$, because it would rely on the words alone (not their order) to make its classification decision. Since permuting removes no words, BOW models

[1] Theoretically, $\Omega_{\text{max}} \rightarrow 1$ if the number of permutations q is large. Thus, in our experiments we set $q = 100$.

should come to the same decisions for permuted examples as for the originals.

4.3 Evaluated Models

We run our experiments on two types of models: **(a)** Transformer-based models and **(b)** Non-Transformer Models. In **(a)**, we investigate the state-of-the-art pre-trained models such as RoBERTa-Large Liu et al. [2019b], BART-Large Lewis et al. [2020b] and DistilBERT Sanh et al. [2020]. For **(b)** we consider several recurrent and convolution based neural networks, such as InferSent Conneau et al. [2017], Bidirectional LSTM Collobert and Weston [2008] and ConvNet Zhao et al. [2015]. We train all models on MNLI, and evaluate on in-distribution (SNLI and MNLI) and out-of-distribution datasets (ANLI). We independently verify results of **(a)** using both our fine-tuned model using HuggingFace Transformers Wolf et al. [2020a] and pre-trained checkpoints from FairSeq Ott et al. [2019] (using PyTorch Model Hub). For **(b)**, we use the InferSent codebase. We sample $q = 100$ permutations for each example in D_{test}, and use 100 seeds for each of those permutations to ensure full reproducibility. We drop examples from test sets where we are unable to compute *all unique* randomizations, typically these are examples with sentences of length of less than 6 tokens. [2]

4.4 Results

4.4.1 Models accept many permuted examples.

We find Ω_{max} is very high for models trained and evaluated on MNLI (in-domain generalization), reaching **98.7%** on MNLI dev. and test sets (in RoBERTa, compared to $\mathcal{A}$ of 90.6% (Table 4.2). Recall, human accuracy is approximately 92% on MNLI dev., Nangia and Bowman 2019). This shows that there exists at least one permutation (usually

[2] Code, data, and model checkpoints are available at https://github.com/facebookresearch/unlu.

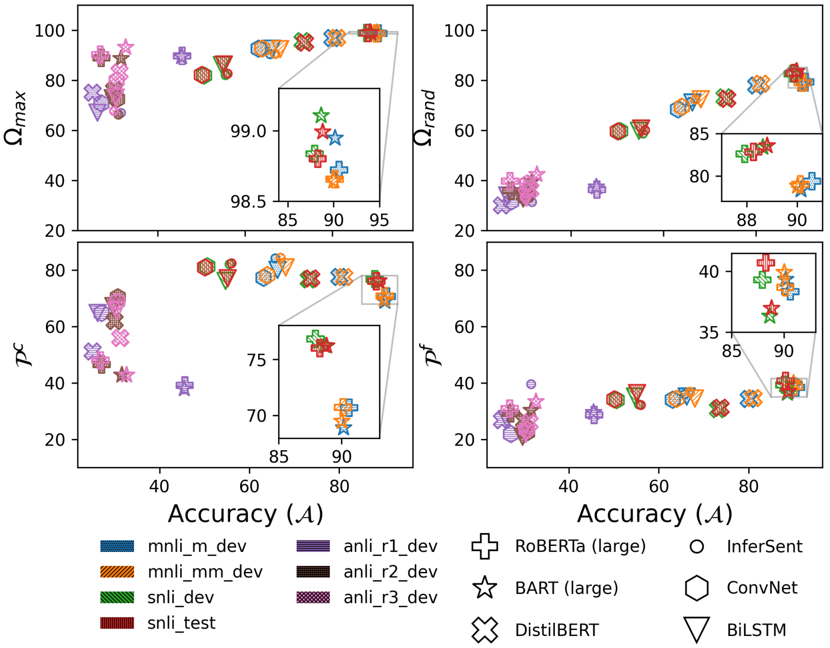


Figure 4.2 Comparison of Ω_{max},Ω_{rand},$\mathcal{P}^c$ and $\mathcal{P}^f$ with the model accuracy $\mathcal{A}$ on multiple datasets, where all models are trained on the MNLI corpus Williams et al. [2018c].

many more) for almost all examples in D_{test} such that model M predicts the gold label. We also observe high Ω_{rand} at 79.4%, showing that there are many examples for which the models outperform even a random baseline in accepting permuted sentences. We provide an example of the behaviour in Table 4.1.

Evaluating out-of-domain generalization with ANLI dataset splits resulted in an Ω_{max} value that is notably higher than $\mathcal{A}$ (89.7% Ω_{max} for RoBERTa compared to 45.6% $\mathcal{A}$). As a consequence, we encounter many *flips*, i.e., examples where the model is unable to predict the gold label, but at least one permutation of that example is able to. However, recall this analysis expects us to know the gold label upfront, so this test can be thought of as running a word-order probe test on the model until the model predicts the gold label (or give up by exhausting our set of q permutations). For out-of-domain generalization, Ω_{rand} decreases considerably (36.4% Ω_{rand} on A1), which means fewer permutations are accepted by the model. Next, recall that a classic bag-of-words model would have $\mathcal{P}^c = 100$ and $\mathcal{P}^f = 0$. No model performs strictly like a classic bag of words although they do perform somewhat BOW-like ($\mathcal{P}^c >> \mathcal{P}^f$ for all test splits, Figure 4.2). We find this BOW-likeness to be higher for certain non-Transformer models, (InferSent) as they exhibit higher $\mathcal{P}^c$ (84.2% for InferSent compared to 70.7% for RoBERTa on MNLI).

Investigating other Ω values

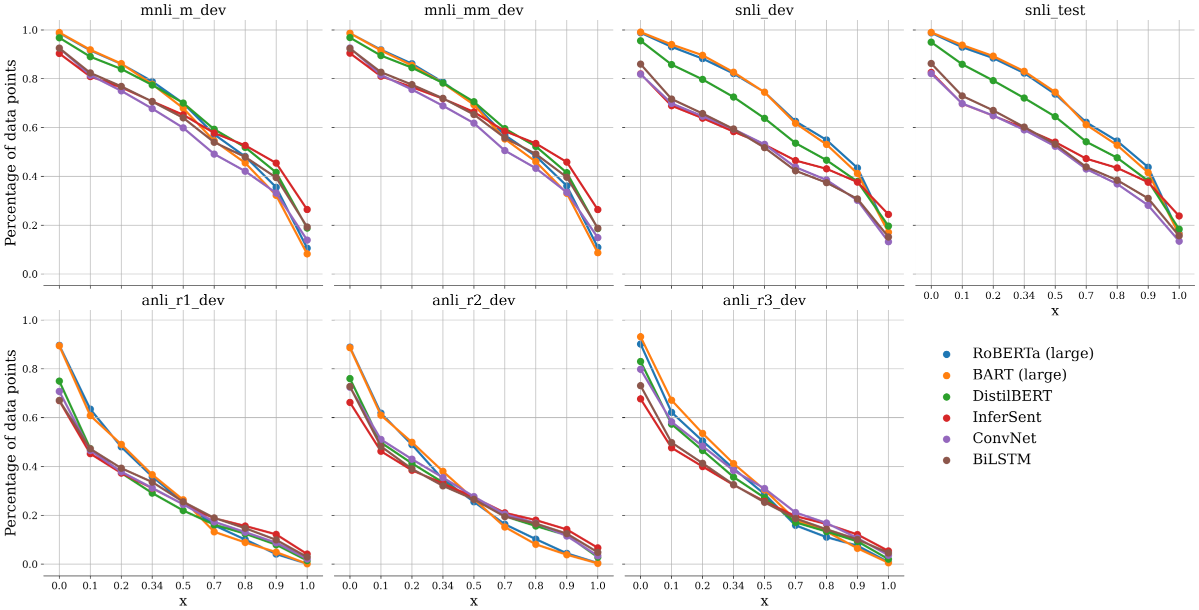


Figure 4.3 Ω_x threshold for all datasets with varying x and computing the percentage of examples that fall within the threshold. The top row consists of in-distribution datasets (MNLI, SNLI) and the bottom row contains out-of-distribution datasets (ANLI)

We defined two variations of Ω_x, Ω_{max} and Ω_{rand}, but theoretically it is possible to define any arbitrary threshold percentage x to evaluate the unnatural language inference mechanisms of different models. In Figure 4.3 we show the effect of different

Model	Eval. Dataset	$\mathcal{A}$	Ω_{max}	$\mathcal{P}^c$	$\mathcal{P}^f$	Ω_{rand}
RoBERTa-Large	MNLI_m_dev	0.906	0.987	0.707	0.383	0.794
	MNLI_mm_dev	0.901	0.987	0.707	0.387	0.790
	SNLI_dev	0.879	0.988	0.768	0.393	0.826
	SNLI_test	0.883	0.988	0.760	0.407	0.828
	A1*	0.456	0.897	0.392	0.286	0.364
	A2*	0.271	0.889	0.465	0.292	0.359
	A3*	0.268	0.902	0.480	0.308	0.397
	Mean	0.652	0.948	0.611	**0.351**	0.623
BART-Large	MNLI_m_dev	0.902	0.989	0.689	0.393	0.784
	MNLI_mm_dev	0.900	0.986	0.695	0.399	0.788
	SNLI_dev	0.886	0.991	0.762	0.363	0.834
	SNLI_test	0.888	0.990	0.762	0.370	0.836
	A1*	0.455	0.894	0.379	0.295	0.374
	A2*	0.316	0.887	0.428	0.303	0.397
	A3*	0.327	0.931	0.428	0.333	0.424
	Mean	**0.668**	**0.953**	0.592	**0.351**	**0.634**
DistilBERT	MNLI_m_dev	0.800	0.968	0.775	0.343	0.779
	MNLI_mm_dev	0.811	0.968	0.775	0.346	0.786
	SNLI_dev	0.732	0.956	0.767	0.307	0.731
	SNLI_test	0.738	0.950	0.770	0.312	0.725
	A1*	0.251	0.750	0.511	0.267	0.300
	A2*	0.300	0.760	0.619	0.265	0.343
	A3*	0.312	0.830	0.559	0.259	0.363
	Mean	0.564	0.883	**0.682**	0.300	0.575
InferSent	MNLI_m_dev	0.658	0.904	0.842	0.359	0.712
	MNLI_mm_dev	0.669	0.905	0.844	0.368	0.723
	SNLI_dev	0.556	0.820	0.821	0.323	0.587
	SNLI_test	0.560	0.826	0.824	0.321	0.600
	A1*	0.316	0.669	0.425	0.395	0.313
	A2*	0.310	0.662	0.689	0.249	0.330
	A3*	0.300	0.677	0.675	0.236	0.332
	Mean	**0.481**	0.780	0.731	**0.322**	0.514
ConvNet	MNLI_m_dev	0.631	0.926	0.773	0.340	0.684
	MNLI_mm_dev	0.640	0.926	0.782	0.343	0.694
	SNLI_dev	0.506	0.819	0.813	0.339	0.597
	SNLI_test	0.501	0.821	0.809	0.341	0.596
	A1*	0.271	0.708	0.648	0.218	0.316
	A2*	0.307	0.725	0.703	0.224	0.356
	A3*	0.306	0.798	0.688	0.234	0.388
	Mean	0.452	**0.817**	**0.745**	0.291	0.519
BiLSTM	MNLI_m_dev	0.662	0.925	0.800	0.351	0.711
	MNLI_mm_dev	0.681	0.924	0.809	0.344	0.724
	SNLI_dev	0.547	0.860	0.762	0.351	0.598
	SNLI_test	0.552	0.862	0.771	0.363	0.607
	A1*	0.262	0.671	0.648	0.271	0.340
	A2*	0.297	0.728	0.672	0.209	0.328
	A3*	0.304	0.731	0.656	0.219	0.331
	Mean	0.472	0.814	0.731	0.301	**0.520**

Table 4.2 Statistics for Transformer-based models trained on MNLI corpus Williams et al. [2018c]. The highest values are bolded (**red** indicates the model most insensitive to permutation) per metric and per model class (Transformers and non-Transformers). A1*, A2* and A3* refer to the ANLI dev. sets [Nie et al., 2020].

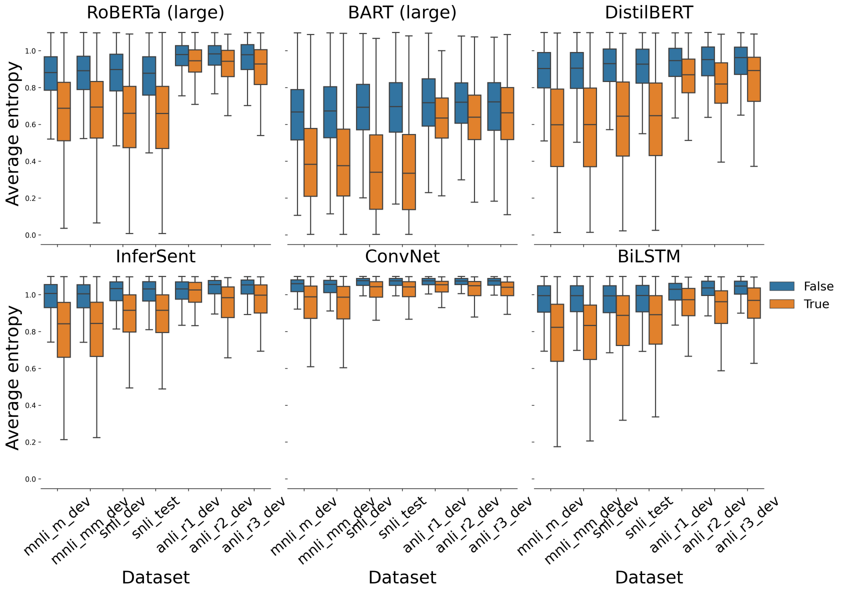


Figure 4.4 Average entropy of model confidences on permutations that yielded the correct results for Transformer-based models (top) and Non-Transformer-based models (bottom). Results are shown for D^c (orange) and D^f (blue). The boxes show the quartiles of the entropy distributions.

thresholds, including Ω_{max} where $x = 1/|D_{\text{test}|}$ and Ω_{rand} where $x = 0.34$. We observe for in-distribution datasets (top row, MNLI and SNLI splits), in the extreme setting when $x = 1.0$, there are more than 10% of examples available, and more than 25% in case of InferSent and DistilBERT. For out-of-distribution datasets (bottom row, ANLI splits) we observe a much lower trend, suggesting generalization itself is the bottleneck in permuted sentence understanding.

4.4.2 Models are very confident.

The phenomenon we observe would be of less concern if the correct label prediction was just an outcome of chance, which could occur when the entropy of the log probabilities of the model output is high (suggesting uniform probabilities on entailment, neutral and contradiction labels, recall Model B from §4.1). We first investigate the model probabilities for the Transformer-based models on the permutations that lead to the correct answer in Figure 4.4. We find overwhelming evidence that model confidences on in-distribution datasets (MNLI, SNLI) are highly skewed, resulting in low entropy, and it varies among different model types. BART proves to be the most skewed Transformer-based model. This skewness is not a property of model capacity, as we observe DistilBERT log probabilities to have similar skewness as RoBERTa (large) model, while exhibiting lower $\mathcal{A}$, Ω_{max}, and Ω_{rand}.

For non-Transformers whose accuracy $\mathcal{A}$ is lower, the Ω_{max} achieved by these models are also predictably lower. We observe roughly the same relative performance in the terms of Ω_{max} (Figure 4.2 and Table 4.2) and Average entropy (Figure 4.4). However, while comparing the averaged entropy of the model predictions, it is clear that there is some benefit to being a worse model—non-Transformer models are not as overconfident on randomized sentences as Transformers are. High confidence of Transformer models can be attributed to the *overthinking* phenomenon commonly observed in deep neural networks Kaya et al. [2019] and BERT-based models Zhou et al. [2020].

4.4.3 Similar artifacts in Chinese NLU.

Model	$\mathcal{A}$	Ω_{max}	$\mathcal{P}^c$	$\mathcal{P}^f$	Ω_{rand}
RoBERTa-Large	**0.784**	**0.988**	0.726	**0.339**	**0.773**
InferSent	0.573	0.931	0.771	0.265	0.615
ConvNet	0.407	0.752	**0.808**	0.199	0.426
BiLSTM	0.566	0.963	0.701	0.271	0.611

Table 4.3 Results on evaluation on OCNLI Dev set. All models are trained on OCNLI corpus Hu et al. [2020a]. Bold marks the highest value per metric (**red** shows the model is insensitive to permutation).

We extended the experiments to the Original Chinese NLI dataset [Hu et al., 2020a, OCNLI], and re-used the pre-trained RoBERTa-Large and InferSent (non-Transformer) models on OCNLI. Our findings are similar to the English results (Table 4.3), thereby suggesting that the phenomenon is not just an artifact of English text or tokenization.

4.4.4 Other Results.

We investigated the effect of sentence length (which correlates with number of possible permutations), and hypothesis-only randomization (models exhibit similar phenomenon even when only hypothesis is permuted). In terms of sentence length, we observe that shorter sentences in general have a somewhat higher probability of acceptance for examples which was originally predicted correctly—since shorter sentences have fewer unique permutations (Figure 4.5). However, for the examples which were originally incorrect, the trend is not present.

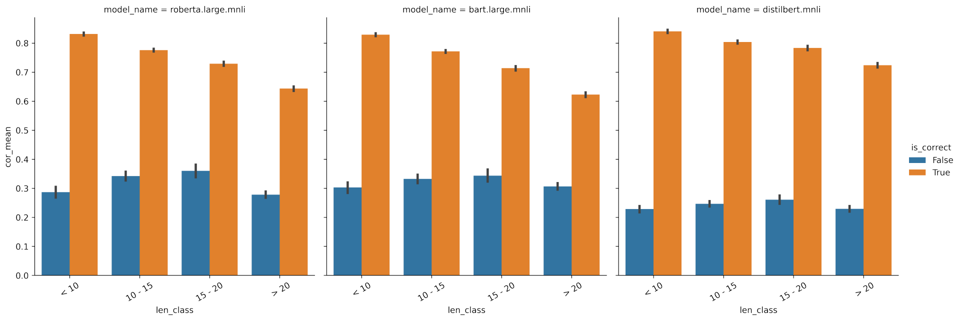


Figure 4.5 Length and Permutation Acceptanceby Transformer-based models.

In recent years, the impact of the hypothesis sentence [Gururangan et al., 2018, Tsuchiya, 2018, Poliak et al., 2018] on NLI classification has been a topic of much interest. As we define in §4.1, logical entailment can only be defined for pairs of propositions. We investigated one effect where we randomize only the hypothesis sentences while keeping the premise intact. Figure 4.6 and Figure 4.7 shows that the $\Omega_{\max}$ value is almost the same for the two schemes; randomizing the hypothesis alone also leads the model to accept many permutations.

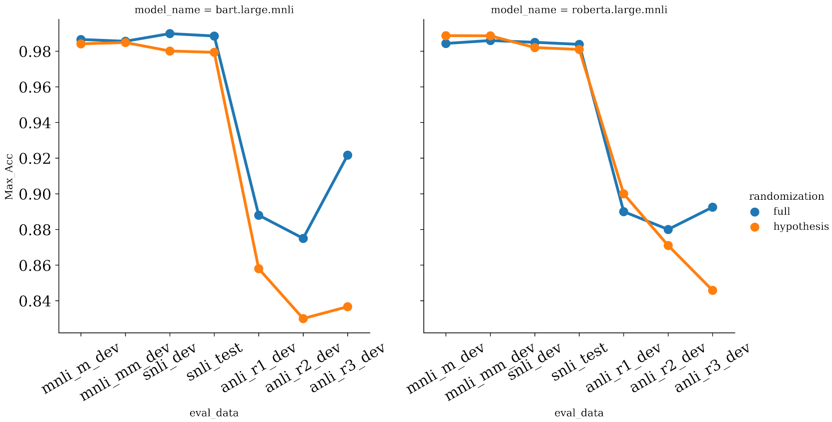


Figure 4.6 Comparing the effect between randomizing both premise and hypothesis and only hypothesis on two Transformer-based models, RoBERTa and BART. Here, we observe the difference of $\Omega_{\max}$ is marginal in in-distribution datasets (SNLI, MNLI), while hypothesis-only randomization is worse for out-of-distribution datasets (ANLI).

4.5 Analysis

4.5.1 Analyzing Syntactic Structure Associated with Tokens

A natural question to ask following our findings: what is it about particular permutations that leads models to accept them? Since the permutation operation is drastic and only rarely preserves local word relations, we first investigate whether there exists a relationship between Permutation Acceptance scores and local word order preservation. Concretely, we compare bi-gram word overlap (BLEU-2) with the percentage of permutations that are deemed correct (Figure 4.8).[3] Although the probability of a permuted sentence to be predicted correctly does appear to track BLEU-2 score (Figure 4.8), the percentage of examples which were assigned the gold label by the Transformer-based models is still higher than we would expect from permutations

[3] We observe, due to our permutation process, the maximum BLEU-3 and BLEU-4 scores are negligibly low (< 0.2 BLEU-3 and < 0.1 BLEU-4), already calling into question the hypothesis that n-grams are the sole explanation for our finding. Because of this, we only compare BLEU-2 scores.

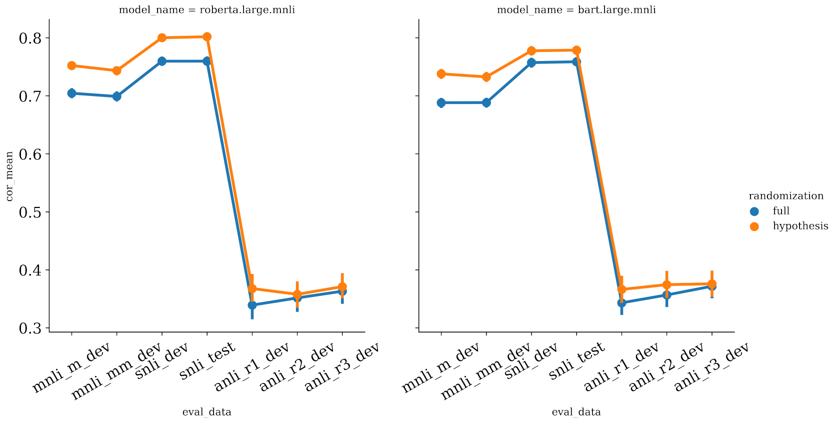


Figure 4.7 Comparing the effect between randomizing both premise and hypothesis and only hypothesis on two Transformer-based models, RoBERTa and BART. In this figure, we compare the mean number of permutations which elicited correct response, and naturally the hypothesis-only randomization causes more percentage of randomizations to be correct.

with lower BLEU-2 (66% for the lowest BLEU-2 range of $0 - 0.15$), suggesting preserved relative word order alone cannot explain the high permutation acceptance rates.

Thus, we find that local order preservation does correlate with Permutation Acceptance, but it doesn't fully explain the high Permutation Acceptance scores. We now further ask whether Ω is related to a more abstract measure of local word relations, i.e., part-of-speech (POS) neighborhood.

Many syntactic formalisms, like Lexical Functional Grammar [Kaplan and Bresnan, 1995, Bresnan et al., 2015, LFG], Head-drive Phrase Structure Grammar [Pollard and Sag, 1994, HPSG] or Lexicalized Tree Adjoining Grammar [Schabes et al., 1988, Abeille, 1990, LTAG], are "lexicalized", i.e., individual words or morphemes bear syntactic features telling us which other words they can combine with. For example, "buy" could be associated with (at least) two lexicalized syntactic structures, one containing two noun phrases (as in *Kim bought cheese*), and another with three (as in *Lee bought Logan cheese*). We speculate that our NLI models might accept permuted examples at high

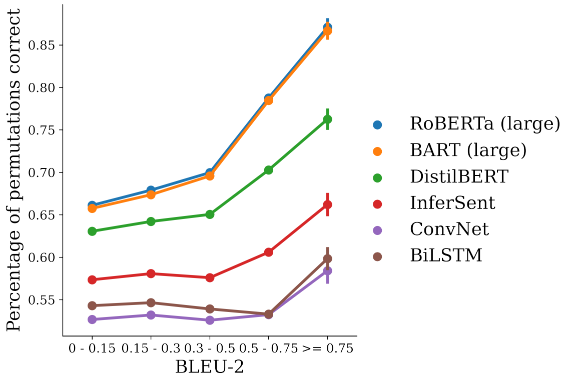


Figure 4.8 BLEU-2 score versus acceptability of permuted sentences across all test datasets. RoBERTa and BART performance is similar but differs considerably from the performance of non-Transformer-based models, such as InferSent and ConvNet.

rates, because they are (perhaps noisily) reconstructing the original sentence from abstract, word-anchored information about common neighbors.

To test this, we POS-tagged D_{train} using 17 Universal Part-of-Speech tags (using spaCy, Honnibal et al. 2020). For each $w_i \in S_i$, we compute the occurrence probability of POS tags on tokens in the *neighborhood* of w_i. The neighborhood is specified by the radius r (a symmetrical window r tokens from $w_i \in S_i$ to the left and right). We denote this sentence level probability of neighbor POS tags for a word w_i as $\psi^r_{\{w_i,S_i\}} \in \mathcal{R}^{17}$. Sentence-level word POS neighbor scores can be averaged across D_{train} to get a type level score $\psi^r_{\{w_i,D_{\text{train}}\}} \in \mathcal{R}^{17}, \forall w_i \in D_{\text{train}}$. Then, for a sentence $S_i \in D_{\text{test}}$, for each word $w_i \in S_i$, we compute a **POS mini-tree overlap score**:

$$\begin{split}\beta^k_{\{w_i,S_i\}} = \frac{1}{k} \mid \text{argmax}_k \psi^r_{\{w_i,D_{\text{train}}\}} \cap \\ \text{argmax}_k \psi^r_{\{w_i,S_i\}} \mid\end{split} \tag{4.4}$$

Concretely, $\beta^k_{\{w_i,S_i\}}$ computes the overlap of top-k POS tags in the neighborhood of a word w_i in S with that of the train statistic. If a word has the same mini-tree in a given sentence as it has in the training set, then the overlap would be 1. For a given sentence S_i, the aggregate $\beta^k_{\{S_i\}}$ is defined by the average of the overlap scores of all its words: $\beta^k_{\{S_i\}} = \frac{1}{|S_i|}\sum_{w_i \in S_i} \beta^k_{\{w_i,S_i\}}$, and we call it a POS minitree *signature*. We can also compute the POS minitree signature of a permuted sentence $\hat{S}_i$ to have $\beta^k_{\{\hat{S}_i\}}$. If the permuted sentence POS signature comes close to that of the true sentence, then their ratio (i.e., $\beta^k_{\{\hat{S}_i\}}/\beta^k_{\{S_i\}}$) will be close to 1. Also, since POS signature is computed with respect to the train distribution, a ratio of > 1 indicates that the permuted sentence is closer to the overall train statistic than to the original unpermuted sentence in terms of POS signature. If high overlap with the training distribution correlates with percentage of permutations deemed correct, then our models treat words as if they project syntactic minitrees. Figure 4.9 provides a snapshot a word “river" from the test set and shows how the POS signature distribution of the word in a particular example match with that of aggregated training statistic. In practice, we select the top k POS tags for the word in the test signature as well as the train, and calculate their overlap. When comparing the model performance with permuted sentences, we compute a ratio between the original test overlap score and an overlap score calculated instead from the permuted test. In the Figure 4.9, ‘river’ would have a POS tag minitree score of 0.75.

We investigate the relationship with percentage of permuted sentences accepted with $\beta^k_{\{\hat{S}_i\}}/\beta^k_{\{S_i\}}$ in Figure 4.10. We observe that the POS Tag Minitree hypothesis holds for Transformer-based models, RoBERTa, BART and DistilBERT, where the percentage of accepted pairs increase as the sentences have higher overlap with the un-permuted sentence in terms of POS signature. For non-Transformer models such as InferSent, ConvNet, and BiLSTM models, the POS signature ratio to percentage of correct permutation remains the same or decreases, suggesting that the reasoning process employed by these models does not preserve local abstract syntax structure (i.e., POS neighbor relations).

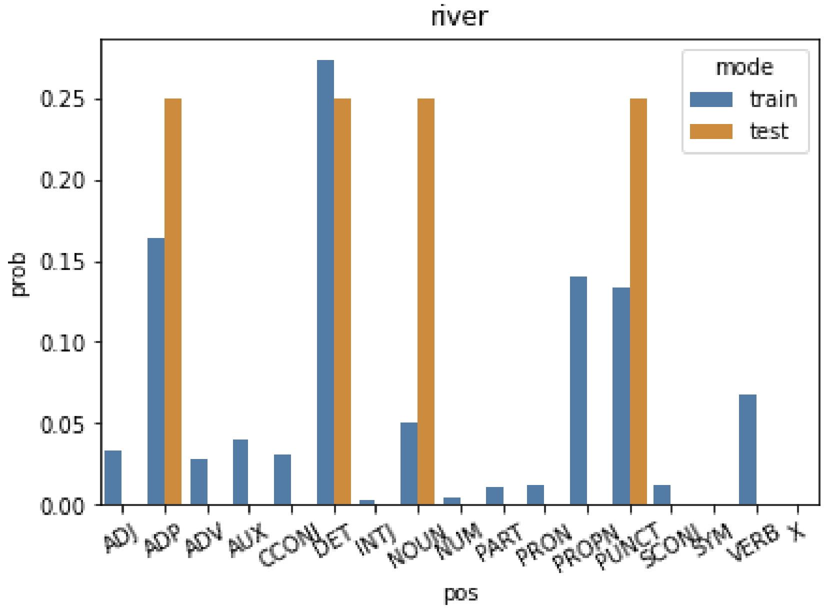


Figure 4.9 Example POS signature for the word 'river', calculated with a radius of 2. Probability of each neighbor POS tag is provided. Orange examples come from the permuted test set, and blue come from the original training data.

4.5.2 Human Evaluation

We expect humans to struggle with UNLI, given our intuitions and the sentence superiority findings (but see Mollica et al. 2020). To test this, we presented two experts in NLI (one a linguist) with permuted sentence pairs to label.[4] Concretely, we draw equal number of examples from MNLI Matched dev set (100 examples where RoBERTa predicts the gold label, D^c and 100 examples where it fails to do so, D^f), and then permute these examples using $\mathcal{F}$. The experts were given no additional information (recall that it is common knowledge that NLI is a roughly balanced 3-way classification task). Unbeknownst to the experts, all permuted sentences in the sample were actually accepted

[4] Concurrent work by Gupta et al. [2021] found that untrained crowdworkers accept NLI examples that have been subjected to different kinds of perturbations at roughly most frequent class levels—i.e., only 35% of the time.

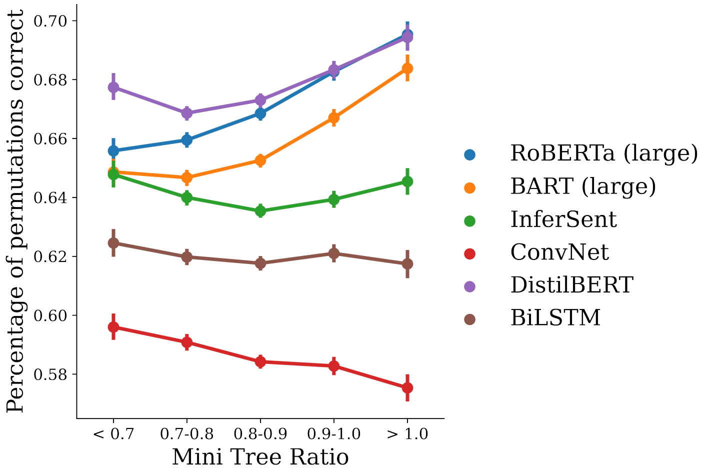


Figure 4.10 POS Tag Mini Tree overlap score and percentage of permutations which the models assigned the gold-label.

Evaluator	Accuracy	Macro F1	Acc on D^c	Acc on D^f
X	0.581 ± 0.068	0.454	0.649 ± 0.102	0.515 ± 0.089
Y	0.378 ± 0.064	0.378	0.411 ± 0.098	0.349 ± 0.087

Table 4.4 Human (expert) evaluation on 200 permuted examples from the MNLI matched development set. Half of the permuted pairs contained shorter sentences and the other, longer ones. All permuted examples were assigned the gold label by RoBERTa-Large.

by the RoBERTa (large) model (trained on MNLI dataset). We observe that the experts performed much worse than RoBERTa (Table 4.4), although their accuracy was a bit higher than random. We also find that for both experts, accuracy on permutations from D^c was higher than on D^f, which verifies findings that showed high word overlap can give hints about the ground truth label [Dasgupta et al., 2018, Poliak et al., 2018, Gururangan et al., 2018, Naik et al., 2019].

4.5.3 Training by Maximizing Entropy

We propose an initial attempt to mitigate the effect of correct prediction on permuted examples. As we observe in §4.4.2, model entropy on permuted examples is significantly lower than expected. Neural networks tend to output higher confidence than random for even unknown inputs Gandhi and Lake [2020], which might be an underlying cause of the high Permutation Acceptance.

An ideal model would be ambivalent about randomized ungrammatical sentences. Thus, we train NLI models baking in the principle of mutual exclusivity [Gandhi and Lake, 2020] by maximizing model entropy. Concretely, we fine-tune RoBERTa on MNLI while maximizing the entropy ($\mathcal{H}$) on a subset of n randomized examples ($(\hat{p}_i, \hat{r}_i)$, for each example (p, h) in MNLI. We modify the loss function as follows:

$$\mathcal{L} = \operatorname*{argmin}_{\theta} \sum_{((p,h),y)} y \log(p(y|(p,h);\theta)) + \sum_{i=1}^{n} \mathcal{H}\left(y|(\hat{p}_i, \hat{h}_i);\theta\right) \tag{4.5}$$

Using this maximum entropy method ($n = 1$), we find that the model improves considerably with respect to its robustness to randomized sentences, all while taking no hit to accuracy (Table 4.5). We observe that no model reaches a Ω_{max} score close to 0, suggesting further room to explore other methods for decreasing models' Permutation Acceptance. Similar approaches have also proven useful [Gupta et al., 2021] for other tasks as well.

4.6 Related Work

Researchers in NLP have realized the importance of syntactic structure in neural networks going back to Tabor [1994]. An early hand annotation effort on PASCAL RTE [Dagan et al., 2006] suggested that "syntactic information alone was sufficient to make a judgment" for roughly one third of examples [Vanderwende and Dolan, 2005]. Anecdotally, large generative language models like GPT-2 or -3 exhibit a seemingly hu-

Eval Dataset	$\mathcal{A}$ (V)	$\mathcal{A}$ (ME)	$\Omega_{\max}$ (V)	$\Omega_{\max}$ (ME)
MNLI_m_dev	0.905	0.908	0.984	0.328
MNLI_mm_dev	0.901	0.903	0.985	0.329
SNLI_test	0.882	0.888	0.983	0.329
SNLI_dev	0.879	0.887	0.984	0.333
ANLI_r1_dev	0.456	0.470	0.890	0.333
ANLI_r2_dev	0.271	0.258	0.880	0.333
ANLI_r3_dev	0.268	0.243	0.892	0.334

Table 4.5 NLI Accuracy ($\mathcal{A}$) and Permutation Acceptance metrics ($\Omega_{\max}$) of RoBERTa when trained on MNLI dataset using vanilla (V) and Maximum Random Entropy (ME) method.

manlike ability to generate fluent and grammatical text [Goldberg, 2019, Wolf, 2019b]. However, the jury is still out as to whether transformers genuinely acquire syntax.

4.6.1 Models appear to have acquired syntax.

When researchers have peeked inside Transformer LM's pretrained representations, familiar syntactic structure [Hewitt and Manning, 2019, Jawahar et al., 2019a, Lin et al., 2019, Warstadt and Bowman, 2020, Wu et al., 2020], or a familiar order of linguistic operations [Jawahar et al., 2019a, Tenney et al., 2019], has appeared. There is also evidence, notably from agreement attraction phenomena [Linzen et al., 2016] that transformer-based models pretrained on LM do acquire some knowledge of natural language syntax [Gulordava et al., 2018a, Chrupała and Alishahi, 2019, Jawahar et al., 2019a, Lin et al., 2019, Manning et al., 2020, Hawkins et al., 2020, Linzen and Baroni, 2021]. Results from other phenomena [Warstadt and Bowman, 2020] such as NPI licensing [Warstadt et al., 2019a] lend additional support. The claim that LMs acquire some syntactic knowledge has been made not only for transformers, but also for convolutional neural nets [Bernardy and Lappin, 2017], and RNNs [Gulordava et al., 2018a, van Schijndel and Linzen, 2018, Wilcox et al., 2018, Zhang and Bowman, 2018, Prasad et al., 2019, Ravfogel et al., 2019]—although there are many caveats (e.g., Ravfogel et al. 2018, White et al. 2018, Davis and van Schijndel 2020, Chaves 2020, Da Costa and Chaves 2020, Kodner and Gupta 2020).

4.6.2 Models appear to struggle with syntax.

Several works have cast doubt on the extent to which NLI models in particular know syntax (although each work adopts a slightly different idea of what "knowing syntax" entails). For example, McCoy et al. [2019b] argued that the knowledge acquired by models trained on NLI (for at least some popular datasets) is actually not as syntactically sophisticated as it might have initially seemed; some transformer models rely mainly on simpler, non-humanlike heuristics. In general, transformer LM performance has been found to be patchy and variable across linguistic phenomena [Dasgupta et al., 2018, Naik et al., 2018, An et al., 2019, Ravichander et al., 2019, Jeretic et al., 2020]. This is especially true for syntactic phenomena [Marvin and Linzen, 2018, Hu et al., 2020b, Gauthier et al., 2020, McCoy et al., 2020, Warstadt et al., 2020b], where transformers are, for some phenomena and settings, worse than RNNs [van Schijndel et al., 2019]. From another angle, many have explored architectural approaches for increasing a network's sensitivity to syntactic structure [Chen et al., 2017, Li et al., 2020]. Williams et al. [2018a] showed that learning jointly to perform NLI and to parse resulted in parse trees that match no popular syntactic formalisms. Furthermore, models trained explicitly to differentiate acceptable sentences from unacceptable ones (i.e., one of the most common syntactic tests used by linguists) have, to date, come nowhere near human performance [Warstadt et al., 2019c].

4.6.3 Insensitivity to Perturbation.

Most relatedly, several concurrent works [Pham et al., 2020, Alleman et al., 2021, Gupta et al., 2021, Sinha et al., 2021a, Parthasarathi et al., 2021] investigated the effect of word order permutations on transformer NNs. Pham et al. [2020] is very nearly a proper subset of our work except for investigating additional tasks (i.e. from the GLUE benchmark of Wang et al. 2018) and performing a by-layer-analysis. Gupta et al. [2021] also relies on the GLUE benchmark, but additionally investigates other types of "destructive" perturbations. Our contribution differs from these works in that we additionally include the following: we (i) outline theoretically-informed predictions for how models *should be expected* to react to permuted input (we outline a few options), (ii) show that permuting can "flip" an incorrect prediction to a correct one, (iii) show that the problem isn't specific to Transformers, (iv) show that the problem persists on out of domain data, (v) offer a suite of flexible metrics, and (vi) analyze *why* models might be accepting permutations (BLEU and POS-tag neighborhood analysis). Finally, we replicate our findings in another language. While our work (and Pham et al., Gupta et al.) only permutes data during fine-tuning and/or evaluation, recently Sinha et al. explored the sensitivity during pre-training, and found that models trained on n-gram permuted sentences perform remarkably close to regular MLM pre-training. In the context of generation, Parthasarathi et al. [2021] crafted linguistically relevant perturbations (on the basis of part-of-speech tagging and dependency parsing) to evaluate whether permutation hinders automatic machine translation models. Relatedly, but not for translation, Alleman et al. [2021] investigated a smaller inventory of perturbations with emphasis on phrasal boundaries and the effects of n-gram perturbations on different layers in the network.

4.6.4 NLI Models are very sensitive to words.

NLI models often over-attend to particular words to predict the correct answer [Gururangan et al., 2018, Clark et al., 2019b]. Wallace et al. [2019] show that some short sequences of non-human-readable text can fool many NLU models, including NLI models trained on SNLI, into predicting a specific label. In fact, Ettinger [2020] observed that for one of three test sets, BERT loses some accuracy in word-perturbed sentences, but that there exists a subset of examples for which BERT's accuracy remains intact. If performance isn't affected (or if permutation helps, as we find it does in some cases), it suggests that these state-of-the-art models actually perform somewhat similarly to bag-of-words models Blei et al. [2003], Mikolov et al. [2013a].

4.7 Discussion

In this chapter, we observe that state-of-the-art models do not rely on sentence structure the way we think they should: NLI models (Transformer-based models, RNNs, and ConvNets) are largely insensitive to permutations of word order that corrupt the original syntax. This raises questions about the extent to which such systems understand "syntax", and highlights the unnatural language understanding processes they employ. To summarize, our primary observations from this chapter are:

- **NLU models can still perform the task even if the word orders are scrambled** We observed overwhelming evidence in §4.4.1 that state-of-the-art NLU models tend to perform the task comparably even on word order scrambled text, which has no inherent semantic meaning.

- **Certain permutations allow NLU models to flip classification labels, leading to better task scores.** We find that certain permutations of the order of words in a given input sentence pair can trigger the model to change the classification labels from the baseline, leading to large performance gains in the scores of the NLI task. For instance, in the examples which the models find difficult to predict, a permutation of the word order can elicit the model to assign the gold label.

- **NLU models display rudimentary understanding of syntax, as evident by the preservation of abstract parts-of-speech neighborhood information.** We do find that models seem to have learned some syntactic information as is evidenced by a correlation between preservation of abstract POS neighborhood information and rate of acceptance by models, but these results do not discount the high rates of Permutation Acceptance, and require further verification.

Given these findings, and coupled with the observation that humans cannot perform UNLI at all well, the high rate of permutation acceptance that we observe leads us to conclude that current models do not yet "know syntax" in the fully systematic and humanlike way we would like them to. This study leads us to further investigate the training dynamics employed by these large language models. In the next chapter, we will investigate the training data dependency of one such model family in detail, to shed more light on the sentence processing pipelines employed by these models.

4.8 Follow-up findings in the community

Our work in this chapter was among the first to discover the issue of word order permutation in the large language models. Concurrent to our publication, Gupta et al. [2021] and Pham et al. [2020] also converge on similar findings on different NLU datasets. Thus, our paper opened a new line of inquiry on the susceptibility of word order of language models, and in general encouraging the community to probe the cause of related systematicity issues displayed by these models. Hessel and Schofield [2021] arrive at similar conclusions on word-order invariance as demonstrated in our work in this chapter. The authors then further utilizes the phenomenon to develop privacy sensitive NLU models which can operate on a bag-of-words representation of a document without losing task performance. Malkin et al. [2021] also provide similar conclusions to our work, and they leverage the word-order invariance of Transformer-family of models to develop a new iterative shuffling inference mechanism to find out the most likely word order of a sentence according to a given model, and thereby demonstrate improvements to language modelling and generation. Perez et al. [2021b] take inspiration from the word-order insensitivity results on Transformer models from our work to develop a novel method, Rissanen Data Analysis, to estimate the models learning ability by measuring the Minimum Description Length (MDL). MDL captures the training behavior of the model during the initial stages of fine-tuning. The authors compute the MDL on various downstream tasks with and without word order shuffling, to find that models display significantly different learning behavior on word order shuffles than on the normal mode [Section 4.3.4, pp. 8. Perez et al., 2021a]. However, having different learning characteristics does not fully explain our results of word-order insensitivity, indicating the issue requires further study to find out the root cause. Parthasarathi et al. [2021] extend our experimental setup to investigate how neural machine translation (NMT) systems translate ungrammatical, shuffled input sentences. They observe that state-of-the-art NMT models overwhelmingly *fix* the ungrammaticality of the input in their translations, thereby exhibiting robust behavior. Clouatre et al. [2022] attempts to explain the word-order insensitivity issue of several types of neural models, including Transformer-family of models. They come to the conclusion that neural models across all types of inductive bias (i.e., including RNN/LSTM/Transformer models) rely on local structure of text instead of the global structure to build the sentence representation, which also corroborates to our syntax structure hypothesis (§4.5.1). Webson and Pavlick [2022] investigate the systematicity issues in prompt-based models, specifically to gain insights on whether high-performing prompt-based models understand the meaning of the said prompts. They find that despite the impressive zero/few-shot performance of massive language models such as GPT3 [Brown et al., 2020], they lack proper systematic understanding capability of the provided prompts, so much so that they produce good predictions even with misleading and irrelevant prompts. Tejankar et al. [2021] leverage the word order insensitivity of Transformer models to develop a bag-of-words image captioning model which are trained on synthetically crafted bag-of-word captions. They achieve superior zero shot performance on ImageNet dataset by using 1/5th amount of image-caption pairs than the baseline model. The following chapter is also a direct consequence of the results observed in this work.

Chapter 5

Probing syntax understanding through distributional hypothesis

The field of natural language processing (NLP) has become dominated by the pretrain-and-finetune paradigm, where we first obtain a good parametric *prior* in order to subsequently model downstream tasks accurately. In particular, masked language model (MLM) pre-training, as epitomized by BERT [Devlin et al., 2019b], has proven wildly successful, although the precise reason for this success has remained unclear. On one hand, we can view BERT as the newest in a long line of NLP techniques [Deerwester et al., 1990, Landauer and Dumais, 1997, Collobert and Weston, 2008, Mikolov et al., 2013b, Peters et al., 2018] that exploit the well-known distributional hypothesis [Harris, 1954]. On the other hand, it has been claimed that BERT "rediscovers the classical NLP pipeline" [Tenney et al., 2019], suggesting that it has learned "the types of syntactic and semantic abstractions traditionally believed necessary for language processing" rather than "simply modeling complex co-occurrence statistics" (ibid. p.1).

In this chapter, we aim to uncover how much of MLM's success comes from learning simple distributional information, as opposed to grammatical abstractions [Tenney et al., 2019, Manning et al., 2020]. Thus, in this chapter I discuss our work [Sinha et al., 2021a], where we disentangle these two hypotheses by measuring the effect of removing word order information during pre-training: any sophisticated (English) NLP pipeline would presumably depend on the syntactic information conveyed by the order of words. We find that surprisingly most of MLM's high performance can in fact be explained by the "distributional prior" rather than its ability to replicate the classical NLP pipeline.

Concretely, we pre-train MLMs (RoBERTa, Liu et al. 2019b) on various corpora with permuted word order while preserving some degree of distributional information, and examine their downstream performance. We also experiment with training MLMs without positional embeddings, making them entirely order agnostic, and with training on a corpus sampled from the source corpus's unigram distribution. We then evaluate these "permuted" models in a wide range of settings and compare with regularly-pre-trained models.

We demonstrate that pre-training on permuted data has surprisingly little effect on downstream task performance after fine-tuning (on non-shuffled training data). In our previous chapter we observed that MLMs are quite robust to permuting downstream test data (§4.4.1) and even do quite well using permuted "unnatural" downstream train data [Sinha et al., 2021b, Gupta et al., 2021]. In this chapter, we show that downstream performance for "unnatural language pre-training" is much closer to standard MLM pre-training than one might expect.

In an effort to shed light on these findings, we experiment with various probing tasks. We verify via non-parametric probes that the permutations do in fact make the model worse at syntax-dependent tasks. However, just like on the downstream fine-tuning tasks, permuted models perform well on parametric syntactic probes, in some cases almost matching the unpermuted model's performance, which is quite surprising given how important word order is crosslinguistically (Greenberg 1963, Dryer 1992, Cinque 1999, i.a.).

Our results can be interpreted in different ways. One could argue that our downstream and probing tasks are flawed, and that we need to examine models with examples that truly test strong generalization and compositionality. Alternatively, one could argue that prior works have overstated the dependence of human language understanding on word order, and that human language understanding depends less on the structure of the sentence and more on the structure of the *world*, which can be inferred to a large extent from distributional information. This work is meant to deepen our understanding of MLM pre-training and, through this, move us closer to finding out what is actually required for adequately modelling natural language.

5.1 Technical Background

Masked Language Modelling (MLM) is a pre-training technique popularized by BERT [Devlin et al., 2019b]. Unlike the autoregressive processing of traditional Transformer [Vaswani et al., 2017] models, BERT introduces bi-directional attention mechanism in which the model no longer have to restrict attentions to past tokens: the model can now attend to any tokens in the input, in both left-to-right and right-to-left directions (§2.2.4). To train such bi-directional self-attention models, MLM pre-training is used where first the input sentence is corrupted by replacing 15% of tokens with mask token (`<MASK>`), and then the training objective entails predicting the masked tokens in the output. Thus, to predict any mask token, the model can attend to the entire context (hence the bi-directionality). This kind of pre-training technique allows models to be trained in larger batch sizes, effectively utilizing the GPU operations. Having access to bi-directional context also allows the model to learn long-range dependencies.

5.2 Experimental Setup

5.2.1 Sentence word order permutation

To investigate to what extent the performance of MLM pre-training is a consequence of distributional information, we construct a training corpus devoid of natural word order but preserving local distributional information. We construct word order-randomized versions of the BookWiki corpus (the Toronto Books Corpus, [Zhu et al., 2015], plus English Wikipedia) from Liu et al. [2019b], following the setup described in §4.2. Concretely, given a sentence S containing N words, we permute the sentence using a seeded random function $\mathcal{F}_1$ such that no word can remain in its original position. In total, there exist $(N-1)!$ possible permutations of a given sentence. We randomly sample a single permutation per sentence, to keep the total dataset size similar to the original.

We extend the permutation function $\mathcal{F}_1$ to a function $\mathcal{F}_n$ that preserves n-gram information. Specifically, given a sentence S of length N and n-gram value n, we sample a starting position i for possible contiguous n-grams $\in \{0, N-n\}$ and convert the span $S[i, i+n]$ to a single token, to form $\hat{S}$, of length $\hat{N} = N - (n+1)$. We continue this process repeatedly (without using the previously created n-grams) until there exists no starting position for selecting a contiguous n-gram in $\hat{S}$. For example, given a sentence of length $N = 6$, $\mathcal{F}_4$ will first convert one span of 4 tokens into a word, to have $\hat{S}$ consisting of three tokens (one conjoined token of 4 contiguous words, and two leftover words). Then, the resulting sentence $\hat{S}$ is permuted using $\mathcal{F}_1$. We train RoBERTa models on four permutation variants of BookWiki corpus, $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$ for each n-gram value $\in \{1, 2, 3, 4\}$.

We provide pseudo-code for $\mathcal{F}_i$ in Algorithm 1. Following the formulation in §4.2, we do not explicitly control whether the permuted words maintain any of their original neighbors. Thus, a certain amount of extra n-grams are expected to co-occur, purely as a product of random shuffling. We quantify the amount of such shuffling on a sample of 1 million sentences drawn from the BookWiki random corpus, and present the BLEU-2, BLEU-3 and BLEU-4 scores in Table 5.1. We provide a sample snapshot of the generated data in Table 5.18.

	BLEU-2	BLEU-3	BLEU-4
$\mathcal{M}_1$	0.493 +/- 0.12	0.177 +/- 0.16	0.040 +/- 0.11
$\mathcal{M}_2$	0.754 +/- 0.07	0.432 +/- 0.18	0.226 +/- 0.19
$\mathcal{M}_3$	0.824 +/- 0.06	0.650 +/- 0.09	0.405 +/- 0.20
$\mathcal{M}_4$	0.811 +/- 0.08	0.671 +/- 0.11	0.553 +/- 0.12

Table 5.1 BLEU-2,3,4 scores (mean and std dev) on a sample of 1M sentences drawn from the corpus used to train $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$and $\mathcal{M}_4$ compared to $\mathcal{M}_{\mathrm{N}}$.

5.2.2 Corpus word order bootstrap resample

The above permutations preserve higher order distributional information by keeping words from the same sentence together. However, we need a baseline to understand how a model would perform without such co-occurrence information. We construct a baseline, $\mathcal{M}_{\mathrm{UG}}$, that captures word/subword information, without access to co-occurrence statistics. To construct $\mathcal{M}_{\mathrm{UG}}$, we sample unigrams from BookWiki according to their frequencies, while also treating named entities as unigrams. We leverage Spacy [Honnibal et al., 2020][1] to extract unigrams and named entities from the corpus, and construct $\mathcal{M}_{\mathrm{UG}}$ by drawing words from this set according to their frequency. This allows us to construct $\mathcal{M}_{\mathrm{UG}}$ such that it has exactly the same size as BookWiki but without any distributional (i.e. co-occurrence) information beyond the unigram frequency distribution. Our hypothesis is that any model pre-trained on this data will perform poorly, but it should provide a baseline for the limits on learning language of the inductive bias of the model in isolation.

[1] https://spacy.io/

Algorithm 1 SentenceRandomizer

```
procedure $\mathcal{F}(S, t, n)$    ▷ Randomize a sentence $S$ with seed $t$ and n grams $n$
    $W$ = tokenize the words in $S$
    Set the seed to $t$
    if $n > 1$ then
        while True do
            $K$ = Sample all possible starting points from $[0, |W| - n]$
            Ignore the starting points in $K$ which overlap with conjoined tokens    ▷ Conjoined tokens consists of joined unigrams
            if $|K| \geq 1$ then
                Sample one position $p \in K$
                $g$ = Extract the n-gram $W[p : p + n]$
                Delete $W[p + 1 : p + n]$
                $W[p]$ = Convert $g$ to a conjoined token
            else
                Break from While loop
    while True do
        $\hat{W}$ = randomly shuffle tokens in $W$
        $r = \sum(\hat{W}[i] = W[i])$    ▷ Count number of positions where the token remains in its original position
        if $r = 0$ then Break out of While loop
    $\hat{S}$ = join the tokens in $\hat{W}$
    Return $\hat{S}$
```

5.2.3 Further baselines

To investigate what happens if a model has absolutely no notion of word order, we also experiment with pre-training RoBERTa on the original corpus without positional embeddings. Concretely, we modify the RoBERTa architecture to remove the positional embeddings from the computation graph, and then proceed to pre-train on the natural order BookWiki corpus. We denote this model $\mathcal{M}_{\texttt{NP}}$. Finally, we consider a randomly initialized RoBERTa model $\mathcal{M}_{\texttt{RI}}$ to observe the extent we can learn from each task with only the model's base inductive bias.

5.3 Evaluated Models & Tasks

We use the RoBERTa (base) [Liu et al., 2019b] MLM architecture, due to its relative computational efficiency and good downstream task performance. We expect that other variants of MLMs would provide similar insights, given their similar characteristics.

In all of our experiments, we use the original 16GB BookWiki corpus.[2] We denote the model trained on the original, un-modified BookWiki corpus as $\mathcal{M}_{\texttt{N}}$ (for "natural"). We use two types of word order randomization methods: permuting words at the sentence level, and resampling words at the corpus level.

5.3.1 Pre-training details

Each model $\in \{\mathcal{M}_{\texttt{N}}, \mathcal{M}_{\texttt{1}}, \mathcal{M}_{\texttt{2}}, \mathcal{M}_{\texttt{3}}, \mathcal{M}_{\texttt{4}}, \mathcal{M}_{\texttt{UG}}, \mathcal{M}_{\texttt{NP}}\}$ is a RoBERTa-base model (12 layers, hidden size of 768, 12 attention heads, 125M parameters), trained for 100k updates using 8k batch-size, 20k warmup steps, and 0.0006 peak learning rate. These are identical hyperparameters to Liu et al. [2019b], except for the number of warmup steps which we changed to 20k for improved training stability. Each model was trained

[2] We release the pre-trained RoBERTa models used in our experiments through the FairSeq repository: https://github.com/pytorch/fairseq/tree/master/examples /shuffled_word_order.

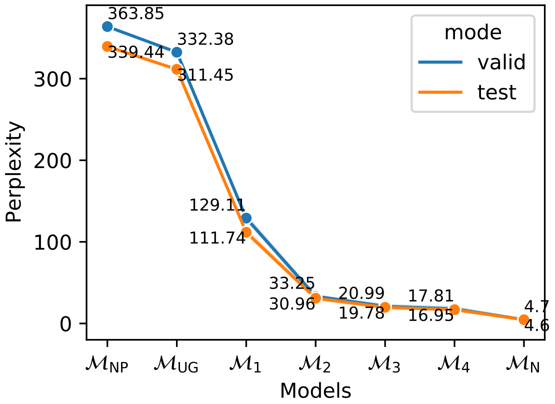


Figure 5.1 Perplexity of various models on Wiki 103 valid and test sets.

using 64 GPUs for up to 72 hours each. We train three seeds for each data configuration. We use FairSeq [Ott et al., 2019] for the pre-training and fine-tuning experiments. We use the Wiki 103 validation and test set to validate and test the array of pre-trained models, as validation on this small dataset is quick, effective, and reproducible for comparison among publicly available datasets (Figure 5.1). We observe that perplexity monotonically increases from $\mathcal{M}_{\mathrm{N}}$, through $\mathcal{M}_4$–$\mathcal{M}_1$, to $\mathcal{M}_{\mathrm{UG}}$, and finally $\mathcal{M}_{\mathrm{NP}}$.

5.3.2 Fine-tuning tasks

We evaluate downstream performance using the General Language Understanding and Evaluation (GLUE) benchmark, the Paraphrase Adversaries from Word Scrambling (PAWS) dataset, and various parametric and non-parametric tasks (see §5.4.2).

GLUE. The GLUE [Wang et al., 2018] benchmark is a collection of 9 datasets for evaluating natural language understanding systems, of which we use Corpus of Linguistic Acceptability [CoLA, Warstadt et al., 2019b], Stanford Sentiment Treebank [SST, Socher et al., 2013], Microsoft Research Paragraph Corpus [MRPC, Dolan and Brockett, 2005], Quora Question Pairs (QQP)[3], Multi-Genre NLI [MNLI, Williams et al., 2018c], Question NLI [QNLI, Rajpurkar et al., 2016, Demszky et al., 2018], Recognizing Textual Entailment [RTE, Giampiccolo et al., 2007a, Haim et al., 2006, Giampiccolo et al., 2007b, Bentivogli et al., 2009]. Pham et al. [2020] show the word order insensitivity of several GLUE tasks (QQP, SST-2), evaluated on public regularly pre-trained checkpoints.

PAWS. The PAWS task [Zhang et al., 2019] consists of predicting whether a given pair of sentences are paraphrases. This dataset contains both paraphrase and non-paraphrase pairs with high lexical overlap, which are generated by controlled word swapping and back translation. Since even a small word swap and perturbation can drastically modify the meaning of the sentence, we hypothesize the randomized pre-trained models will struggle to attain a high performance on PAWS.

Fine-tuning details. We use the same fine-tuning methodology used by Liu et al. [2019b], where we run hyperparameter search over the learning rates $\{1 \times 10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}\}$ and batch sizes $\{16, 32\}$ for each model. For the best hyperparam configurations of each model, we fine-tune with 5 different seeds and report the mean and standard deviation for each setting. $\mathcal{M}_{\mathrm{NP}}$ is fine-tuned without positional embeddings, matching the way it was pre-trained.

5.4 Results

5.4.1 Downstream task results

In this section, we present the downstream task performance of the models defined in §5.2. For evaluation, we report Matthews correlation for CoLA and accuracy for all other tasks.

Model	QNLI	RTE	QQP	SST-2	MRPC	PAWS	MNLI-m/mm	CoLA
$\mathcal{M}_{\text{N}}$	92.45 +/- 0.2	73.62 +/- 3.1	91.25 +/- 0.1	93.75 +/- 0.4	89.09 +/- 0.9	94.49 +/- 0.2	86.08 +/- 0.2 / 85.4 +/- 0.2	52.45 +/- 21
$\mathcal{M}_4$	91.65 +/- 0.1	70.94 +/- 1.2	91.39 +/- 0.1	92.46 +/- 0.3	86.90 +/- 0.3	94.26 +/- 0.2	83.79 +/- 0.2 / 83.94 +/- 0.3	35.25 +/- 32
$\mathcal{M}_3$	91.56 +/- 0.4	69.75 +/- 2.8	91.22 +/- 0.1	91.97 +/- 0.5	86.22 +/- 0.8	94.03 +/- 0.1	83.83 +/- 0.2 / 83.71 +/- 0.1	40.78 +/- 23
$\mathcal{M}_2$	90.51 +/- 0.1	70.00 +/- 2.5	91.33 +/- 0.0	91.78 +/- 0.3	85.90 +/- 1.2	93.53 +/- 0.3	83.45 +/- 0.3 / 83.54 +/- 0.3	50.83 +/- 5.8
$\mathcal{M}_1$	89.05 +/- 0.2	68.48 +/- 2.5	91.01 +/- 0.0	90.41 +/- 0.4	86.06 +/- 0.8	89.69 +/- 0.6	82.64 +/- 0.1 / 82.67 +/- 0.2	31.08 +/- 10
$\mathcal{M}_{\text{NP}}$	77.59 +/- 0.3	54.78 +/- 2.2	87.78 +/- 0.4	83.21 +/- 0.6	72.78 +/- 1.6	57.22 +/- 1.2	63.35 +/- 0.4 / 63.63 +/- 0.2	2.37 +/- 3.2
$\mathcal{M}_{\text{UG}}$	66.94 +/- 9.2	53.70 +/- 1.0	85.57 +/- 0.1	83.17 +/- 1.5	70.57 +/- 0.7	58.59 +/- 0.3	71.93 +/- 0.2 / 71.33 +/- 0.5	0.92 +/- 2.1
$\mathcal{M}_{\text{RI}}$	62.17 +/- 0.4	52.97 +/- 0.2	81.53 +/- 0.2	82.0 +/- 0.7	70.32 +/- 1.5	56.62 +/- 0.0	65.70 +/- 0.2 / 65.75 +/- 0.3	8.06 +/- 1.6

Table 5.2 GLUE and PAWS-Wiki dev set results on different RoBERTa (base) models trained on variants of the BookWiki corpus (with mean and std). The top row is the original model, the middle half contains our primary models under investigation, and the bottom half contains the baselines.

Word order permuted pre-training

In our first set of experiments, we finetune the pre-trained models on the GLUE and PAWS tasks. We report the results in Table 5.2.[4] First, we observe that the model without access to distributional or word order information, $\mathcal{M}_{\text{UG}}$ (unigram) performs much worse than $\mathcal{M}_{\text{N}}$ overall: $\mathcal{M}_{\text{UG}}$ is 18 points worse than $\mathcal{M}_{\text{N}}$ on average across the accuracy-based tasks in Table 5.2 and has essentially no correlation with human judgments on CoLA. $\mathcal{M}_{\text{UG}}$ $\mathcal{M}_{\text{NP}}$ and $\mathcal{M}_{\text{RI}}$ perform comparably on most of the tasks, while achieving surprisingly high scores in QQP and SST-2. However, all three models perform significantly worse on GLUE and PAWS, compared to $\mathcal{M}_{\text{N}}$ (Table 5.2, bottom half). $\mathcal{M}_{\text{UG}}$ reaches up to 71.9 on MNLI - possibly due to the fact that $\mathcal{M}_{\text{UG}}$ has access to (bags of) words and some phrases (from NER) is beneficial for MNLI. For the majority of tasks, the difference between $\mathcal{M}_{\text{NP}}$ and $\mathcal{M}_{\text{RI}}$ is small - a pure bag of words model performs comparably to a randomly initialized model.

Next, we observe a significant improvement on all tasks when we give models ac-

[3] http://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

[4] The $\mathcal{M}_{\text{N}}$ results are not directly comparable with that of publicly released `roberta-base` model by Liu et al. [2019b], as that uses the significantly larger 160GB corpus, and is trained for 500K updates. For computational reasons, we restrict our experiments to the 16GB BookWiki corpus and 100K updates, mirroring the RoBERTa ablations.

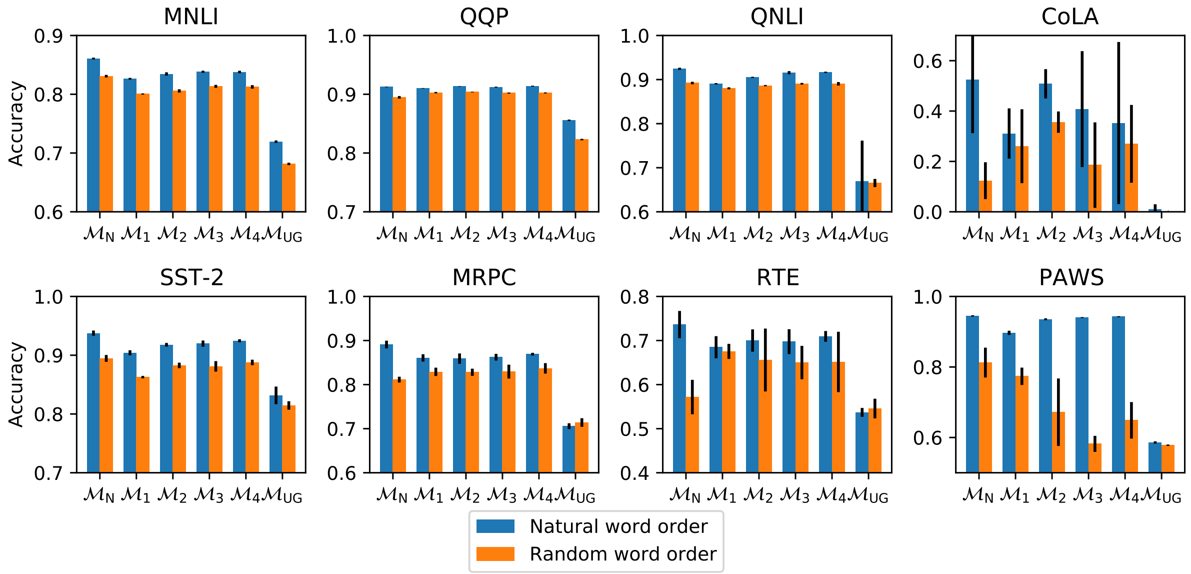


Figure 5.2 GLUE & PAWS task dev performance when finetuned on naturally (blue) and randomly ordered (orange) text, respectively, using pre-trained RoBERTa (base) models trained on different versions of BookWiki corpus.

cess to sentence-level distributional information during pre-training. $\mathcal{M}_1$, the model pre-trained on completely shuffled sentences, is on average only 3.3 points lower than $\mathcal{M}_{\mathrm{N}}$ on the accuracy-based tasks, and within 0.3 points of $\mathcal{M}_{\mathrm{N}}$ on QQP. Even on PAWS, which was designed to require knowledge of word order, $\mathcal{M}_1$ is within 5 points of $\mathcal{M}_{\mathrm{N}}$. Randomizing n-grams instead of words during pre-training results in a (mostly) smooth increase on these tasks: $\mathcal{M}_4$, the model pre-trained on shuffled 4-grams, trails $\mathcal{M}_{\mathrm{N}}$ by only 1.3 points on average, and even comes within 0.2 points of $\mathcal{M}_{\mathrm{N}}$ on PAWS. We observe a somewhat different pattern on CoLA, where $\mathcal{M}_2$ does almost as well as $\mathcal{M}_{\mathrm{N}}$ and outperforms $\mathcal{M}_3$ and $\mathcal{M}_4$, though we also observe very high variance across random seeds for this task. Crucially, we observe that $\mathcal{M}_1$ outperforms $\mathcal{M}_{\mathrm{NP}}$ by a large margin. This shows that positional embeddings are critical for learning, even when the word orders themselves are not natural. Recall, $\mathcal{M}_{\mathrm{NP}}$ is fed natural sentences as $\mathcal{M}_{\texttt{N}}$ while not having the ability to learn positional embeddings. To further quantify the effect of positional embeddings, we also investigated the effect of shuffling the entire context window, to keep the co-occurrence information same as $\mathcal{M}_{\texttt{NP}}$ in §5.5.1. We observed this model to be worse than $\mathcal{M}_{\texttt{1}}$ but significantly better than $\mathcal{M}_{\texttt{NP}}$ to support the claim about the importance of positional embeddings while training.

Overall, these results confirm our hypothesis that RoBERTa's strong performance on downstream tasks can be explained for a large part by the distributional prior.

Word order permuted fine-tuning

There are two possible explanations for the results in §5.4.1: either the tasks do not need word order information to be solved, or any necessary word order information can be acquired during fine-tuning. To examine this question, we permute the word order during fine-tuning as well. Concretely, for each task, we construct a unigram order-randomized version of each example in the fine-tuning training set using $\mathcal{F}_1$. We then fine-tune our pre-trained models on this shuffled data and evaluate task performance. For all experiments, we evaluate and perform early stopping on the original, natural word order dev set, in order to conduct a fair evaluation on the exact same optimization setup for all models.

Our results in Figure 5.2 provide some evidence for both hypotheses. On QQP and QNLI, accuracy decreases only slightly for models fine-tuned on shuffled data. Models can also achieve above 80% accuracy on MNLI, SST-2, and MRPC when fine-tuned on shuffled data, suggesting that purely lexical information is quite useful on its own.

On the other hand, for all datasets besides QQP and QNLI, we see noticeable drops in accuracy when fine-tuning on shuffled data and testing on normal order, both for $\mathcal{M}_{\texttt{N}}$ and for shuffled models $\mathcal{M}_{\texttt{1}}$ through $\mathcal{M}_{\texttt{4}}$. This suggests both that word order information is useful for these tasks, and that shuffled models must be learning to use word order information during fine-tuning.[5] Having word order during fine-tuning is especially important for achieving high accuracy on CoLA, RTE (cf. Pham et al. 2020), as well as PAWS, suggesting that these tasks are the most word order reliant. Recent research [Yu and Ettinger, 2021] raised some questions about potential artefacts inflating performance on PAWS: their swapping-distance cue of appears consistent both with our finding of high PAWS performance for n-gram shuffled models in Table 5.2, and with our PAWS results in Figure 5.2, which suggests that PAWS performance does in fact rely to some extent on natural word order at the fine-tuning stage.

Finally, for CoLA, MRPC, and RTE, performance is higher after fine-tuning on shuffled data for $\mathcal{M}_1$ than $\mathcal{M}_\mathrm{N}$. We hypothesize that $\mathcal{M}_\mathrm{N}$ represents shuffled and non-shuffled sentences very differently, resulting in a domain mismatch problem when fine-tuning on shuffled data but evaluating on non-shuffled data.[6] Since $\mathcal{M}_1$ never learns to be sensitive to word order during pre-training or fine-tuning, it does not suffer from that issue. Our results in this section also highlights the issues with these datasets, concurrent to the findings that many GLUE tasks does not need sophisticated linguistic knowledge to solve, as models typically tend to exploit the statistical artefacts and spurious correlations during fine-tuning (cf. Gururangan et al. 2018, Poliak et al. 2018, Tsuchiya 2018, McCoy et al. 2019b). However, our results overwhelmingly support the fact that word order does not matter during pre-training, if the model has the opportunity to learn the necessary information about word order during fine-tuning.

[5] We perform additional experiments on how the model representations change during fine-tuning for shuffled training using Risannen Data Analysis in §5.5.6.

[6] We further study the domain mismatch problem by evaluating on shuffled data *after* fine-tuning on the shuffled data for models in §5.5.3. We observe that models improves their scores on evaluation on shuffled data when the training data source is changed from natural to shuffled - highlighting domain match effect.

5.4.2 Probing results

To investigate how much syntactic information is contained in the MLM representations, we evaluate several probing tasks on our trained models. We consider two classes of probes: *parametric* probes, which make use of learnable parameters, and *non-parametric* probes, which directly examine the language model's predictions.

Parametric Probing

To probe our models for syntactic, semantic and other linguistic properties, we investigate dependency parsing using Pareto probing [Pimentel et al., 2020a] and the probing tasks from Conneau et al. [2018a] in SentEval [Conneau and Kiela, 2018].

Pimentel et al. [2020a] proposed a framework based on Pareto optimality to probe for syntactic information in contextual representations. They suggest that an optimal probe should balance optimal performance on the probing task with the complexity of the probe. Following their setup, we use the "difficult" probe: dependency parsing (DEP). We also investigate the "easy" probes, dependency arc labeling (DAL) and POS tag prediction (POS), results are reported §5.5.8. We probe with Linear and MLP probes, and inspect the task accuracy in terms of Unlabeled Attachment Score (UAS). The dependency parsing probe used in Pimentel et al. [2020a] builds on the Biaffine Dependency Parser [Dozat and Manning, 2017], but with simple MLPs on top of the Transformer representations.[7]

Training setup. Similar to the setup by Pimentel et al. [2020a], we run 50 random hyperparameter searches on both MLP and Linear probes by uniformly sampling from the number of layers (0-5), dropout (0-0.5), log-uniform hidden size $[2^5, 2^{10}]$. We triple this experiment size by evaluating on three pre-trained models of different seeds for each model configuration. We consider Pimentel et al.'s English dataset, derived from

[7] We experimented with a much stronger, state-of-the-art Second order Tree CRF Neural Dependency Parser Zhang et al. [2020], but did not observe any difference in UAS with different pre-trained models (see §5.5.4)

Universal Dependencies EWT (UD EWT) [Bies et al., 2012, Silveira et al., 2014] which contains 12,543 training sentences. Additionally, we experiment on the Penn Treebank dataset (PTB), which contains 39,832 training sentences.[8] We report the mean test accuracy over three seeds for the best dev set accuracy for each task.[9]

Model	UD EWT		PTB	
	MLP	Linear	MLP	Linear
$\mathcal{M}_N$	80.41 +/- 0.85	66.26 +/- 1.59	86.99 +/- 1.49	66.47 +/- 2.77
$\mathcal{M}_4$	78.04 +/- 2.06	65.61 +/- 1.99	85.62 +/- 1.09	66.49 +/- 2.02
$\mathcal{M}_3$	77.80 +/- 3.09	64.89 +/- 2.63	85.89 +/- 1.01	66.11 +/- 1.68
$\mathcal{M}_2$	78.22 +/- 0.88	64.96 +/- 2.32	84.72 +/- 0.55	64.69 +/- 2.50
$\mathcal{M}_1$	69.26 +/- 6.00	56.24 +/- 5.05	79.43 +/- 0.96	57.20 +/- 2.76
$\mathcal{M}_{UG}$	74.15 +/- 0.93	65.69 +/- 7.35	80.07 +/- 0.79	57.28 +/- 1.42

Table 5.3 Unlabeled Attachment Score (UAS) (mean and std) on the dependency parsing task (DEP) on two datasets, UD EWT and PTB, using the Pareto Probing framework [Pimentel et al., 2020a].

Model	**Length** (Surface)	**WordContent** (Surface)	**TreeDepth** (Syntactic)	**TopConstituents** (Syntactic)	**BigramShift** (Syntactic)	**Tense** (Semantic)	**SubjNumber** (Semantic)	**ObjNumber** (Semantic)	**OddManOut** (Semantic)	**CoordInversion** (Semantic)
$\mathcal{M}_N$	78.92 +/- 1.91	31.83 +/- 1.75	35.97 +/- 1.38	**78.26** +/- 4.08	**81.82** +/- 0.55	87.83 +/- 0.51	85.05 +/- 1.23	75.94 +/- 0.68	58.40 +/- 0.33	**70.87** +/- 2.46
$\mathcal{M}_4$	92.88 +/- 0.15	57.78 +/- 0.36	40.05 +/- 0.29	72.50 +/- 0.51	76.12 +/- 0.29	88.32 +/- 0.13	**85.65** +/- 0.13	82.95 +/- 0.05	**58.89** +/- 0.30	61.31 +/- 0.19
$\mathcal{M}_3$	91.52 +/- 0.16	48.81 +/- 0.26	38.63 +/- 0.61	70.29 +/- 0.31	77.36 +/- 0.12	86.74 +/- 0.12	83.83 +/- 0.38	80.99 +/- 0.26	57.01 +/- 0.21	60.00 +/- 0.26
$\mathcal{M}_2$	**93.54** +/- 0.29	62.52 +/- 0.21	**41.40** +/- 0.32	74.31 +/- 0.29	75.44 +/- 0.14	**87.91** +/- 0.35	84.88 +/- 0.11	83.98 +/- 0.14	57.60 +/- 0.36	59.46 +/- 0.37
$\mathcal{M}_1$	88.33 +/- 0.14	**64.03** +/- 0.34	40.24 +/- 0.20	70.94 +/- 0.38	58.37 +/- 0.40	87.88 +/- 0.08	83.49 +/- 0.12	**83.44** +/- 0.06	56.51 +/- 0.26	56.98 +/- 0.50
$\mathcal{M}_{UG}$	86.69 +/- 0.33	36.60 +/- 0.33	32.53 +/- 0.76	61.54 +/- 0.60	57.42 +/- 0.04	68.45 +/- 0.23	71.25 +/- 0.12	66.63 +/- 0.21	50.06 +/- 0.40	56.26 +/- 0.17

Table 5.4 SentEval Probing [Conneau et al., 2018a, Conneau and Kiela, 2018] results (with mean and std) on different model variants.

Results. We observe that the UAS scores follow a similar linear trend as the finetuning results in that $\mathcal{M}_1 \approx \mathcal{M}_{UG} < \mathcal{M}_2 < \mathcal{M}_3 < \mathcal{M}_4 < \mathcal{M}_N$ (Table 5.3). Surprisingly, $\mathcal{M}_{UG}$ probing scores seem to be somewhat better than $\mathcal{M}_1$ (though with large overlap in their standard deviations), even though $\mathcal{M}_{UG}$ cannot learn information related to either word order or co-occurrence patterns. The performance gap appears to be task- and probe specific. We observe a low performance gap in several scenarios, the lowest being between $\mathcal{M}_N$ vs. $\mathcal{M}_3/\mathcal{M}_4$, for PTB using the both MLP and Linear probes.

[8] PTB data [Kitaev et al., 2019] is used from github.com/nikitakit/self-attentive-parser/tree/master/data.

[9] Pimentel et al. [2020a] propose computing the *Pareto Hypervolume* over all hyperparameters in each task. We did not observe a significant difference in the hypervolumes for the models, as reported in §5.5.8.

SentEval Probes

We also investigate the suite of 10 probing tasks [Conneau et al., 2018a] available in the SentEval toolkit [Conneau and Kiela, 2018]. This suite contains a range of semantic, syntactic and surface level tasks. Jawahar et al. [2019b] utilize this set of probing tasks to arrive at the conclusion that *"BERT embeds a rich hierarchy of linguistic signals: surface information at the bottom, syntactic information in the middle, semantic information at the top"*. We re-examine this hypothesis by using the same probing method and comparing against models trained with random word order.

Training setup. We run the probes on the final layer of each of our pre-trained models for three seeds, while keeping the encoder frozen. SentEval trains probes on top of fixed representations individually for each task. We follow the recommended setup and run grid search over the following hyperparams: number of hidden layer dimensions ($[0, 50, 100, 200]$), dropout ($[0, 0.1, 0.2]$), 4 epochs, 64 batch size. We select the best performance based on the dev set, and report the test set accuracy.

Results. We provide the results in Table 5.4. The $\mathcal{M}_{\mathrm{N}}$ pre-trained model scores better than the unnatural word order models for only one out of five semantic tasks and in none of the lexical tasks. However, $\mathcal{M}_{\mathrm{N}}$ does score higher for two out of three syntactic tasks. Even for these two syntactic tasks, the gap among $\mathcal{M}_{\mathrm{UG}}$ and $\mathcal{M}_{\mathrm{N}}$ is much higher than $\mathcal{M}_1$ and $\mathcal{M}_{\mathrm{N}}$. These results show that while natural word order is useful for at least some probing tasks, the distributional prior of randomized models alone is enough to achieve a reasonably high accuracy on syntax sensitive probing.

Non-Parametric Probing

How to probe effectively with parametric probes is a matter of much recent debate [Hall Maudslay et al., 2020, Belinkov, 2021]. From our results so far, it is unclear whether parametric probing meaningfully distinguishes models trained with corrupted word order from those trained with normal orders. Thus, we also investigate non-parametric probes [Linzen et al., 2016, Marvin and Linzen, 2018, Gulordava et al., 2018a] using the formulation of Goldberg [2019] and Wolf [2019a].

We consider a set of non-parametric probes that use a range of sentences varying in their linguistic properties. For each, the objective is for a pre-trained model to provide higher probability to a grammatically correct word than to an incorrect one. Since both the correct and incorrect options occupy the same sentential position, we call them "focus words". Linzen et al. [2016] use sentences from Wikipedia containing present-tense verbs, and compare the probability assigned by the encoder to plural vs. singular forms of the verb; they focus on sentences containing at least one noun between the verb and its subject, known as "agreement attractors." Gulordava et al. [2018a] instead replace focus words with random substitutes from the same part-of-speech and inflection. Finally, Marvin and Linzen [2018] construct minimal pairs of grammatical and ungrammatical sentences, and compare the model's probability for the words that differ.

Setup. In our experiments, we mask the focus words in the stimuli and compute the probability of the correct and incorrect token respectively. To handle Byte-Pair Encoding (BPE), we use the WordPiece [Wu et al., 2016] tokens prepended with a space. We observe that the Linzen et al. [2016] and Gulordava et al. [2018a] datasets are skewed towards singular focus words, which could disproportionately help weaker models that just happen to assign more probability mass to singular focus words. To counter this, we balance these datasets to have an equal number of singular and plural focus words by upsampling, and report the aggregated and balanced results in Table 5.5 (see §5.5.9 for more detailed results). We verify our experiments by using three pre-trained models with different seeds for each model configuration.

Results. We observe for the Linzen et al. [2016] and Marvin and Linzen [2018] datasets that the gap between the $\mathcal{M}_{\mathrm{N}}$ and randomization models is relatively large. The Gulordava et al. [2018a] dataset shows a smaller gap between $\mathcal{M}_{\mathrm{N}}$ and the randomization models. While some randomization models (e.g., $\mathcal{M}_2$, $\mathcal{M}_3$, and $\mathcal{M}_4$) performed quite similarly to $\mathcal{M}_{\mathrm{N}}$ according to the parametric probes, they all are markedly worse than $\mathcal{M}_{\mathrm{N}}$ according to the non-parametric ones. This suggests that non-parametric probes identify certain syntax-related modeling failures that parametric ones do not.

Model	Linzen et al. [2016] *	Gulordava et al. [2018b] *	Marvin and Linzen [2018]
$\mathcal{M}_{\mathrm{N}}$	91.17 +/- 2.6	68.66 +/- 11.6	88.05 +/- 6.5
$\mathcal{M}_4$	66.93 +/- 3.2	69.47 +/- 4.9	70.66 +/- 12.5
$\mathcal{M}_3$	64.60 +/- 2.7	66.10 +/- 5.9	73.82 +/- 15.7
$\mathcal{M}_2$	61.27 +/- 3.1	60.20 +/- 7.6	73.95 +/- 14.3
$\mathcal{M}_1$	58.96 +/- 1.8	68.10 +/- 14.4	70.69 +/- 11.6
$\mathcal{M}_{\mathrm{UG}}$	65.36 +/- 7.1	60.88 +/- 24.3	50.10 +/- 0.2

Table 5.5 Mean (and std) non-parametric probing accuracy on different datasets. * indicates rebalanced datasets, see §5.5.9 for more details.

5.5 Analysis

5.5.1 Word-order pre-training ablations

We also train further model ablations with low to high distributional priors. Following the construction of the corpus bootstrap resample, we train a model where words are drawn uniformly from BookWiki corpus, thus destroying the natural frequency distribution ($\mathcal{M}_{\mathrm{UF}}$). We further study an ablation for a high distributional prior, $\mathcal{M}_{512}$, where we shuffle words (unigram) in a buffer created with joining multiple sentences such that maximum token length of the buffer is 512. This ablation—which is similar to the paragraph word shuffle condition in Gauthier and Levy [2019]—will allow us to study the effect of unigram shuffling in a window larger than the one for $\mathcal{M}_1$. Buffer size is chosen to be 512 because BERT/RoBERTa is typically trained with that maximum sequence length.

We observe dev set results on the GLUE benchmark of these ablations, along with

Model	QNLI	RTE	QQP	SST-2	MRPC	PAWS	MNLI-m/mm	CoLA
$\mathcal{M}_{\mathrm{N}}$	92.45 +/- 0.2	73.62 +/- 3.1	91.25 +/- 0.1	93.75 +/- 0.4	89.09 +/- 0.9	94.49 +/- 0.2	86.08 +/- 0.2 / 85.4 +/- 0.2	52.45 +/- 21.2
$\mathcal{M}_{4}$	91.65 +/- 0.1	70.94 +/- 1.2	91.39 +/- 0.1	92.46 +/- 0.3	86.90 +/- 0.3	94.26 +/- 0.2	83.79 +/- 0.2 / 83.94 +/- 0.3	35.25 +/- 32.2
$\mathcal{M}_{3}$	91.56 +/- 0.4	69.75 +/- 2.8	91.22 +/- 0.1	91.97 +/- 0.5	86.22 +/- 0.8	94.03 +/- 0.1	83.83 +/- 0.2 / 83.71 +/- 0.1	40.78 +/- 23.0
$\mathcal{M}_{2}$	90.51 +/- 0.1	70.00 +/- 2.5	91.33 +/- 0.0	91.78 +/- 0.3	85.90 +/- 1.2	93.53 +/- 0.3	83.45 +/- 0.3 / 83.54 +/- 0.3	50.83 +/- 5.80
$\mathcal{M}_{1}$	89.05 +/- 0.2	68.48 +/- 2.5	91.01 +/- 0.0	90.41 +/- 0.4	86.06 +/- 0.8	89.69 +/- 0.6	82.64 +/- 0.1 / 82.67 +/- 0.2	31.08 +/- 10.0
$\mathcal{M}_{512}$	84.97 +/- 0.3	56.09 +/- 0.6	90.15 +/- 0.1	86.11 +/- 0.7	79.41 +/- 0.6	77.3 +/- 12.63	77.58 +/- 0.3 / 77.89 +/- 0.4	12.54 +/- 5.57
$\mathcal{M}_{\mathrm{NP}}$	77.59 +/- 0.3	54.78 +/- 2.2	87.78 +/- 0.4	83.21 +/- 0.6	72.78 +/- 1.6	57.22 +/- 1.2	63.35 +/- 0.4 / 63.63 +/- 0.2	2.37 +/- 3.20
$\mathcal{M}_{\mathrm{UF}}$	77.69 +/- 0.4	53.84 +/- 0.6	85.92 +/- 0.1	84.00 +/- 0.6	71.35 +/- 0.8	58.43 +/- 0.3	72.10 +/- 0.4 / 72.58 +/- 0.4	8.89 +/- 1.40
$\mathcal{M}_{\mathrm{UG}}$	66.94 +/- 9.2	53.70 +/- 1.0	85.57 +/- 0.1	83.17 +/- 1.5	70.57 +/- 0.7	58.59 +/- 0.3	71.93 +/- 0.2 / 71.33 +/- 0.5	0.92 +/- 2.10
$\mathcal{M}_{\mathrm{RI}}$	62.17 +/- 0.4	52.97 +/- 0.2	81.53 +/- 0.2	82.0 +/- 0.7	70.32 +/- 1.5	56.62 +/- 0.0	65.70 +/- 0.2 / 65.75 +/- 0.3	8.06 +/- 1.60

Table 5.6 GLUE and PAWS-Wiki dev set results on different ablations of the RoBERTa (base) models, trained on variants of the BookWiki corpus (with mean and std dev). The top row is the original model, the middle half contains the sentence randomization models, and the bottom half contains the ablations.

Model	RTE	MRPC	SST-2	CoLA	QQP	QNLI	MNLI	PAWS
$\mathcal{M}_{1}$	68.48	85.97	90.41	31.07	91.01	89.05	82.64	89.69
$\mathcal{M}_{1}{}^{*}$	68.41	85.75	90.17	50.14	91.02	89.50	82.92	91.99

Table 5.7 Reconstruction experiments on shuffled word order sentences by fixing the same seed for every sentence ($\mathcal{M}_{1}$) and having different seed for different shards of the corpus ($\mathcal{M}_{1}{}^{*}$). We observe minimal difference in the downstream GLUE and PAWS scores.

baselines $\mathcal{M}_{\mathrm{UG}}$, $\mathcal{M}_{\mathrm{RI}}$ and $\mathcal{M}_{\mathrm{NP}}$ and random shuffles in Table 5.6 and Figure 5.3. We observe that $\mathcal{M}_{512}$ exhibits worse overall scores than $\mathcal{M}_{1}$, however it is still significantly better than $\mathcal{M}_{\mathrm{NP}}$ or $\mathcal{M}_{\mathrm{UG}}$ baselines. We observe that destroying the natural frequency distribution of words ($\mathcal{M}_{\mathrm{UF}}$) yields comparable or slightly better results compared to random corpus model $\mathcal{M}_{\mathrm{UG}}$. This result shows that merely replicating the natural distribution of words without any context is not useful for the model to learn. These results indicate that at least some form of distributional prior is required for MLM-based models to learn a good downstream representation.

One might argue that the superior results displayed by the unnatural models is due to the ability of RoBERTa to "reconstruct" the natural word order from shuffled sentences. The data generation algorithm, $\mathcal{F}_i$ requires a seed t for every sentence. In

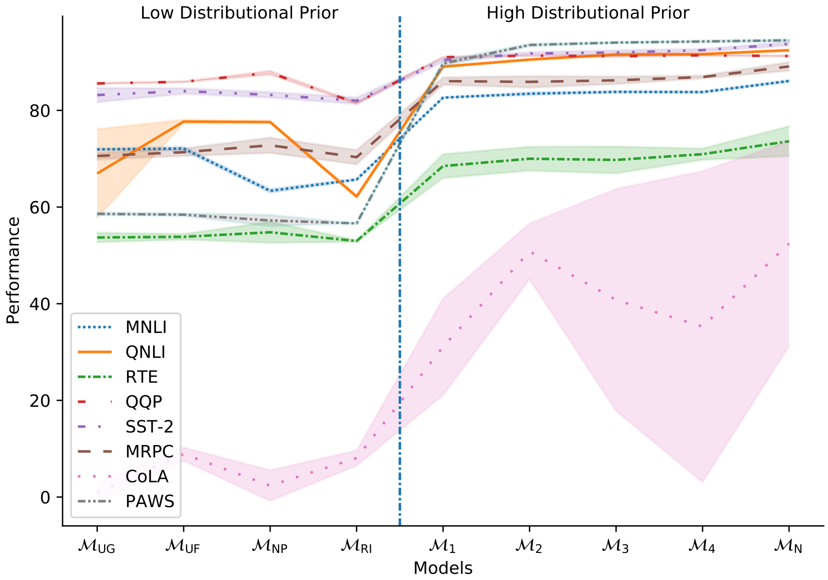


Figure 5.3 GLUE results on various model ablations using BookWiki corpus.

our experiments, we had set the same seed for every sentence in the corpus to ensure reproducibility. However, it could be problematic if the sentences of the same length are permuted with the same seed, which could be easier for the model to “reconstruct” the natural word order to learn the necessary syntax. We tested this hypothesis by constructing a new corpus with different seeds for every sentence in every shard in the corpus (1/5th of BookWiki corpus is typically referred to as a *shard* for computational purposes), to build the model $\mathcal{M}_1{}^*$. We observe that there is minimal difference in the raw numbers among $\mathcal{M}_1$ and $\mathcal{M}_1{}^*$ for most of the tasks (Table 5.7) (with the exception of CoLA which performs similar to $\mathcal{M}_2$ possibly due to a difference in initialization). This result consequently proves that even with same seed, it is difficult for the model to just reconstruct the unnatural sentences during pre-training.

Model	QNLI	RTE	QQP	SST-2	MRPC	CoLA	PAWS	MNLI
$\mathcal{M}_1$	3.70	7.04	0.26	3.58	3.42	40.74	5.12	3.62
$\mathcal{M}_2$	2.11	4.95	-0.09	2.12	3.61	3.09	9.06	2.63
$\mathcal{M}_3$	0.97	5.30	0.03	1.91	3.24	22.25	0.49	2.31
$\mathcal{M}_4$	0.87	3.67	-0.15	1.39	2.47	32.79	0.25	2.19
$\mathcal{M}_{\text{UG}}$	27.74	27.25	6.26	11.35	20.91	98.24	38.20	16.56
$\mathcal{M}_{\text{NP}}$	16.16	25.77	3.83	11.30	18.42	95.48	39.66	26.10

Table 5.8 $\Delta_{\{D_i\}}(\mathcal{T})$, scaled by a factor of 100 for GLUE and PAWS tasks.

5.5.2 Measuring Relative difference

In this section, we further measure the difference in downstream task performance reported in §5.4.1 using as a metric the *relative difference*. Let us denote the downstream task performance as $\mathcal{A}(\mathcal{T}|D)$, where $\mathcal{T}$ is the task and D is the pre-trained model. We primarily aim to evaluate the relative performance gap, i.e. how much the performance differs between our natural and unnatural models. Thus, we define the *Relative Difference* ($\Delta_{\{D\}}(\mathcal{T})$):

$$\Delta_{\{D\}}(\mathcal{T}) = \frac{\mathcal{A}(\mathcal{T}|OR) - \mathcal{A}(\mathcal{T}|D))}{\mathcal{A}(\mathcal{T}|OR) - \mathcal{A}(\mathcal{T}|\emptyset)}, \tag{5.1}$$

where $\mathcal{A}(\mathcal{T}|\emptyset)$ is the random performance on the task $\mathcal{T}$ (0.33 for MNLI, 0 for CoLA, and 0.5 for rest) $\Delta_{\{D\}}(\mathcal{T}) \to 0$ when the performance of a pre-trained model reaches that of the pre-trained model trained with natural word order.

We observe the relative difference on the tasks in Table 5.8. CoLA has the largest $\Delta_{\{D\}}(\mathcal{T})$ among all tasks, suggesting the expected high word order reliance. $\Delta_{\{D\}}(\mathcal{T})$ is lowest for QQP.

name	fine-tune-train	fine-tune-eval	MNLI	QNLI	RTE	CoLA	MRPC	SST-2	PAWS
$\mathcal{M}_{\text{N}}$	natural	natural	86.08 +/- 0.15	92.45 +/- 0.24	73.62 +/- 3.09	52.44 +/- 21.22	89.09 +/- 0.88	93.75 +/- 0.44	94.49 +/- 0.18
	natural	shuffled	68.11 +/- 0.52	81.08 +/- 0.38	56.72 +/- 3.29	4.77 +/- 1.82	75.94 +/- 1.01	80.78 +/- 0.37	62.22 +/- 0.09
	shuffled	natural	82.99 +/- 0.16	89.32 +/- 0.23	57.9 +/- 4.71	0.0 +/- 0.0	79.71 +/- 2.57	89.12 +/- 0.5	72.03 +/- 13.79
	shuffled	shuffled	79.96 +/- 0.1	87.51 +/- 0.09	59.07 +/- 3.2	1.4 +/- 2.17	79.17 +/- 0.35	86.11 +/- 0.5	65.15 +/- 0.48
$\mathcal{M}_1$	natural	natural	82.64 +/- 0.15	89.05 +/- 0.15	68.48 +/- 2.51	31.07 +/- 9.97	85.97 +/- 0.89	90.41 +/- 0.43	89.69 +/- 0.59
	natural	shuffled	76.67 +/- 0.34	87.21 +/- 0.17	65.8 +/- 6.11	23.06 +/- 5.3	81.84 +/- 0.43	83.94 +/- 0.33	62.86 +/- 0.19
	shuffled	natural	79.87 +/- 0.1	87.81 +/- 0.36	65.65 +/- 2.33	24.53 +/- 13.63	82.51 +/- 0.82	86.45 +/- 0.41	73.34 +/- 6.88
	shuffled	shuffled	79.75 +/- 0.0	88.21 +/- 0.24	64.88 +/- 6.32	22.43 +/- 10.79	82.65 +/- 0.42	86.25 +/- 0.4	63.15 +/- 2.2
$\mathcal{M}_{\text{UG}}$	natural	natural	71.93 +/- 0.21	66.94 +/- 9.21	53.7 +/- 1.01	0.92 +/- 2.06	70.57 +/- 0.66	83.17 +/- 1.5	58.59 +/- 0.33
	natural	shuffled	62.27 +/- 0.57	63.13 +/- 7.13	52.42 +/- 2.77	0.09 +/- 0.21	70.56 +/- 0.33	79.41 +/- 0.63	56.91 +/- 0.16
	shuffled	natural	67.62 +/- 0.3	66.49 +/- 0.49	52.17 +/- 1.26	0.0 +/- 0.0	70.37 +/- 0.93	79.93 +/- 1.01	57.59 +/- 0.29
	shuffled	shuffled	67.02 +/- 0.33	66.24 +/- 0.33	53.44 +/- 0.53	0.08 +/- 0.18	70.28 +/- 0.62	80.05 +/- 0.4	57.38 +/- 0.16

Table 5.9 Fine-tuning evaluation by varying different sources of word order (with mean and std dev). We vary the word order contained in the pre-trained model ($\mathcal{M}_{\text{N}}$,$\mathcal{M}_1$,$\mathcal{M}_{\text{UG}}$); in fine-tuning training set (natural and shuffled); and in fine-tuning evaluation (natural and shuffled). Here, *shuffled* corresponds to unigram shuffling of words in the input. In case of fine-tune evaluation containing shuffled input, we evaluate on a sample of 100 unigram permutations for each data point in the dev set of the corresponding task.

5.5.3 Fine-tuning with randomized data

We perform additional experiments using the fine-tuned models from §5.4.1. Specifically, we construct unigram randomized train and test sets (denoted as *shuffled*) of a subset of tasks to evaluate whether models fine-tuned on natural or unnatural task data (having natural or unnatural pre-training prior) are able to understand unnatural data during testing. In the previous chapter (§4.4.1), we observed for MNLI there exists at least one permutation for many examples which can be predicted correctly by the model. However, we also observed that every sentence can have many permutations which cannot be predicted correctly as well (§4.7). Similarly in this section we construct 100 permutations for each example in the dev set for each task to capture the overall accuracy.

Concretely, we use $\mathcal{M}_{\text{N}}$, $\mathcal{M}_1$ and $\mathcal{M}_{\text{UG}}$ as our pre-trained representations (trained with natural, unigram sentence shuffle and corpus shuffle data respectively) and evaluate the effect of training and evaluation on natural and unnatural data in Table 5.9.

We observe that all models perform poorly on the *shuffled* test set, compared to natural evaluation. However, interestingly, models have a slight advantage with a unigram randomized prior ($\mathcal{M}_1$), with CoLA having the biggest performance gain. PAWS task suffers the biggest drop in performance (from 94.49 to 62.22) but the lowest gain in $\mathcal{M}_1$, confirming our conclusion from §5.4.1 that most of the word order information necessary for PAWS is learned from the task itself.

Furthermore, training on shuffled data surprisingly leads to high performance on natural data for $\mathcal{M}_{\mathrm{N}}$ in case of several tasks, the effect being weakest in case of CoLA and PAWS. This suggests that for tasks other than CoLA and PAWS, spurious correlations are leveraged by the models during fine-tuning (cf. Gururangan et al. 2018, Poliak et al. 2018, Tsuchiya 2018). We also observe evidence of *domain matching*, where models improve their performance on evaluation on shuffled data when the training data source is changed from natural to shuffled (for $\mathcal{M}_{\mathrm{N}}$, MNLI shuffled evaluation improves from 68.11 to 79.96 just by changing the training corpus from natural to shuffled). We observe this behavior consistently for all tasks with all pre-trained representations.

5.5.4 Dependency parsing using Second order Tree CRF Neural Dependency Parser

Model	UD EWT		PTB	
	UAS	LAS	UAS	LAS
$\mathcal{M}_{\mathrm{N}}$	90.92%	87.87%	95.42%	93.75%
$\mathcal{M}_1$	91.18%	88.19%	95.90%	94.35%
$\mathcal{M}_2$	91.11%	88.12%	95.74%	94.16%
$\mathcal{M}_3$	91.05%	87.94%	95.73%	94.14%
$\mathcal{M}_4$	90.88%	87.78%	95.77%	94.16%
$\mathcal{M}_{\mathrm{UG}}$	90.47%	87.42%	95.81%	94.28%

Table 5.10 Unlabeled Attachment Score (UAS) on Dependency parsing task on two datasets, UD EWT and PTB, using the Second order Tree CRF Neural Dependency Parser Zhang et al. [2020]

We also conduct extensive experiments with Second Order Tree CRF Neural Dependency parser from Zhang et al. [2020], using their provided codebase.[10] We report the results on UD EWT and PTB corpus in Table 5.10. Strangely enough, we find the gap to be even smaller across the different randomization models, even for some cases the performance on R_1 improves over OR. We suspect this result is due to two reasons: **(a)** Due to the presence of the complex Biaffine Dependency parser consisting of multiple LSTMs and individual MLP heads for each dependency arc (left and right), the majority of learning of the task is done by the parser itself; **(b)** Zhang et al. [2020] downsample the BERT representation to 100 dimensions which is then combined with the learned LSTM representations, thereby minimizing the impact of the pre-trained representations. Our hypothesis is confirmed by the published results of Zhang et al. [2020] on the Github repository, which shows a minimal gap between models with or without BERT.

5.5.5 Perplexity analysis

We measure perplexity of various pre-trained randomization models on text that is randomized using the same function $\mathcal{F}$. Conventional language models compute the perplexity of a sentence S by using past tokens ($S_{<t} = (w_1, w_2, \dots, w_{t-1})$) and the application of chain rule ($\sum_{t=1}^{|S|} \log P_{LM}(w_t|S_{t-1})$). However, this formulation is not defined for MLM, as a word is predicted using the entire sentence as a context. Following Salazar et al. [2020], we measure *Pseudo-Perplexity*, i.e., given a sentence S, we compute the log-probability of the missing word in S by iteratively masking out the specific word, and computing the average log-probability per word in S:

$$\texttt{PLL}(S) = \frac{1}{|S|} \sum_{w \in S} \log P_{\texttt{MLM}}(w|S_{\setminus w}; \theta) \tag{5.2}$$

We bootstrap the `PLL` score of a test corpus T by drawing 100 samples five times

[10] https://github.com/yzhangcs/parser

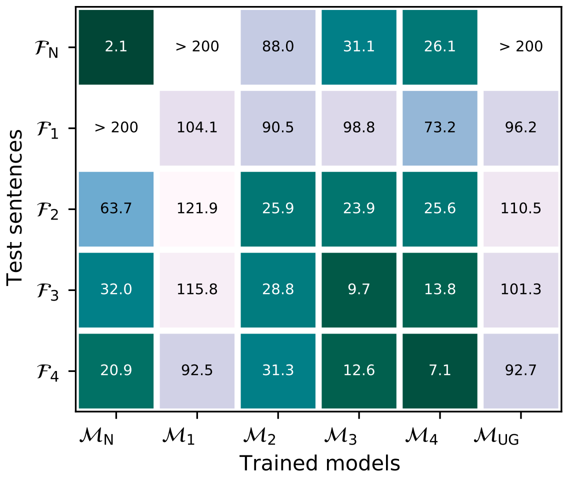


Figure 5.4 BPPL scores per model per test scenario.

with replacement. We also similarly compute the bootstrap perplexity following Salazar et al.:

$$\texttt{BPLL}_T = \exp(-\frac{1}{N}\sum_{S\in W}\texttt{PLL}(S)), \tag{5.3}$$

where W is the combined bootstrap sample containing N sentences drawn with replacement from T. We compute this score on 6 pre-trained models, over four randomization schemes on the bootstrapped sample W (i.e., we use the same n-gram randomization function $\mathcal{F}_i$). Thus, we obtain a 5x6 matrix of $\texttt{BPLL}$ scores, which we plot in Figure 5.4.

We observe that the pre-trained model $\mathcal{M}_\text{N}$ has the lowest perplexity on the sentences with natural word order. Pre-trained models with random word order exhibit significantly higher perplexity than the normal word order sentences (top row). With the exception of $\mathcal{M}_1$, the models pre-trained on randomized data ($\mathcal{M}_2$, $\mathcal{M}_3$ and $\mathcal{M}_4$)

all display the lowest perplexity for their respective $n = 2, 3, 4$ randomizations. These results indicate that the models retain and detect the specific word order for which they were trained.

5.5.6 The usefulness of word order

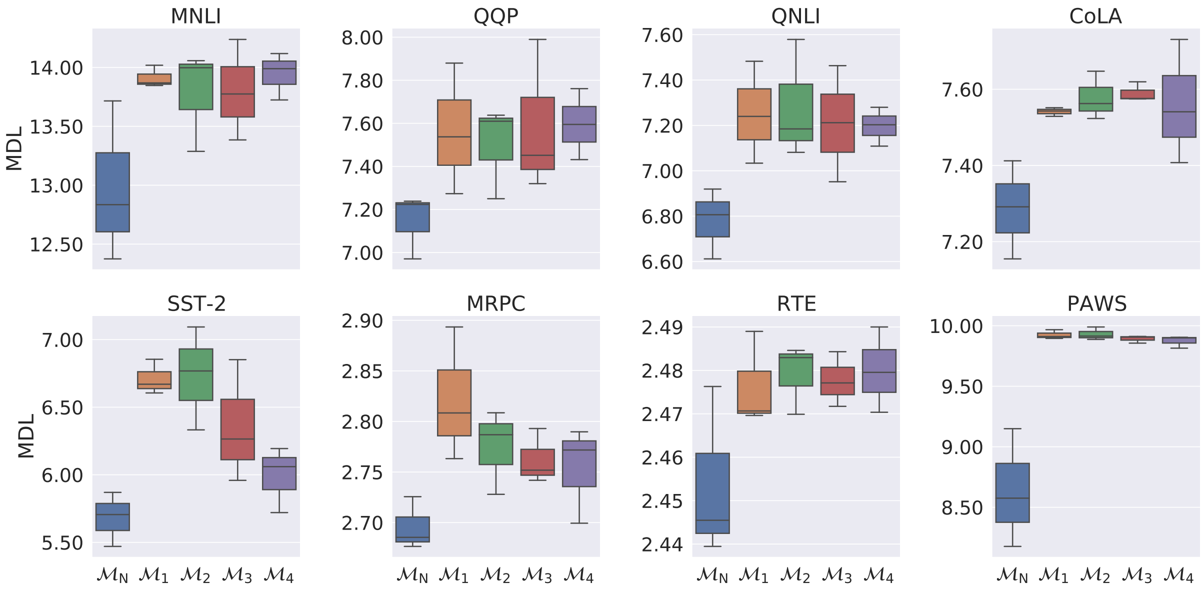


Figure 5.5 Rissanen Data Analysis [Perez et al., 2021b] on the GLUE benchmark and PAWS datasets. The lower minimum description length (MDL, measured in kilobits), the better the learning ability of the model.

The results in §5.4.1 suggest that, with proper fine-tuning, an unnaturally trained model can reach a level of performance comparable to that of a naturally pre-trained model. However, we want to understand whether natural word order pre-training offers any advantage during the early stages of fine-tuning. Towards that end, we turn to compute the Minimum Description Length [MDL; Rissanen, 1984]. MDL is designed to characterize the complexity of data as the length of the shortest program required to generate it. Thus, the length of the minimum description (in bits) should provide a fair estimate of how much word order is useful for fine-tuning in a few-shot setting. Specifically, we leverage the Rissanen Data Analysis (RDA) framework from

Perez et al. [2021b] to evaluate the MDL of pre-trained models on our set of downstream tasks. Under mild assumptions, if a pre-trained model θ_1 is useful for solving a particular task T over θ_2, then the MDL in bits obtained by using θ_1 should be shorter than θ_2. We follow the experimental setup of Perez et al. to compute the MDL on several tasks using $\theta = \{\mathcal{M}_{\text{N}},\mathcal{M}_1,\mathcal{M}_2,\mathcal{M}_3,\mathcal{M}_4\}$, over three seeds and on three epochs of training. Concretely, RDA involves sampling 9 blocks of data from the dataset at random, where the size of each block is increased monotonically, training on 8 blocks while evaluating the model's loss (or *codelength*) on the ninth. The minimum number of data samples in the smallest block is set at 64, while the largest number of data samples used in the last block is 10,000.

We observe that the value of MDL is consistently lowest for naturally pre-trained data (Figure 5.5). For purportedly word order reliant datasets such as RTE, CoLA and PAWS, the gap between the MDL scores among the natural and unnatural models is high. PAWS, specifically, has the largest advantage in the beginning of optimization, however with more fine-tuning, the model re-learns correct word order (§5.4.1). The present analyses, when taken in conjunction with our main results in §5.4.1, suggest that fine-tuning on large training datasets with complex classifiers in the pursuit of state-of-the-art results has mostly nullified the impact of word order in the pre-trained representations. Few shot [Bansal et al., 2020] and few sample [Zhang et al., 2021] learning and evaluation could potentially require more word order signal, thereby encouraging the model to leverage its own learned syntax better.

5.5.7 At what point do models learn word order during pre-training?

Results from §5.4.1 beg the question: when, if at all, during pre-training does a model learn the natural word order? We aim to answer that question by comparing downstream task performance of RoBERTa base on intermediate checkpoints with that of the random word order pretrained models. The idea is to find the point during pre-

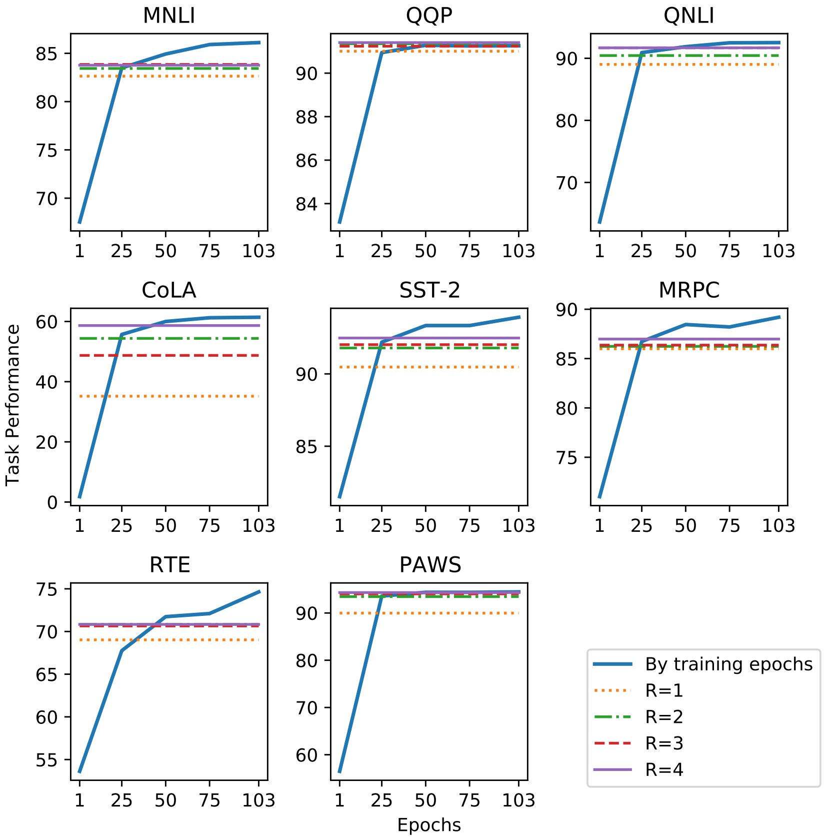


Figure 5.6 Comparison among GLUE task performance from different steps in pre-training of RoBERTa on BookWiki Corpus.

training on natural corpus at which the model exceeds the task performance of the random pre-training model.

Performance on all tasks (Figure 5.6) increases rapidly during the first 20-25 epochs of pre-training. For some tasks, the word order information only helps after 30-50 pre-training epochs.

Model	UD EWT		PTB	
	MLP	Linear	MLP	Linear
$\mathcal{M}_{\text{N}}$	93.74 +/- 0.15	88.82 +/- 0.42	97.07 +/- 0.38	93.1 +/- 0.65
$\mathcal{M}_1$	88.60 +/- 3.43	80.76 +/- 3.38	95.33 +/- 0.37	87.83 +/- 1.86
$\mathcal{M}_2$	93.39 +/- 0.45	87.58 +/- 1.06	96.96 +/- 0.15	91.80 +/- 0.50
$\mathcal{M}_3$	92.89 +/- 0.65	86.78 +/- 1.32	97.03 +/- 0.13	91.70 +/- 0.70
$\mathcal{M}_4$	92.83 +/- 0.61	87.23 +/- 0.77	96.96 +/- 0.12	92.08 +/- 0.39
$\mathcal{M}_{\text{UG}}$	89.10 +/- 0.21	79.75 +/- 0.5	94.12 +/- 0.01	84.15 +/- 0.51

Table 5.11 Accuracy on the part-of-speech labelling task (POS) on two datasets, UD EWT and PTB, using the Pareto Probing framework Pimentel et al. [2020b].

Model	UD EWT		PTB	
	MLP	Linear	MLP	Linear
$\mathcal{M}_{\text{N}}$	89.63 +/- 0.60	84.35 +/- 0.78	93.96 +/- 0.63	88.35 +/- 1.00
$\mathcal{M}_1$	83.55 +/- 3.31	75.26 +/- 3.08	91.10 +/- 0.38	82.34 +/- 1.37
$\mathcal{M}_2$	88.57 +/- 0.68	82.05 +/- 1.10	93.27 +/- 0.26	86.88 +/- 0.87
$\mathcal{M}_3$	88.69 +/- 1.09	82.37 +/- 1.26	93.46 +/- 0.29	87.12 +/- 0.72
$\mathcal{M}_4$	88.66 +/- 0.76	82.58 +/- 1.04	93.49 +/- 0.33	87.30 +/- 0.79
$\mathcal{M}_{\text{UG}}$	84.93 +/- 0.34	76.30 +/- 0.52	89.98 +/- 0.43	78.59 +/- 0.68

Table 5.12 Accuracy on the dependency arc labelling task (DAL) on two datasets (with mean and std dev), UD EWT and PTB, using the Pareto Probing framework Pimentel et al. [2020a].

5.5.8 More results from Syntactic Probes

We computed the Pareto Hypervolume on the dependency parsing task [Pimentel et al., 2020a]. The Pareto Hypervolume is computed as the Area Under Curve (AUC) score over all hyperparameter runs, where the models are arranged based on their complexity. We observe minimal differences in the Pareto Hypervolumes (Table 5.13) among $\mathcal{M}_{\text{N}}$ and the randomization models for both datasets.

We also investigated two "easy" tasks, Part-of-Speech tagging (POS) and Dependency Arc Labeling (DAL) from the Pareto Probing framework. For POS (Table 5.11) and DAL (Table 5.12), since these tasks are simpler than DEP, the gap between $\mathcal{M}_{\text{N}}$ and

Model	UD EWT	PTB
$\mathcal{M}_{\mathrm{N}}$	0.528 +/- 0.01	0.682 +/- 0.01
$\mathcal{M}_1$	0.489 +/- 0.03	0.648 +/- 0.01
$\mathcal{M}_2$	0.529 +/- 0.00	0.681 +/- 0.01
$\mathcal{M}_3$	0.528 +/- 0.02	0.689 +/- 0.01
$\mathcal{M}_4$	0.525 +/- 0.00	0.683 +/- 0.01
$\mathcal{M}_{\mathrm{UG}}$	0.510 +/- 0.01	0.640 +/- 0.05

Table 5.13 Pareto Hypervolume of dependency parsing task (DEP) on two datasets (with mean and std dev), UD EWT and PTB, using the Pareto Probing framework Pimentel et al. [2020b].

unnaturally pre-trained models reduces even more drastically. The gap between $\mathcal{M}_{\mathrm{N}}$ and $\mathcal{M}_1$ reduces to just 3.5 points on average for PTB in both POS and DAL.

5.5.9 Non parametric probes

Probability difference. In the original formulation [Goldberg, 2019, Wolf, 2019a], the effectiveness of each stimulus is determined by the accuracy metric, computed as the number of times the probability of the correct focus word is greater than that of the incorrect word ($P(\text{good}) > P(\text{bad})$). We observed that this metric might not be reliable per se, since the probabilities may themselves be extremely low for all tokens, even when focus word probability decreases drastically from $\mathcal{M}_{\mathrm{N}}$ to $\mathcal{M}_{\mathrm{UG}}$. Thus, we also report the mean difference of probabilities, $(\frac{1}{N}\sum_i^N P(\text{good}_i) - P(\text{bad}_i))$, scaled up by a factor of 100 for ease of observation, in Figure 5.9, Figure 5.8 and Figure 5.7. We observe the highest difference between probabilities of the correct and incorrect focus words for the model pretrained on the natural word order ($\mathcal{M}_{\mathrm{N}}$). Moreover, with each step from $\mathcal{M}_1$ to $\mathcal{M}_4$, the difference between probablities of correct and incorrect focus words increases, albeit marginally, showing that pre-trained models with fewer n-gram words perturbed capture more word order information. $\mathcal{M}_{\mathrm{UG}}$, the model with the distributional prior ablated, performs the worst, as expected.

model condition	$\mathcal{M}_{\mathrm{N}}$	$\mathcal{M}_{\mathrm{UG}}$	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$
1	93.45 (0.89) [25.04]	58.87 (0.41) [0.0]	59.96 (1.58) [0.08]	63.63 (0.6) [1.25]	64.7 (1.44) [2.79]	70.47 (1.9) [4.01]
2	92.8 (1.22) [23.8]	63.03 (1.35) [0.01]	58.22 (1.5) [0.09]	61.15 (2.07) [0.82]	63.84 (2.41) [2.09]	64.7 (1.92) [3.07]
3	87.71 (1.34) [22.03]	64.06 (3.52) [0.0]	56.69 (2.98) [0.03]	56.83 (3.63) [0.85]	61.1 (0.32) [2.02]	63.0 (3.36) [2.35]
4	92.67 (0.52) [22.16]	76.33 (1.38) [0.0]	62.33 (7.61) [0.08]	63.17 (9.09) [1.12]	69.42 (1.77) [2.1]	67.67 (7.02) [3.43]

Table 5.14 Linzen et al. [2016] stimuli results in raw accuracy. Values in parenthesis reflect the standard deviation over different seeds of pre-training. Values in square brackets indicate the mean probability difference among correct and incorrect words.

model condition	$\mathcal{M}_{\mathrm{N}}$	$\mathcal{M}_{\mathrm{UG}}$	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$
0	79.42 (5.5) [2.43]	47.83 (3.76) [-0.0]	53.67 (1.38) [0.03]	58.75 (6.38) [0.05]	63.58 (4.11) [0.14]	63.75 (3.28) [0.17]
1	72.83 (4.07) [2.55]	44.5 (0.5) [0.0]	70.83 (5.8) [0.02]	64.83 (1.76) [-0.09]	71.67 (6.71) [0.21]	71.5 (2.65) [0.61]
2	55.56 (0.0) [0.92]	88.89 (11.11) [0.0]	81.48 (12.83) [0.03]	51.85 (6.42) [0.04]	62.96 (6.42) [0.38]	74.07 (16.97) [0.61]

Table 5.15 Gulordava et al. [2018b] stimuli results in raw accuracy.Values in parenthesis reflect the standard deviation over different seeds of pre-training. Values in square brackets indicate the mean probability difference among correct and incorrect words.

Accuracy comparison. We provide the accuracy as measured by Goldberg [2019], Wolf [2019a] on the probing stimuli in Table 5.14, Table 5.15 and Table 5.16. We also highlight the difference in probability ($P(\text{good}) - P(\text{bad})$) in the table to provide a more accurate picture. All experiments were conducted on three pre-trained seeds for each model in our set of models. However, the low token probabilities in $\mathcal{M}_{\mathrm{UG}}$ tend to present unreliable scores. For example, in the case of Gulordava et al. [2018a] stimuli, unnatural models provide better scores compared to the natural model. We also observe for the Linzen et al. [2016] stimuli that the results on model condition 4 (number of attractors) are surprisingly high for $\mathcal{M}_{\mathrm{UG}}$ whereas the individual token probabilities are lowest. We believe these inconsistencies stem from extremely low token probabilities themselves.

Balancing datasets on inflection by upsampling. The stimuli datasets of Linzen et al. [2016] and Gulordava et al. [2018a] turned out to be heavily skewed towards words where singular was the correct inflection (as opposed to plural). This dataset imbalance caused the weak models (such as $\mathcal{M}_{\mathrm{UG}}$) to have surprisingly high scores - the weak models were consistently providing higher probability for the singular inflection (Table 5.17). We upsample for both datasets, balancing the frequency of correct singular and plural inflections. We compute the upsampling number to the next multiple of 100 of the count of original singular inflections. For example, in condition 4 of Linzen et al. [2016] dataset, we upsample both S and P to 300 rows each. This type of balancing via upsampling largely alleviated the inconsistencies we observed, and might prove to be useful when evaluating other models on these datasets in future.

Model condition	$\mathcal{M}_{\mathrm{N}}$	$\mathcal{M}_{\mathrm{UG}}$	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$
AOR	89.98 (1.96) [29.16]	50.0 (0.01) [0.0]	60.17 (1.61) [1.7]	66.61 (7.1) [3.87]	63.57 (2.39) [2.45]	61.26 (4.91) [1.16]
AOR-T	77.4 (7.74) [13.84]	50.0 (0.0) [0.0]	78.88 (0.64) [0.15]	52.17 (2.14) [0.43]	48.85 (3.8) [0.25]	57.06 (3.49) [0.55]
APP	89.94 (4.16) [27.06]	50.01 (0.02) [-0.0]	70.34 (1.9) [0.68]	53.61 (3.3) [0.2]	53.03 (1.75) [0.79]	60.6 (4.41) [0.91]
ARC	85.06 (5.92) [1.2]	50.05 (0.08) [-0.0]	62.39 (1.91) [0.11]	74.57 (5.99) [0.07]	67.55 (3.84) [0.07]	62.88 (3.45) [0.03]
ASR	87.19 (3.58) [26.18]	50.0 (0.0) [-0.0]	78.55 (10.01) [1.91]	81.73 (5.1) [2.91]	62.8 (0.35) [1.38]	67.23 (6.82) [1.48]
IOR	89.83 (3.33) [0.4]	50.55 (0.95) [-0.0]	56.28 (2.66) [0.01]	58.96 (4.28) [0.04]	70.49 (2.2) [0.04]	62.82 (8.51) [0.08]
IOR-T	74.05 (8.26) [0.2]	50.61 (1.05) [-0.0]	52.63 (2.07) [0.01]	57.35 (4.88) [0.01]	61.85 (4.75) [0.01]	55.16 (6.59) [0.03]
ISC	85.87 (9.6) [5.27]	50.0 (0.0) [0.0]	67.85 (2.62) [0.07]	82.66 (9.43) [0.0]	77.69 (4.51) [0.31]	68.65 (5.71) [0.06]
LVC	93.0 (0.75) [26.58]	49.92 (0.14) [-0.0]	70.42 (6.79) [0.1]	87.5 (7.26) [1.44]	85.42 (3.84) [0.63]	81.08 (5.13) [0.66]
SCM	88.6 (3.49) [5.72]	50.0 (0.0) [0.02]	63.73 (7.94) [0.12]	82.12 (0.92) [0.17]	86.44 (3.67) [2.62]	80.27 (2.46) [0.26]
SRX	91.0 (6.07) [2.72]	50.0 (0.0) [0.0]	88.0 (10.11) [0.1]	92.25 (10.27) [0.32]	94.25 (5.02) [0.02]	91.0 (6.5) [3.07]
SVA	95.33 (7.23) [31.35]	50.0 (0.0) [0.02]	86.0 (5.29) [0.57]	85.17 (12.87) [3.17]	94.67 (5.25) [6.15]	88.83 (9.57) [30.57]
SVC	97.54 (1.58) [19.27]	50.0 (0.0) [-0.0]	83.58 (4.58) [0.84]	83.71 (8.78) [1.17]	93.29 (7.4) [6.39]	81.04 (3.66) [11.07]

Table 5.16 Marvin and Linzen [2018] stimuli results in raw accuracy. Values in parenthesis reflect the standard deviation over different seeds of pretraining. Values in square brackets indicate the mean probability difference among correct and incorrect words. Abbreviations: Simple Verb Agreement (SVA), In a sentential complement (SCM), Short VP Coordination (SVC), Long VP Coordination (LVC), Across a prepositional phrase (APP), Across a subject relative clause (ASR), Across an object relative clause (AOR), Across an object relative (no *that*) (AOR-T), In an object relative clause (IOR), In an object relative clause (no *that*) (IOR-T), Simple Reflexive (SRX), In a sentential complement (ISC), Across a relative clause (ARC), Simple NPI (SNP).

Model condition	$\mathcal{M}_{\mathrm{N}}$	$\mathcal{M}_{\mathrm{UG}}$	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	S/P
1	94.04 (0.8)	62.64 (0.5)	62.18 (1.33)	64.91 (0.14)	65.35 (1.78)	70.88 (1.88)	14011 / 10112
2	93.28 (0.94)	71.24 (0.85)	63.03 (1.69)	62.92 (2.57)	65.25 (3.13)	65.61 (2.35)	3120 / 1312
3	89.1 (0.58)	74.05 (1.85)	62.94 (3.13)	59.18 (3.32)	63.54 (1.72)	63.05 (2.0)	733 / 215
4	90.53 (0.9)	80.03 (0.59)	63.16 (4.83)	63.94 (6.92)	66.41 (3.17)	66.28 (4.64)	206 / 51

Table 5.17 Linzen et al. [2016] stimuli results in raw accuracy on original, unbalanced data. Values in parenthesis reflect the standard deviation. S/P reflects the count of correct singular and plural focus words.

5.6 Related Work

5.6.1 Sensitivity to word order in NLU

Information order has been a topic of research in computational linguistics since Barzilay and Lee [2004] introduced the task of ranking sentence orders as an evaluation for language generation quality, an approach which was subsequently also used to evaluate readability and dialogue coherence [Barzilay and Lapata, 2008, Laban et al., 2021].

More recently, several research groups have investigated information order for words rather than sentences as an evaluation of model humanlikeness. In our previous chapter (Chapter §4), we investigate the task of natural language inference (NLI) and find high accuracy on permuted examples for different Transformer and pre-Transformer era models, across English and Chinese datasets [Hu et al., 2020a]. Gupta et al. [2021] use targeted permutations on RoBERTa-based models and show word order insensitivity across natural language inference (MNLI), paraphrase detection (QQP) and sentiment analysis tasks (SST-2). Pham et al. [2020] show insensitivity on a larger set of tasks, including the entire GLUE benchmark, and find that certain tasks in GLUE, such as CoLA and RTE are more sensitive to permutations than others. Ettinger [2020] recently observed that BERT accuracy decreases for some word order perturbed examples, but not for others. In all these prior works, models were given access to normal

	OR	R1	R2	R3	R4
1	They are commonly known as daturas, but also known as devil’s trumpets, not to be confused with angel’s trumpets, its closely related genus "Brugmansia".	be They angel’s also but trumpets, genus related devil’s as commonly closely known its daturas, trumpets, as "Brugmansia". confused with known are to not	as devil’s They genus not to trumpets, closely related "Brugmansia". are commonly trumpets, its also known known as be confused daturas, but with angel’s	"Brugmansia". related They are commonly trumpets, its closely as daturas, but known genus also known as trumpets, confused with angel’s devil’s not to be	its closely related genus They are commonly known trumpets, as trumpets, daturas, but also known as "Brugmansia". not to be confused with angel’s devil’s
2	They are also sometimes called moonflowers, jimsonweed, devil’s weed, hell’s bells, thorn-apple, and many more.	are devil’s bells, called weed, hell’s thorn-apple, and many They also more. moonflowers, jimsonweed, sometimes	more. They are hell’s bells, also sometimes and many called moonflowers, jimsonweed, devil’s weed, thorn-apple,	jimsonweed, devil’s weed, They are also thorn-apple, and many bells, more. hell’s sometimes called moonflowers,	moonflowers, They are also sometimes bells, thorn-apple, and many more. called jimsonweed, devil’s weed, hell’s
3	Its precise and natural distribution is uncertain, owing to its extensive cultivation and naturalization throughout the temperate and tropical regions of the globe.	throughout owing precise extensive temperate and naturalization and tropical of to natural is its Its distribution cultivation the globe. uncertain, regions the and	and natural distribution is tropical to its and naturalization throughout the the temperate and globe. Its precise uncertain, owing extensive cultivation regions of	uncertain, owing to Its precise and its extensive cultivation of globe. natural distribution is the the and tropical regions and naturalization throughout temperate	globe. Its precise and natural cultivation distribution the is uncertain, owing to its extensive and naturalization throughout the temperate and tropical regions of
4	Its distribution within the Americas and North Africa, however, is most likely restricted to the United States, Mexico and Southern Canada in North America, and Tunisia in Africa where the highest species diversity occurs.	distribution Mexico occurs. likely diversity North however, species most the Tunisia where in and and North Canada Southern America, highest Africa United the and in Americas Its within States, is to the restricted Africa,	and Tunisia the Americas distribution within Mexico and is most United States, Africa, however, Africa where in North Its and North in Southern Canada America, the to the likely restricted occurs. highest species diversity	likely Its highest species diversity United States, Mexico restricted to the Africa where the occurs. distribution within the and Tunisia in however, is most Americas and Southern Canada and North Africa, in North America,	Tunisia occurs. Its distribution within the Africa where the highest in restricted to the United Canada in North America, most North Africa, however, is and Americas likely diversity States, Mexico and Southern species and
5	All species of "Datura" are poisonous, especially their seeds and flowers.	seeds and species of poisonous, "Datura" their are All flowers. especially	"Datura" are especially their flowers. seeds and of All species poisonous,	especially their seeds flowers. "Datura" are poisonous, All species of and	flowers. poisonous, species of "Datura" are All especially their seeds and
6	Some South American plants formerly thought of as "Datura" are now treated as belonging to the distinct genus "Brugmansia" ("Brugmansia" differs from "Datura" in that it is woody, making shrubs or small trees, and it has pendulous flowers, rather than erect ones).	and "Datura" treated from than flowers, it small belonging woody, thought as ones). South differs Some "Brugmansia" American as are in the rather pendulous distinct making now erect "Datura" to ("Brugmansia" of formerly trees, or is it that plants genus has shrubs	"Brugmansia" ("Brugmansia" than erect pendulous genus and ones). is woody, small trees, of as the distinct flowers, rather Some South differs from American plants treated as formerly thought belonging to "Datura" in making that it "Datura" are it has now shrubs or	woody, small trees, and has pendulous flowers, as belonging to Some making shrubs or as rather than erect "Datura" are now "Brugmansia" ("Brugmansia" differs the distinct genus from "Datura" in formerly thought of it treated that it is ones). South American plants	belonging to the distinct has making Some ("Brugmansia" differs from "Datura" in are now treated as genus pendulous shrubs flowers, rather than erect or ones). "Brugmansia" that it is woody, South American plants formerly thought of as "Datura" small trees, and it
7	Other related taxa include	taxa Other include related	include Other related taxa	include Other related taxa	Other related taxa include
8	"Hyoscyamus niger", "Atropa belladonna", "Mandragora officinarum", Physalis, and many more.	and many niger", officinarum", belladonna", "Mandragora "Atropa "Hyoscyamus more. Physalis,	belladonna", "Mandragora "Hyoscyamus niger", many Physalis, and more. officinarum", "Atropa	more. Physalis, and many belladonna", "Mandragora officinarum", "Hyoscyamus niger", "Atropa	niger", more. belladonna", "Mandragora officinarum", Physalis, "Atropa many and "Hyoscyamus
9	The name "Datura" is taken from Sanskrit ’ ’thorn-apple’, ultimately from Sanskrit ’ ’white thorn-apple’ (referring to "Datura metel" of Asia).	of Asia). taken from name The "Datura" ’ is to ’thorn-apple’, Sanskrit ’ Sanskrit metel" ’white (referring from "Datura thorn-apple’ ultimately	"Datura" is taken from to ’ ’thorn-apple’, Sanskrit ’ ’white of thorn-apple’ (referring Asia). The name Sanskrit ultimately from "Datura metel"	Sanskrit ’ The name "Datura" ’thorn-apple’, ultimately from metel" Asia). is taken from of ’white (referring to "Datura Sanskrit ’ thorn-apple’	Asia). The name "Datura" is from taken of from Sanskrit ’ ’thorn-apple’, ultimately Sanskrit ’ ’white thorn-apple’ (referring to "Datura metel"
10	In the Ayurvedic text Sushruta different species of Datura are also referred to as ’ and ’.	the of also Sushruta Datura are referred to as In Ayurvedic and different species ’ text ’.	species of referred to are also Datura Sushruta different and as ’ Ayurvedic text In the ’.	as ’ and In the Ayurvedic also referred to species of Datura are text Sushruta different ’.	different In the Ayurvedic text also referred to as and Sushruta ’ species of Datura are ’.

Table 5.18 First 10 lines from the BookWiki corpus, and their respective n-gram permutations.

Model	RTE	MRPC	SST-2	CoLA	QQP	QNLI	MNLI	PAWS
$\mathcal{M}_{\texttt{N}}$	2e-05	2e-05	1e-05	2e-05	1e-05	1e-05	1e-05	2e-05
$\mathcal{M}_{\texttt{1}}$	2e-05	1e-05	1e-05	1e-05	3e-05	1e-05	2e-05	2e-05
$\mathcal{M}_{\texttt{2}}$	2e-05	2e-05	1e-05	1e-05	2e-05	1e-05	1e-05	3e-05
$\mathcal{M}_{\texttt{3}}$	3e-05	1e-05	2e-05	2e-05	3e-05	1e-05	1e-05	2e-05
$\mathcal{M}_{\texttt{4}}$	3e-05	1e-05	2e-05	2e-05	2e-05	1e-05	1e-05	2e-05
$\mathcal{M}_{\texttt{512}}$	1e-05	3e-05	2e-05	2e-05	3e-05	2e-05	3e-05	2e-05
$\mathcal{M}_{\texttt{UG}}$	2e-05	1e-05	3e-05	1e-05	3e-05	3e-05	3e-05	2e-05
$\mathcal{M}_{\texttt{UF}}$	2e-05	1e-05	3e-05	2e-05	3e-05	3e-05	3e-05	1e-05
$\mathcal{M}_{\texttt{RI}}$	1e-05	1e-05	3e-05	1e-05	1e-05	1e-05	2e-05	1e-05
$\mathcal{M}_{\texttt{NP}}$	1e-05	3e-05	2e-05	1e-05	1e-05	1e-05	1e-05	1e-05

Table 5.19 Fine-tuning hyperparam Learning rate of each model for each task in GLUE and PAWS

Model	RTE	MRPC	SST-2	CoLA	QQP	QNLI	MNLI	PAWS
$\mathcal{M}_{\texttt{N}}$	16	16	32	16	16	32	32	16
$\mathcal{M}_{\texttt{1}}$	32	32	16	32	32	16	32	16
$\mathcal{M}_{\texttt{2}}$	32	16	32	16	32	32	16	32
$\mathcal{M}_{\texttt{3}}$	32	32	16	32	32	16	32	32
$\mathcal{M}_{\texttt{4}}$	32	16	32	16	32	32	32	32
$\mathcal{M}_{\texttt{512}}$	32	16	16	32	32	16	16	16
$\mathcal{M}_{\texttt{UG}}$	16	16	16	16	32	16	16	32
$\mathcal{M}_{\texttt{UF}}$	16	32	16	16	32	16	16	16
$\mathcal{M}_{\texttt{RI}}$	16	16	32	16	16	16	32	16
$\mathcal{M}_{\texttt{NP}}$	16	32	16	16	32	16	16	16

Table 5.20 Finetuning hyperparam batch size of each model for each task in GLUE and PAWS

word order at (pre-)training time, but not at fine-tuning or test time. It was not clear whether the model acquires enough information about word order during the fine-tuning step, or whether it is ingrained in the pre-trained model. In this work, we take these investigations a step further: we show that the word order information needed for downstream tasks does not need to be provided to the model during pre-training. Since models can learn whatever word order information they do need largely from fine-tuning alone, this likely suggests that our downstream tasks don't actually require much complex word order information in the first place (cf., Glavaš and Vulić 2021).

5.6.2 Randomization ablations

Random controls have been explored in a variety of prior work. Wieting and Kiela [2019] show that random sentence encoders are surprisingly powerful baselines. Gauthier and Levy [2019] use random sentence reordering to label some tasks as "syntax-light" making them more easily decodeable from images of the brain. Shen et al. [2021] show that entire layers of MLM transformers can be randomly initialized and kept frozen throughout training without detrimental effect and that those layers perform better on some probing tasks than their frozen counterparts. Models have been found to be surprisingly robust to randomizing or cutting syntactic tree structures they were hoped to rely on Scheible and Schütze [2013], Williams et al. [2018b], and randomly permuting attention weights often induces only minimal changes in output Jain and Wallace [2019]. In computer vision, it is well known that certain architectures constitute good "deep image priors" for fine-tuning Ulyanov et al. [2018] or pruning Frankle et al. [2020], and that even randomly wired networks can perform well at image recognition Xie et al. [2019]. Here, we explore randomizing the data, rather than the model, to assess whether certain claims about which phenomena the model has learned are established in fact.

5.6.3 Synthetic pre-training

Kataoka et al. [2020] found that pre-training on synthetically generated fractals for image classification is a very strong prior for subsequent fine-tuning on real image data. In language modeling, Papadimitriou and Jurafsky [2020] train LSTMs [Hochreiter and Schmidhuber, 1997] on non-linguistic data with latent structure such as MIDI music or Java code provides better test performance on downstream tasks than a randomly initialized model. They observe that even when there is no vocabulary overlap among source and target languages, LSTM language models leverage the latent hierarchical structure of the input to obtain better performance than a random, Zipfian corpus of the same vocabulary.

5.6.4 On the utility of probing tasks

Many recent papers provide compelling evidence that BERT contains a surprising amount of syntax, semantics, and world knowledge [Giulianelli et al., 2018, Rogers et al., 2020, Lakretz et al., 2019, Jumelet et al., 2019, 2021]. Many of these works involve diagnostic classifiers [Hupkes et al., 2018] or *parametric* probes, i.e. a function atop learned representations that is optimized to find linguistic information. How well the probe learns a given signal can be seen as a proxy for linguistic knowledge encoded in the representations. However, the community is divided on many aspects of probing [Belinkov, 2021] including how complex probes should be. Many prefer *simple* linear probes over the complex ones [Alain and Bengio, 2017, Hewitt and Manning, 2019, Hall Maudslay et al., 2020]. However, complex probes with strong representational capacity are able to extract the most information from representations [Voita and Titov, 2020, Pimentel et al., 2020b, Hall Maudslay et al., 2020]. In this chapter, we follow Pimentel et al. [2020a] and use *both* simple (linear) and complex (non-linear) models, as well as "complex" tasks (dependency parsing). As an alternative to parametric probes, stimulus-based *non-parametric* probing [Linzen et al., 2016, Jumelet and Hupkes, 2018, Marvin and Linzen, 2018, Gulordava et al., 2018a, Warstadt et al., 2019a, Ettinger, 2020, Lakretz et al., 2021, Warstadt et al., 2020b,c] has been used to show that even without a learned probe, BERT can predict syntactic properties with high confidence [Goldberg, 2019, Wolf, 2019a]. We use this class of non-parametric probes to investigate RoBERTa's ability to learn word order during pre-training.

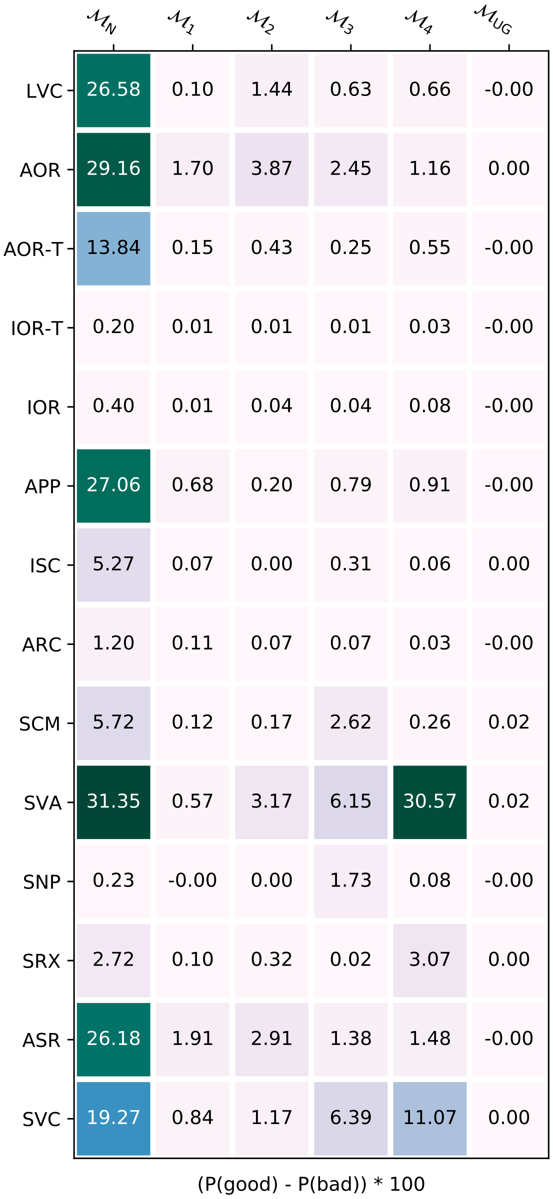
5.7 Discussion

The assumption that word order information is crucial for any classical NLP pipeline (especially for English) is deeply ingrained in our understanding of syntax itself [Chomsky, 1957]: without order, many linguistic constructs are undefined. (e.g. dependency or constituency parses would no longer be syntactic trees, what would sentences be but mere lists of words). Our fine-tuning results in §5.4.1 and parametric probing results in §5.4.2, however, suggests that MLMs do not need to rely much on word order to achieve high accuracy, bringing into question previous claims that they learn a "classical NLP pipeline."

One might ask, though, whether an NLP pipeline would really need natural word order at all: can transformers not simply learn what the correct word order is from unordered text? First, the lower non-parametric probing accuracies of the randomized models indicate that they are not able to accurately reconstruct the original word order (see also §5.5.1). But even if models were able to "unshuffle" the words under our unnatural pre-training set up, they would only be doing so based on distributional information. Models would then abductively learn only the most likely word order. While models might infer a distribution over possible orders and use that information to structure their representations [Papadimitriou et al., 2021], syntax is not about *possible* or even *the most likely* orders: it is about the *actual* order. That is, even if one concludes in the end that Transformers are able to perform word order reconstruction based on distributional information, and recover almost all downstream performance based solely on that, we ought to be a lot more careful when making claims about what our evaluation datasets are telling us.

Thus, our results seem to suggest that we may need to revisit what we mean by "linguistic structure," and perhaps subsequently acknowledge that we may not need human-like linguistic abilities for most NLP tasks. Or, our results can be interpreted as evidence that we need to develop more challenging and more comprehensive evaluations, if we genuinely want to measure linguistic abilities, however those are defined, in NLP models.

To summarize, in this chapter, we revisited the hypothesis that masked language modelling's impressive performance can be explained in part by its ability to learn classical NLP pipelines. We investigated targeted pre-training on sentences with various degrees of randomization in their word order, and observed overwhelmingly that MLM's success is most likely not due to its ability to discover syntactic and semantic mechanisms necessary for a traditional language processing pipeline during pre-training. Instead, our experiments suggest that MLM's success can largely be explained by it having learned higher-order distributional statistics that make for a useful prior for subsequent fine-tuning. These results should hopefully encourage the development of better, more challenging tasks that require sophisticated reasoning, and harder probes to narrow down what exact linguistic information is present in the representations learned by our models.



	$\mathcal{M}_N$	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$	$\mathcal{M}_4$	$\mathcal{M}_{UG}$
LVC	26.58	0.10	1.44	0.63	0.66	-0.00
AOR	29.16	1.70	3.87	2.45	1.16	0.00
AOR-T	13.84	0.15	0.43	0.25	0.55	-0.00
IOR-T	0.20	0.01	0.01	0.01	0.03	-0.00
IOR	0.40	0.01	0.04	0.04	0.08	-0.00
APP	27.06	0.68	0.20	0.79	0.91	-0.00
ISC	5.27	0.07	0.00	0.31	0.06	0.00
ARC	1.20	0.11	0.07	0.07	0.03	-0.00
SCM	5.72	0.12	0.17	2.62	0.26	0.02
SVA	31.35	0.57	3.17	6.15	30.57	0.02
SNP	0.23	-0.00	0.00	1.73	0.08	-0.00
SRX	2.72	0.10	0.32	0.02	3.07	0.00
ASR	26.18	1.91	2.91	1.38	1.48	-0.00
SVC	19.27	0.84	1.17	6.39	11.07	0.00

Figure 5.7 The difference in word probabilities for stimuli in Marvin and Linzen [2018]: Simple Verb Agreement (SVA), In a sentential complement (SCM), Short VP Coordination (SVC), Long VP Coordination (LVC), Across a prepositional phrase (APP), Across a subject relative clause (ASR), Across an object relative clause (AOR), Across an object relative (no *that*) (AOR-T), In an object relative clause (IOR), In an object relative clause (no *that*) (IOR-T), Simple Reflexive (SRX), In a sentential complement (ISC), Across a relative clause (ARC), Simple NPI (SNP).

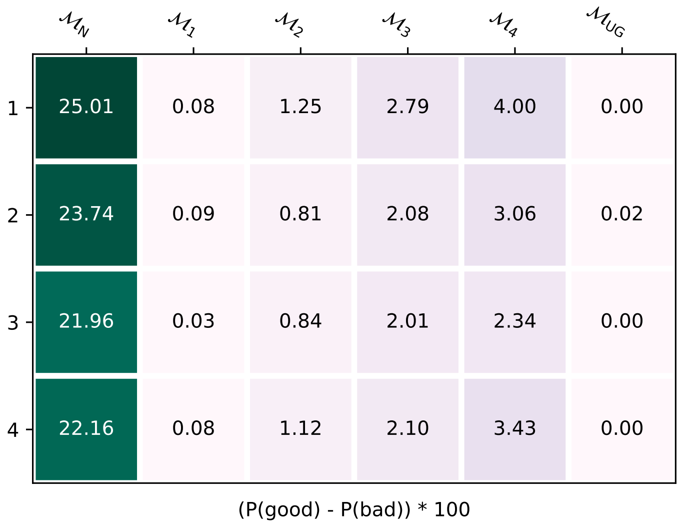


Figure 5.8 Linzen et al. [2016]

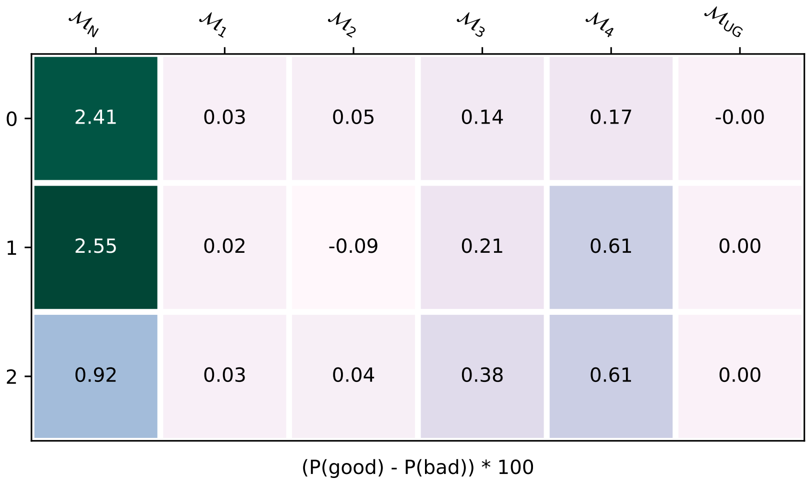


Figure 5.9 Gulordava et al. [2018a]

Chapter 6

Measuring systematic generalization by exploiting absolute positions

Recently, Transformer [Vaswani et al., 2017] language models (TLMs) have been widely used for natural language applications. Such models incorporate positional encodings: vectors encoding information about the order of words in context. Many models, such as RoBERTa [Liu et al., 2019b], GPT3 [Brown et al., 2020] and OPT [Zhang et al., 2022b], utilize *absolute* position embeddings (APEs) that directly encode absolute (linear) word order. APEs appear to contribute to the performance of such models.

However, in our previous chapters (§4 and §5), we observe models lack a sense of relative positions, as they become (in)sensitive to ablative word scrambles. Furthermore, recent studies have shown that removing APE's seem to work optimally [Haviv et al., 2022]. Thus, what precisely APEs contribute remains unclear.

It is conceivable that APEs may enable the model to handle the relative distances between words. If models were somehow learning relative position information despite using *absolute* positional embeddings, we would expect sentence encodings to be the same in most cases, regardless of where they appear in the context window. For example, the meaning of "smoking kills" should be constant in "Kim said *smoking kills*" (positions 2–3) and “It was commonly believed by most adult Americans in the 90s that *smoking kills*” (positions 13–14), despite the fact that these words appear in different absolute positions. Given this, our central question is: do APEs enable the model to learn the relative distances between the words in a sentence?

Prior work has attempted to explore the consequences of APEs using probing methods [Wang et al., 2021]. APEs have been found to not capture the meaning of absolute or relative positions [Wang and Chen, 2020]. APEs have also been found to bias model output with positional artefacts [Luo et al., 2021], leading to better performance on token to position de-correlation [Ke et al., 2021]. Haviv et al. [2022] even find that causal TLMs perform adequately even without an explicit APEs. However, a systematic study on relativity of positional encodings is still needed.

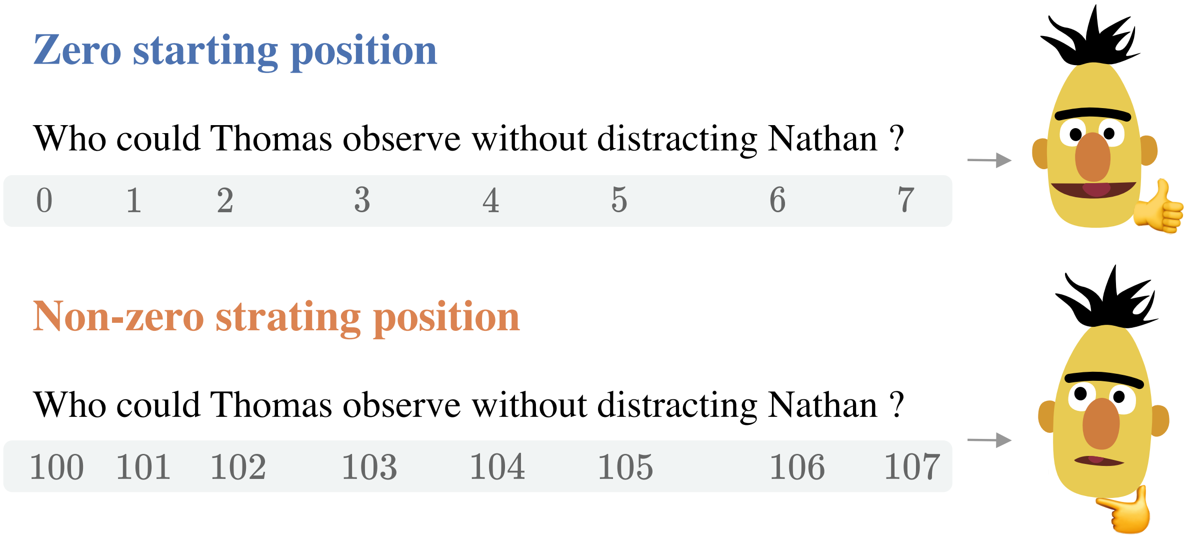

Zero starting position
Who could Thomas observe without distracting Nathan ?
0 1 2 3 4 5 6 7
Non-zero strating position
Who could Thomas observe without distracting Nathan ?
100 101 102 103 104 105 106 107

Figure 6.1 Transformer models with absolute positional embeddings have different representations for sentences starting from non-zero positions.

To better understand the relativity of absolute position embeddings, we first need to ascertain the robustness of relative position understanding for a given input. TLMs are typically trained in a batch containing multiple sentences, with a limited sequence window size, which is typically much larger than an average sentence. We hypothesize that a systematic model should encode the same sentence equally throughout this context window. However, evaluating the encoding of a sentence starting from any position in this window in isolation is hard, as the representation of the sentence would depend on the prior context Misra et al. [2020], Kassner and Schütze [2020].

In this chapter, we talk about our work Sinha et al. [2022], where we subject models from several different architectures and sizes to *phase shifting*. In this paradigm, the sentences exposed to the model are provided contiguous position identifiers starting from a non-zero position (Figure 6.1). Such inspection allows us to gauge the model's sentence encodings on different positions, emulating sub-window sentence representation, while factoring out the influence of prior context. We investigate several zero shot, few shot and full shot tasks by shifting the start positions of the sentences. We observe the following:

- TLMs display different sub-window sentence representation capabilities, resulting in decreased zero shot task performance and variability in sentence perplexities.
- Autoregressive models, including the recently published OPT Zhang et al. [2022b], show erratic zero and few-shot performance on sub-window representations, highlighting the brittleness of in-context learning evaluation.
- Masked Language Models (MLMs) encode sentences in non-standard positions better than their autoregressive counterparts.
- During fine-tuning models suffer drastically on cross phase-shifted evaluation, suggesting position specific overfitting.

We aim to raise awareness about issues with APEs, which are still widely used in pre-training large language models. Our results highlight the severity of position shortcuts taken by the model during pre-training and fine-tuning, and imply that TLMs may have vastly varying sub-window sentence representation capability than previously assumed.

6.1 Technical Background

Position encodings used by TLMs come in three broad categories: fixed sinusoidal embeddings as proposed by Vaswani et al. [2017], absolute or learned popularized by BERT Devlin et al. [2019b] family of masked language models, and relative positions [Shaw et al., 2018] used by T5 Raffel et al. [2020]. Fixed position embeddings Vaswani et al. [2017] consists of representing token positions with a sinusoidal function. BERT [Devlin et al., 2019b]-style family of masked language models propose using absolute position embeddings, which learn an unique vector assigned to each position. Subsequently, relative position embedding techniques have been proposed, which involve computing position embeddings on the fly based on a neighborhood window. Table 6.1 provides an overview of different positional encodings used in public models. Wang et al. [2021] presents a comprehensive overview of current encoding strategies.

Name	Release Year	Positional Encoding Type
BERT [Devlin et al., 2019a]	2019	Learned Absolute
RoBERTa [Liu et al., 2019a]	2019	Learned Absolute
GPT2 [Radford et al., 2019a]	2019	Learned Absolute
BART [Lewis et al., 2020a]	2020	Learned Absolute
LongFormer [Beltagy et al., 2020]	2020	Learned Absolute
T5 [Raffel et al., 2020]	2020	Relative Learned Bias
GPT3 [Brown et al., 2020]	2020	Learned Absolute
GPT-Neo [Black et al., 2021]	2021	Learned Absolute
Fairseq-Dense [Artetxe et al., 2021]	2021	Fixed Absolute
ShortFormer [Press et al., 2021]	2021	Fixed Absolute
GPT-J [Wang, 2021]	2021	Rotary
GPT-NeoX [Black et al., 2022]	2022	Rotary
OPT [Zhang et al., 2022b]	2022	Learned Absolute
PaLM [Chowdhery et al., 2022]	2022	Rotary

Table 6.1 Positional encoding of commonly used pretrained language models.

Despite being an older method, absolute positional embeddings (APEs) are reportedly better than its relative counterparts on several tasks [Ravishankar et al., 2021], and are still used by majority of the large pre-trained TLMs, including the recently released OPT Zhang et al. [2022b]. APEs compute token representation after adding the input token to the position embedding for the corresponding position: $x_i = \theta_W[w_i] + \theta_P[i]$, where, $\theta_W \in \mathbf{R}^{|V| \times d}$ is the token vocabulary of size $|V|$, embedding dimension d, and the absolute position embedding matrix $\theta_P \in \mathbf{R}^{|T| \times d}$, where T is the maximum context window size of the model. Now, a sentence $S = [w_1, w_2 ... w_n]$ containing n tokens, is mapped during inference to positions 1,2, ... n contiguously for all models.

TLM offer various sizes of *context window*, which is the maximum sequence length in tokens it can train and infer on. Since this context window is usually larger than the average sentence length, multiple sentences can be packed together to “fill" the context window during pre-training. This allows TLMs to learn that sentences can start from various positions in their context window. If models trained with APEs do encode relativity of position, then the sentence representations should be roughly equal throughout the context window, regardless of their starting position.

To understand the relativity of APEs, we examine the model performance under *phase shift* conditions. Phase shift[1] involves right-shifting the absolute positions of all tokens in the sentence by an equal distance k, such that the tokens are now mapped to new positions $1 + k, 2 + k, ..., n + k$, or $x_i = \theta_W[w_i] + \theta_P[i + k]$. As such, phase shifting changes only the absolute position, but preserves the relative distances between tokens in the a sentence. Theoretically, we can shift the positions within the context window as long as $k + n \leq T$.

[1] More related to our work, Kiyono et al. [2021] train a Transformer model from scratch using shifted positional embeddings for machine translation, and observe improved performance in extrapolation and intrapolation setup.

6.2 Evaluated Models

We used 11 publicly available pretrained language models in this work, ranging across different architecture families: Encoder, Sequence-to-Sequence, and Auto regressive models. All of them use absolute positional embeddings (APE) that is learned during pretraining. In §6.2.1, we follow the standard practice for in-context learning evaluation [Brown et al., 2020, Black et al., 2022, Gao et al., 2021] and use autoregressive models. In our initial experiments, we found GPT2 to have a similar behaviour to OPT models, and since the OPT models are available in a wider range of sizes, we primarily focus on them for these experiments. In fine-tuning (§6.4.3) and acceptability (§6.4.1) experiments, we assess all model families. However, because of the computational costs associated with these experiments, we opt for model variants with < 1B parameters. The details of all models can be found in Table 6.2. We use HuggingFace [Wolf et al., 2020b] model hub to load, fine-tune train, and run infererence for all models.

Model	Type	Pretraining Objective	Context Size	First Position	# Layers	Hidden Size	# Params
RoBERTa family [Liu et al., 2019a]							
RoBERTa$_{\text{BASE}}$	encoder-only	Masked Language Modeling	514	2	12	768	123M
RoBERTa$_{\text{LARGE}}$	encoder-only	Masked Language Modeling	514	2	24	1024	325M
BART family [Lewis et al., 2020a]							
BART$_{\text{BASE}}$	encoder-decoder	Masked Language Modeling	1024	2	6	768	140M
BART$_{\text{LARGE}}$	encoder-decoder	Masked Language Modeling	1024	2	12	1024	400M
GPT2 family [Radford et al., 2019a]							
GPT2	decoder-only	Next Token Prediction	1024	0	12	768	125M
GPT2$_{\text{MEDIUM}}$	decoder-only	Next Token Prediction	1024	0	24	1024	345M
OPT family [Zhang et al., 2022b]							
OPT$_{\text{125M}}$	decoder-only	Next Token Prediction	2048	2	12	768	125M
OPT$_{\text{350M}}$	decoder-only	Next Token Prediction	2048	2	24	1024	350M
OPT$_{\text{2.7M}}$	decoder-only	Next Token Prediction	2048	2	32	2560	2.7B
OPT$_{\text{13B}}$	decoder-only	Next Token Prediction	2048	2	40	5120	13B
OPT$_{\text{30B}}$	decoder-only	Next Token Prediction	2048	2	48	7168	30B

Table 6.2 Details of the models we used in this paper.

6.2.1 Prompting

For evaluating zero-shot inference and in-context learning, we make use of EleutherAI Language Model Evaluation Harness [Gao et al., 2021], an open-source library that is used for evaluating autoregressive pretrained language models [Black et al., 2022]. In the zero-shot setting, each example is converted to a prompt using task-specific templates. Then, the prompt is fed to the language model to elicit the answer. Similarly, in the few-shot setup, a prompt is created from the concatenation of few dataset examples base on the same template and are prepended as a context to validation instances. In our experiments, we use default templates provided by the EleutherAI Language Model Evaluation Harness. The task performance is computed over the validation set of due to the lack of public test sets, except for ARC, where we evaluate the models on the test set. We set the number of few-shots examples to be five and randomly sample them from the training set of each dataset. We report the few-shot results averaged over five random seeds. Note that feeding inputs to the models still follows the same protocol introduced in §6.4.1.

6.3 Evaluated Datasets

Dataset	# Train	# Test/Validation
BliMP	-	67000
COPA	400	100
PIQA	16113	1838
WinoGrande	40398	1267
ARC (Easy)	2251	2376
MRPC	3668	408
RTE	2490	277
CoLA	8551	1043

Table 6.3 Dataset statistics we used in this work.

We use BLiMP [Warstadt et al., 2020b] for the grammatical acceptability experiments in §6.4.1 as it is typically employed in a inference-only setting and does not require additional training. For §6.4.3, we take three tasks from the standard language understanding benchmark GLUE [Wang et al., 2018] which is often used for finetuning language models: MRPC, RTE, and COLA. In addition to these three tasks, we use four other datasets, COPA, PIQA, WinoGrande, and ARC, on which the OPT family have previously demonstrated good performance Zhang et al. [2022b]. Table 6.3 shows the statistics of all datasets, and the following provides a brief description of them:

- **BLiMP** [Warstadt et al., 2020b] is a challenge set designed to measures the model's ability to distinguish between acceptable and unacceptable English sentences. This benchmark consists of synthetic examples created based on expert-crafted grammars, where each instance comes with two versions: one acceptable and one unacceptable.

- **COPA** [Gordon et al., 2012] is an open-domain commonsense causal reasoning task, where the model is given a premise and must correctly identify its cause or effect. COPA consists of short hand-crafted sentences and is provided as a multi-choice task.

- **PIQA** [Bisk et al., 2020] is a physical commonsense benchmark dataset, challenging language models' idea of the physical world. Given a physical goal, a model must choose the most plausible solution between two choices. This benchmark is used in the multi-choice format.

- **WinoGrande** [Sakaguchi et al., 2020] is a commonsense reasoning benchmark based on the Winograd Schema Challenge (WSC) [Levesque et al., 2011] with increased hardness and scale. The dataset is provided as a pronoun resolution problem, where the model must recover an ambiguous pronoun in a given context.

- **ARC** [Clark et al., 2018] is collected from grade-school-level science questions commonly asked in exams. This question-answering dataset is provided in a multi-choice QA format suitable for evaluating pretrained language models. We use the "`easy`" subset of this benchmark.

- **MRPC** [Dolan and Brockett, 2005] is a paraphrase identification dataset collected from online news websites and has become a standard benchmark in the NLP community. We follow the previous works and treat the data as a text classification task.

- **RTE** [Giampiccolo et al., 2007a] is one of original subtasks in the GLUE benchmark and comprises textual entailment challenges. We follow the standard format and use Natural Language Inference (NLI) protocol for this dataset.

- **CoLA** [Warstadt et al., 2019d] is a linguistic acceptability dataset, where each example is an English sentence annotated with a binary label showing whether it is a grammatical sentence. This is a text classification dataset and we follow the standard protocol and report Matthews correlation coefficient [Matthews, 1975].

6.3.1 Grammatical acceptability

We use all 67 subsets (a total of 67K data instances) of BLiMP Warstadt et al. [2020b]. A model achieves a score of 1 if it successfully assigns a lower perplexity to the grammatical version of each example. We report the average score across the entire dataset for starting positions that are shifted in the intervals of 10. The inputs are fed to the models in the format explained in §6.1. Recall that perplexities are ill-defined in case of Masked Language Models. Thus, we follow the formulation of Salazar et al. [2020] to compute a pseudo-perplexity for RoBERTa and BART. We adopt the Minicons Misra [2022] library to compute the perplexities, which provides a unified interface for models hosted in HuggingFace [Wolf et al., 2020b].

6.4 Results

6.4.1 Impact of phase shifts on grammatical acceptability

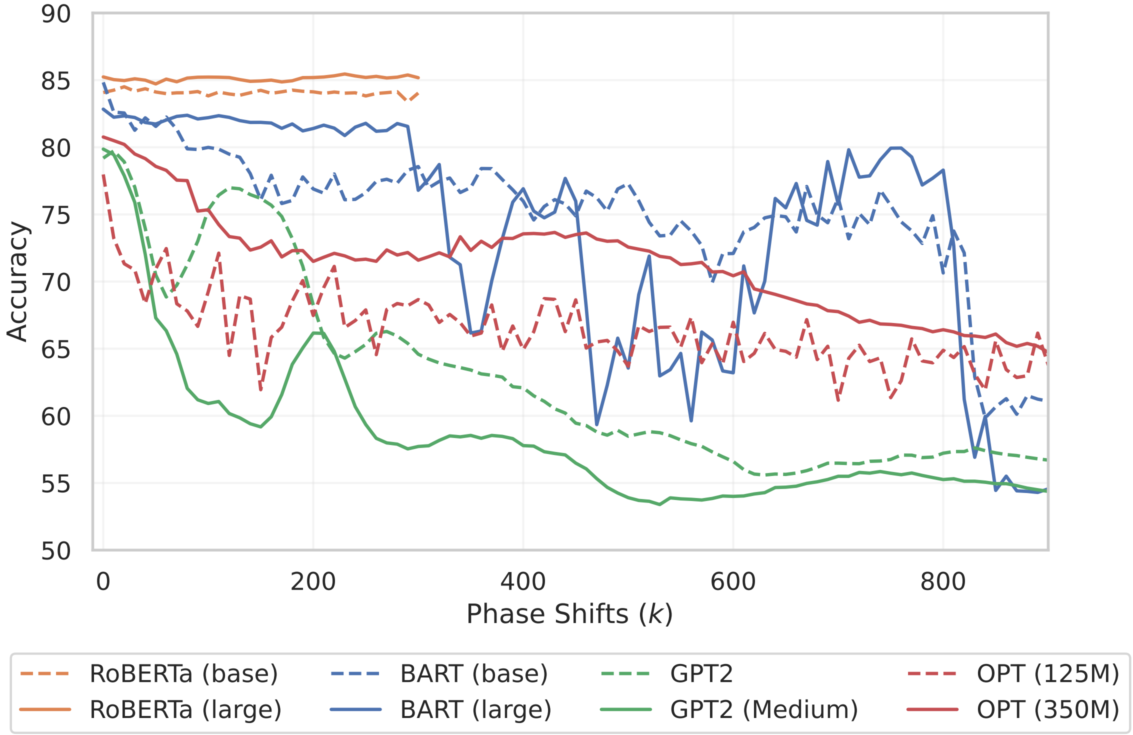


Figure 6.2 Acceptability Scores in BLiMP Warstadt et al. [2020a] dataset across different phase shifts. RoBERTa only supports context window of size $T = 512$, so we capped the scores to phase shift $k = 300$ to allow for sentences of maximum length in BLiMP to be evaluated.

First, we investigate the impact of phase shifting on the model performance. We compute the perplexities of several publicly available models—RoBERTa [Liu et al., 2019b], BART [Lewis et al., 2020b], GPT2 [Radford et al., 2019a] and OPT [Zhang et al., 2022b]—to evaluate the grammatical acceptability capabilities of the model, using the BLiMP Warstadt et al. [2020b] benchmark.[2] We compute the task score by comparing

[2] We adopt the perplexity computation strategy for RoBERTa and BART from Salazar et al. [2020]

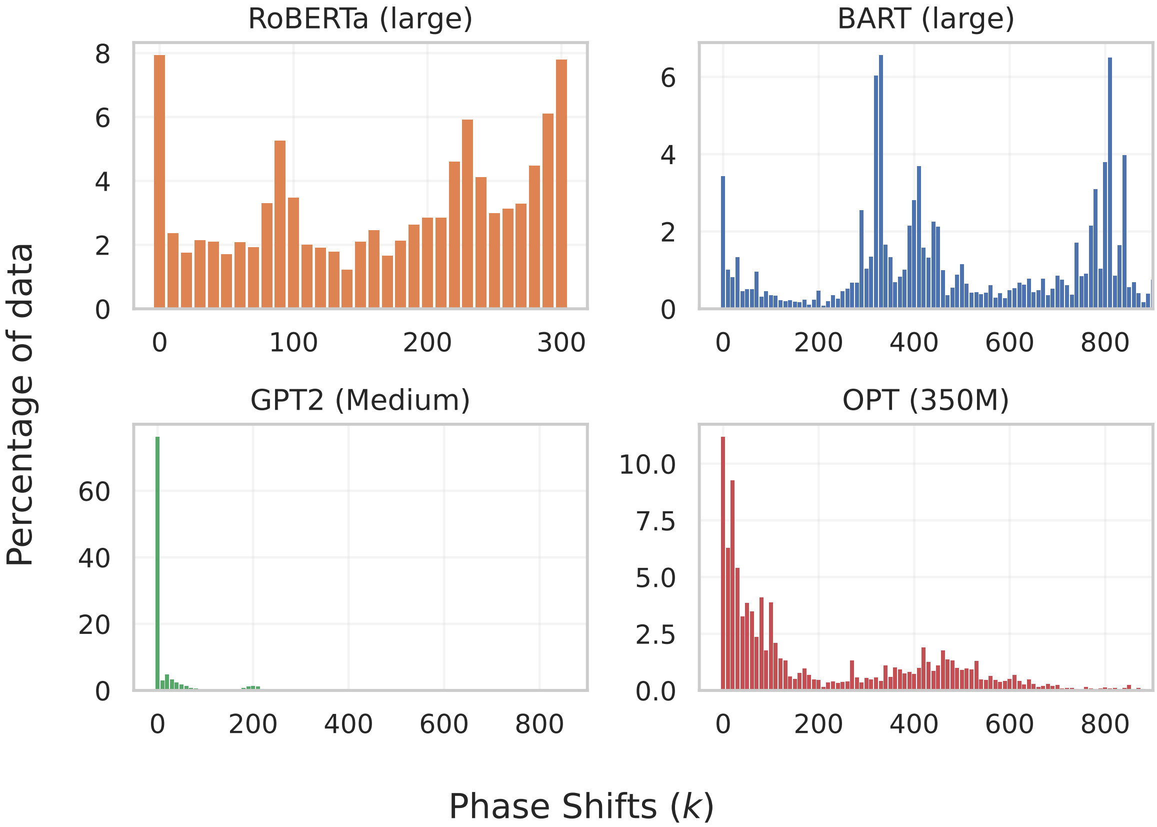


Figure 6.3 Distribution of sentences in BLiMP Warstadt et al. [2020a] having the lowest perplexities (i.e., are deemed most acceptable) for each phase shift.

grammatical and ungrammatical sentence perplexities, and applying the phase shift in increasing values of k to the sentences and models (Figure 6.2).

While computing the task scores and perplexities of the models, we observed that all of the models exhibit poor task performance on phase shifts. Due to the non-shiftable nature of the `[CLS]` token in masked language models (MLMs), we fixed the position of `[CLS]` token to start position during phase shifting. However, we observed a marked improvement in task performance when we use trigger tokens in the beginning of the sentence, typically the end-of-sentence (`[EOS]`) token in case of MLM models (RoBERTa, BART). An explanation for this ambiguity in results is that typically when models are pre-trained, multiple sentences are packed together in the context window by delimiting the start of each sentence with an `[EOS]` token. While this is not the case for GPT2, we also observed improved performance in some cases when we add a beginning of sentence (`[BOS]`) token to the sentence and add a special `[EOS]` token to delimit the start of a sentence. Thus, in all of our results, we opt with this configuration (adding an `[EOS]` token before the sentence) to ensure fairer evaluation for all model families. Concretely, the input to a model uses the following template:

```
[CLS][EOS]<sentence>
```

In cases where a model does not have the `[CLS]` token, we instead use `[BOS]`. If none of those are available, we replace it with `[EOS]` (so a total of two `[EOS]`'s will be prepended). For phase shift k, we fix the position of `[CLS]` token to be the first available position in model's APE (refer to Table 6.2) and we shift every position id by k. For example, given phase shift $k = 100$, and first position id being 1, and sentence length of n, we have the following vector of position ids:

$$\vec{p} = [1, 100, 101, \ldots, n + 100]$$

We observe that the task performance of all models, except for RoBERTa, drastically suffers from phase shifting. Autoregressive models in particular display worse results. This is likely due to a mismatch of position information learned due to the causal language modelling objective vs the position information provided to the model during phase shift [Haviv et al., 2022]. We also compare the perplexities of each sentence across different phase shifts and plot the frequency of sentences having the lowest perplexity in each k (Figure 6.3). We observe in GPT2 that more that 70% of the sentences have their best perplexity in $k = 0$, highlighting a severe zero-position bias. OPT_{350M} has better sub-window sentence representation capacity than similarly sized GPT2, which is also evident from the acceptability results in Figure 6.2.

6.4.2 Impact of phase shifts on in-context learning

More recently, zero-shot and few-shot inference, commonly referred to as in-context learning, have become a de facto standard in evaluating pretrained language models [Brown et al., 2020]. In this approach, the model's predictions are produced by conditioning it on certain prompts, such as instructions (zero-shot setting) or a few examples of input-output pairs (few-shot setup). In both cases, the model faces an extended input text, and we suspect it will be affected by deficiencies of APE. To evaluate this hypothesis, we employ an experimental setup similar to §6.4.1. Under zero-shot and five-shot inference regimes, we assess the model performance on standard NLP tasks when it is fed with inputs in increasing values of phase shifts. We choose OPT model family, because it is available in a wide range of sizes (125M to 30B parameters), allowing allows us to examine the behavior of APE at different scales. Moreover, our evaluations take into account four tasks reported in the original paper: Winogrande [Sakaguchi et al., 2020], COPA [Gordon et al., 2012], PIQA [Bisk et al., 2020], and ARC [Clark et al., 2018] as well as two classification datasets from GLUE benchmark [Wang et al., 2018]: MRPC and RTE. We provide an aggregated view of the models' perfor-

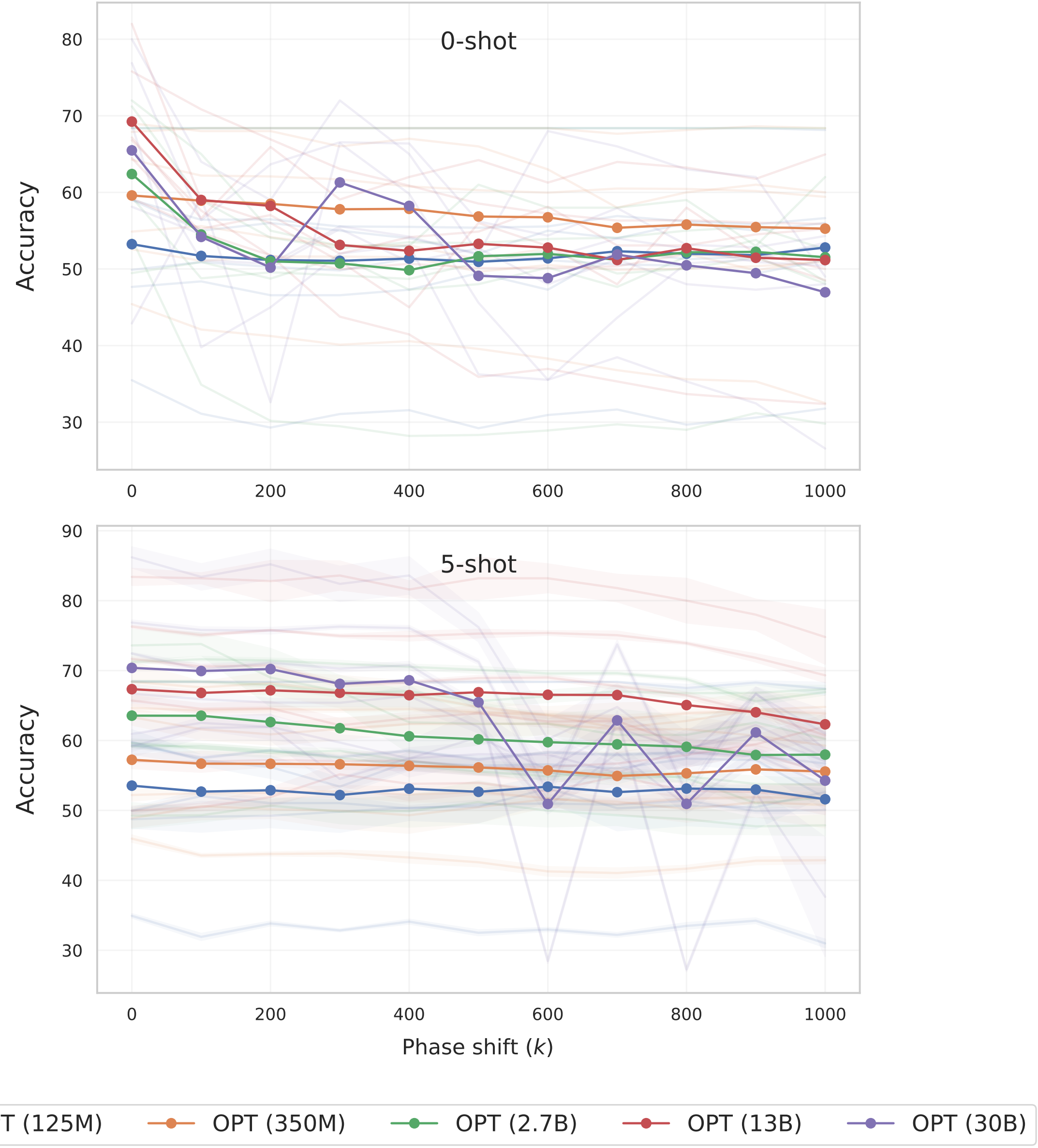


Figure 6.4 Aggregate performance of OPT family on six NLP tasks when various phase shifts are applied.

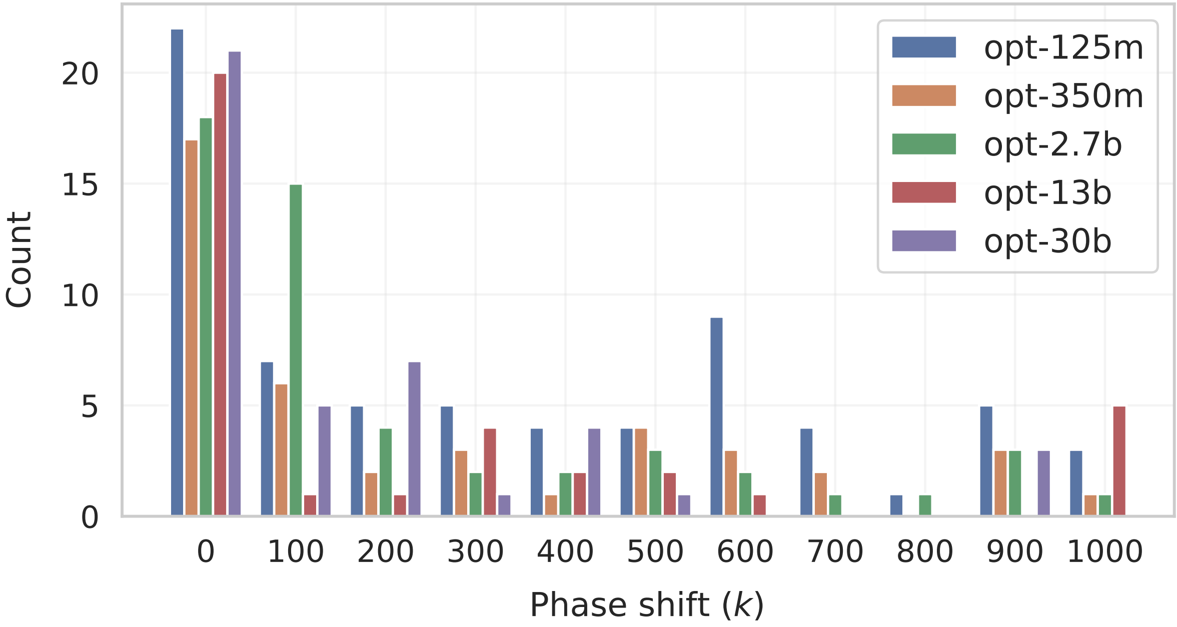


Figure 6.5 Distribution of prompts with best accuracy across all six tasks.

mance on all six accuracy-dominated benchmarks in Figure 6.4. The detailed plots for each task are in §6.5.1.

In most tasks, the performance deteriorates when the model process inputs in any other phase shift than zero, especially in zero-shot inference. More importantly, the model's performance is not always adversely affected by phase shifts. In fact, Figure 6.5 shows that non-zero starting positions result in the best accuracy for many prompts. This erratic performance is present in all model sizes, and scaling the number of parameters does not help. Furthermore, one can see larger models are more affected by shifted starting position, which suggests that absolute positional embedding might need more data or training as the number of parameters increases.

6.4.3 Impact of phase-shifts on fine-tuning

Finally, we investigate the effect of phase shift in fine-tuning. We ask whether the models can generalize to out-of-phase sentences for a given task. We train RoBERTa, BART, GPT2 and OPT models on CoLA, RTE and MRPC tasks from the GLUE benchmark [Wang et al., 2018] and evaluate them on phase-shifts. We choose these three relatively small tasks in order to decrease the number of gradient updates to position embeddings during fine-tuning. We perform a cross-phase analysis by training and evaluating across different phase shifts ($k = 0, 100, 200, 300$) for all models on the same set of datasets, and show the averaged performance. We observe for all models, the task performance drops during out-of-phase evaluation (non-diagonals in Figure 6.6).

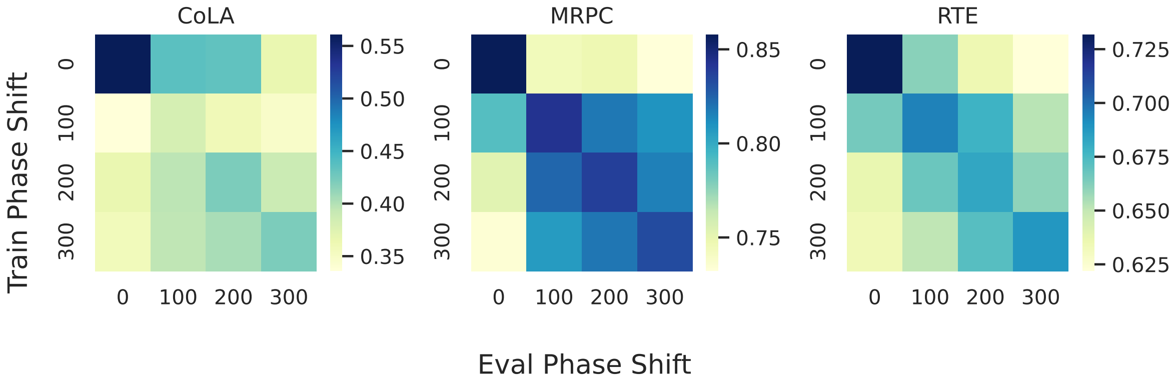


Figure 6.6 GLUE task heatmap with varying fine-tuning train and test phase shifts, averaged across all models. Darker colors represent better task performance.

The drop in performance of evaluating out-of-phase sentences might just be simply attributed to overfitting on position information during fine-tuning. However, we observe that for all tasks, training and evaluating on the same phase-shift is worse when $k \neq 0$ (diagonals in Figure 6.6). Out-of-phase training appears to be worst for CoLA, which suffers drastically when fine-tuning on different phase shifts. These results highlight a potential task data bias with respect to different positions.

6.5 Analysis

6.5.1 Further evaluation on Phase shifting with prompts

We displayed a holistic view of zero-shot and five-shot experiments in Figure 6.4, covering the accuracies averaged over all six datasets. In this section, we now report and analyze the result of each dataset individually. Figure 6.7 and Figure 6.8 showcase models' performance in zero-shot and five-shot configurations. The same pattern can be seen across all model sizes in COPA, WinoGrande, PIQA, ARC (Easy), and RTE. Concretely, the zero-shot abilities of the models sharply decrease as we increase the starting position. Moreover, five-shot inference, typically referred to as in-context learning, is also subject to decreased performance, ranging from -2% to -40%. However, the degradation is not as severe as with zero-shot setting. Only MRPC exhibits stable phase shift performance, but even in this case, larger models are still adversely affected. Due to the exceptionally poor performance of OPT family on CoLA, we exclude these results from our analysis (Figure 6.8).

The erratic behaviour observed in majority of evaluated datasets makes it evident that models struggle to encode the relative distances of words as their understanding of inputs heavily change with various phase shifts. It is important to note that our findings demonstrate models' unstable functioning as opposed to solely highlighting their failure. Indeed, Figure 6.5 shows that one can extract better and improved accuracies with non-zero starting positions. Namely, OPT_{30B} has the best zero-shot performance on phase shift $k = 300$ in the case of MRPC; the same pattern can also be observed in RTE five-shot for OPT_{13B} on phase shift $k = 300$. Another noteworthy observation is that the performance drop is often a *non-monotonic* function of phase shifts. i.e., for some prompts, the model might be more accurate for $k = 1000$ than for $k = 0$. This observation suggests that some positional biases might be learned during pre-training and are well-captured by APE. So, increasing values of k in some occasions lands the

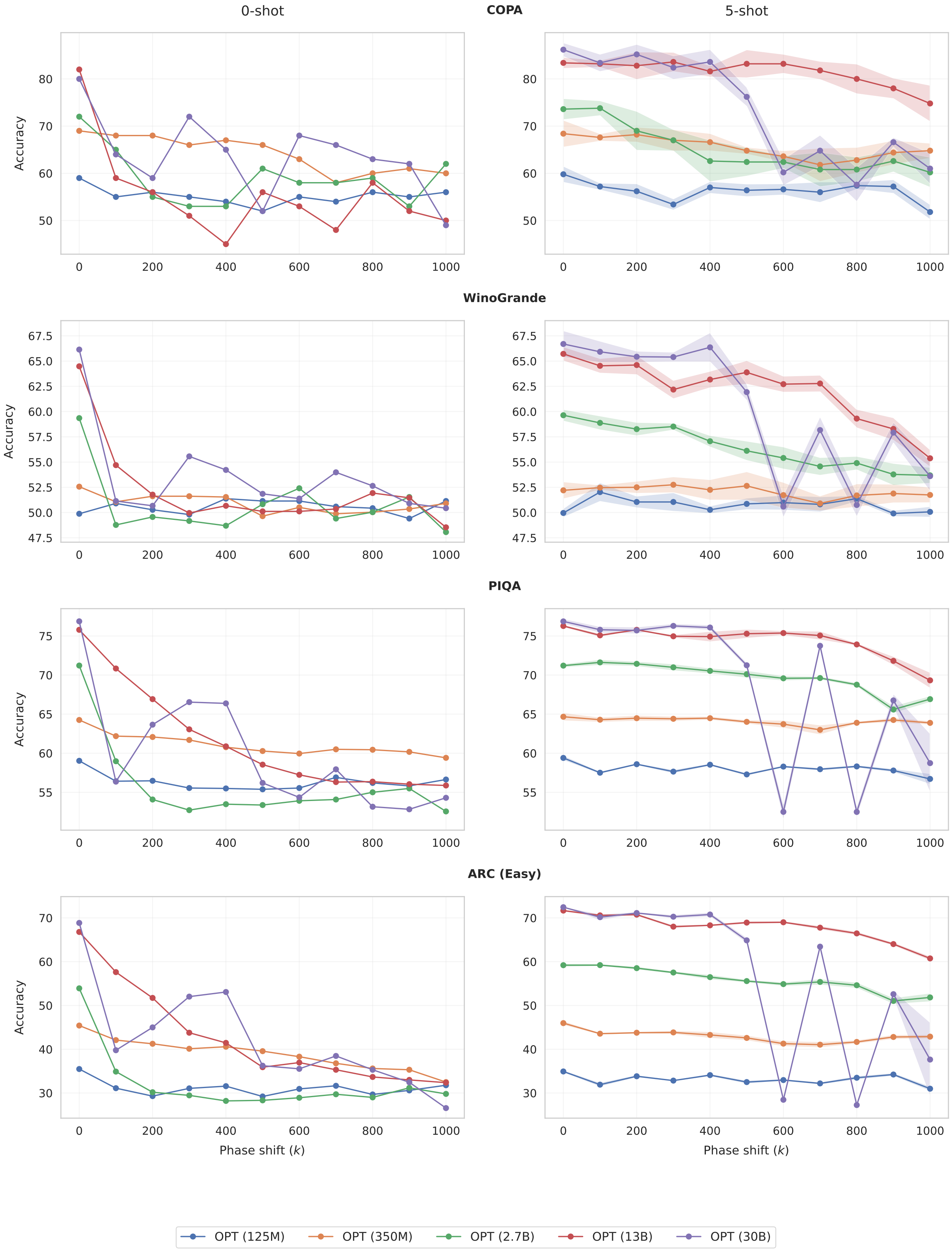


Figure 6.7 Zero-shot and Few-shot performance of OPT family with various phase shifts for each individual dataset (Part 1)

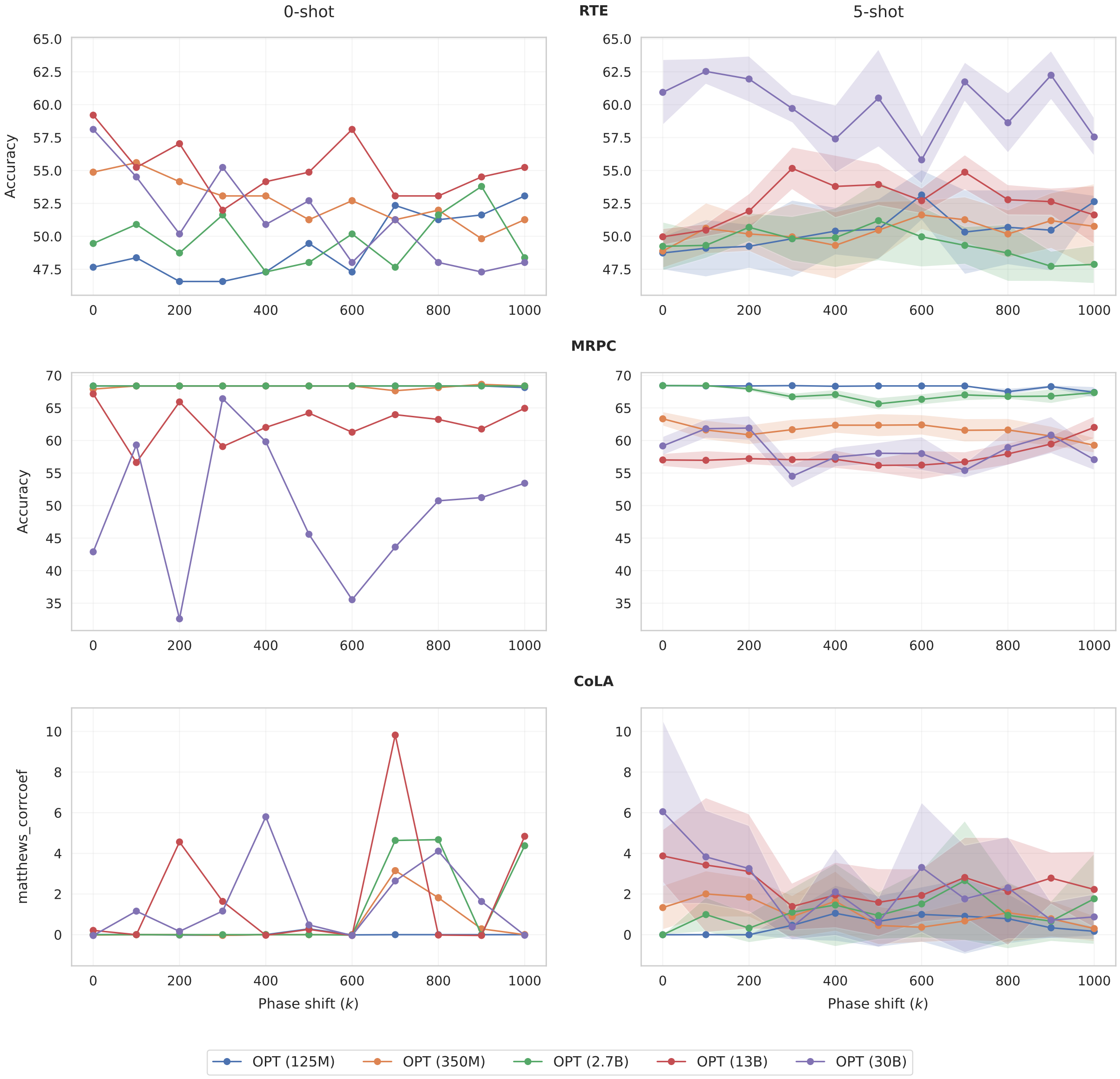


Figure 6.8 Zero-shot and Few-shot performance of OPT family with various phase shifts for each individual dataset (Part 2)

model attentions in a "sweet spot" in the processing window, such that the model benefits from some positional biases learned during pre-training.

We observe the presence of erratic behavior across a fairly wide range of model sizes in the OPT family. Additionally, it can be seen that larger models are more prone to fail at encoding relative positions than their smaller counterparts. One possible explanation for this is that in order for the models to encode relative positional information, they need to view all combinations of words and sentences in every position. This coverage rarely occurs in natural data, resulting in data sparsity issues. Hence, models with a large number of parameters may require more data/training to learn the relative ordering of words.

6.5.2 Variation of best perplexity across phase shifts

In this section, we investigate the perplexity of individual sentences from the BLiMP dataset across each phase shift for each model. We plot the distribution of sentences achieving lowest perplexity in each phase shift for the range of models in Figure 6.9. We observe several modes of phase shift for RoBERTa and BART models where they have the least perplexity on phase shifts other than the standard (zero position). In the case of GPT2 and OPT, the distribution is more skewed towards zero, indicating they almost always achieve the lowest perplexity in the zero position, i.e. when there is no phase shift.

6.5.3 Variation in attention patterns with phase shift

We further perform attention analysis on GPT2, RoBERTa and BART to visualize whether the model's attention pattern changes with phase shifts. Following the experimental protocol of Raghu et al. [2021], we first collect a summary of attention weights computed with token distances for each token-pair in a sentence. This summary metric is then further normalized for sentence length. The values of this metric show whether the attention is local (low values)—focused on small token distances—or global (high values)—i.e. focused on the whole sentence.

We compute this attention summary metric on a sample of 5000 sentences drawn from the BLiMP dataset Warstadt et al. [2020b]. We then plot the summary values per layer and sort according to the values for each attention head, as per Raghu et al. [2021]. The idea is to discover whether this attention summary metric is drastically different under different phase shift conditions.

We do observe drastic differences in attention patterns in all layers for GPT2 (Figure 6.10) and GPT2-Medium (Figure 6.11). Comparing this with of RoBERTa (base) (Figure 6.12) and RoBERTa (large) (Figure 6.13), we can corroborate our findings from §6.4.1—RoBERTa is much more robust to phase shifts. Consequently, BART (Figure 6.14 and Figure 6.15) also displays differences in attention patterns, but they are not as drastic as GPT2.

6.6 Related Work

Positional encoding has been always an important part of the Transformer architecture, and since it original introduction different variants of it have been deployed by pretrained models (see Table 6.1 for a summary of positional encoding used by some of popular state-of-the-art models.)

Positional encodings have garnered a niche community over the past several years. Wang and Chen [2020] investigate whether position embeddings learn the meaning of positions and how do they affect the learnability for different downstream tasks. Wang et al. [2021] explore different positional encodings and establish monotonicity, translation and symmetry properties of different methods, including APEs. They also report that learned APE's demonstrate superior performance for text classification, further adding to the evidence APE's enable exploitation of positional biases. Luo et al. [2021]

report that masked language model embeddings consists of positional artefacts which bias the model output. More related to our work, Kiyono et al. [2021] train a Transformer model from scratch using shifted positional embeddings for machine translation, and observe improved performance in extrapolation and intrapolation setup. Haviv et al. [2022] reports a surprising finding that autoregressive Transformer models trained without explicit positional information still perform on-par with their counterparts having access to positional information. This result is attributed to the causal attention structure induced by the autoregressive training only, as this effect is not observed with masked language models, as highlighted by both Haviv et al. [2022] and Sinha et al. [2021a]. Ke et al. [2021] proposes a novel technique to de-correlate the position encodings and token embeddings, and achieve better downstream performance than baselines. Ravishankar et al. [2021] find relative positional encoding does not improve over APE in multi-lingual setting.

On the other hand, multiple works have shown the advantage of explicit relative positional encoding for length extrapolation. Csordás et al. [2021] show Transformers equipped with variants of relative positional encoding [Dai et al., 2019, Shaw et al., 2018] significantly outperform their absolute counterparts when it comes to length generalization. In the same line of work, Ontanon et al. [2022] also find that for numerous synthetic benchmarks, the best extrapolation performance can only be obtained by relative positional encoding. Press et al. [2022] take the experiments beyond synthetic datasets and show that APE's struggle in generalization to longer sequence of natural language. All of these amount to the evidence that points to APE's as one of the potential reasons Transformers are known to fail in length generalization and productivity [Hupkes et al., 2020, Lake and Baroni, 2018]. Although the benefits of using explicit relative positional bias is mentioned in various works, they typically come at the cost of slowing the training down: [Press et al., 2022] report that training T5 (which uses a relative variant of positional encoding) is almost twice as slow as training a model with sinusoidal absolute embedding. Thus, the gained runtime efficiency allows longer training of the APE model, which in turn enables the further extrapolation capabilities. These works suggest that we have a lot left to explore about positional encoding and highlight the fact that the consequences of particular choices is still an open field of ongoing research.

6.7 Discussion

In this chapter, we investigate the abilities of APEs in encoding the relative positions of the tokens in an input. We observe that TLMs using APEs encode sentences differently based on the starting position of the sentence in the context window. To summarize our findings:

- **Reduced sub-context window sentence processing capability of TLMs.** Majority of TLM's show worse perplexity scores when the position information is shifted, emulating a different start of sentence (§6.4.1).

- **MLMs offer better sub-context sentence representations.** Masked Language Models have much better ability to reconcile sentence representations from within a context window, compared to their autoregressive counterparts (Figure 6.2).

- **MLM models have lower surprisal scores in sub-context positions.** Even so, Masked LM's also display large variations in perplexity, leading to wide fluctuations in the best starting position for a given sentence. This highlights different sub-window representation capability of MLMs, which can be leveraged further to develop more generalizable models.

- **Systematicity issues for Autoregressive models in Prompting.** Autoregressive models remains highly susceptible to the shift in starting positions, possibly due to mismatch in their own, implicit position representation vs the provided one

[Haviv et al., 2022]. This is evident with the wildly fluctuating and significantly worse results on prompting (§6.5.1).

- **Poor out-of-context generalization in full fine-tuning.** Full finetuning with different class of models also highlights the over-dependence of models (both MLM and Autoregressive) on the starting position of a given sentence. Notably, models display poor *out-of-phase* generalization, and present the best results when trained without any phase shift, highlighting a potential position-to-data bias (§6.4.3).
- **Different sentence processing behavior in sub-context positions.** When provided with different starting positions, the models attention behaviors also drastically changes (§6.5.3), exhibiting systematicity issues in in-context sentence representation capability of large language models.

These results has major implications in the way we perceive the sentence processing capabilities of TLMs. Specifically, we observe that the representation of the same sentence varies depending on where it is in the context window, such that it impacts zero shot, few shot and full shot task performance of sub-window sentences. The results and analysis in this chapter also explains the erratic behavior of models towards different word order as observed in §4 and §5, in that APE's do not contain the necessary inductive bias to represent the relative position information of words in a sentence. Future work could leverage the start position in building robust and position-generalizable models. We hope our work can inform the community on the pitfalls of using APEs, and inspire development and adoption of alternative relative position embedding based approaches.

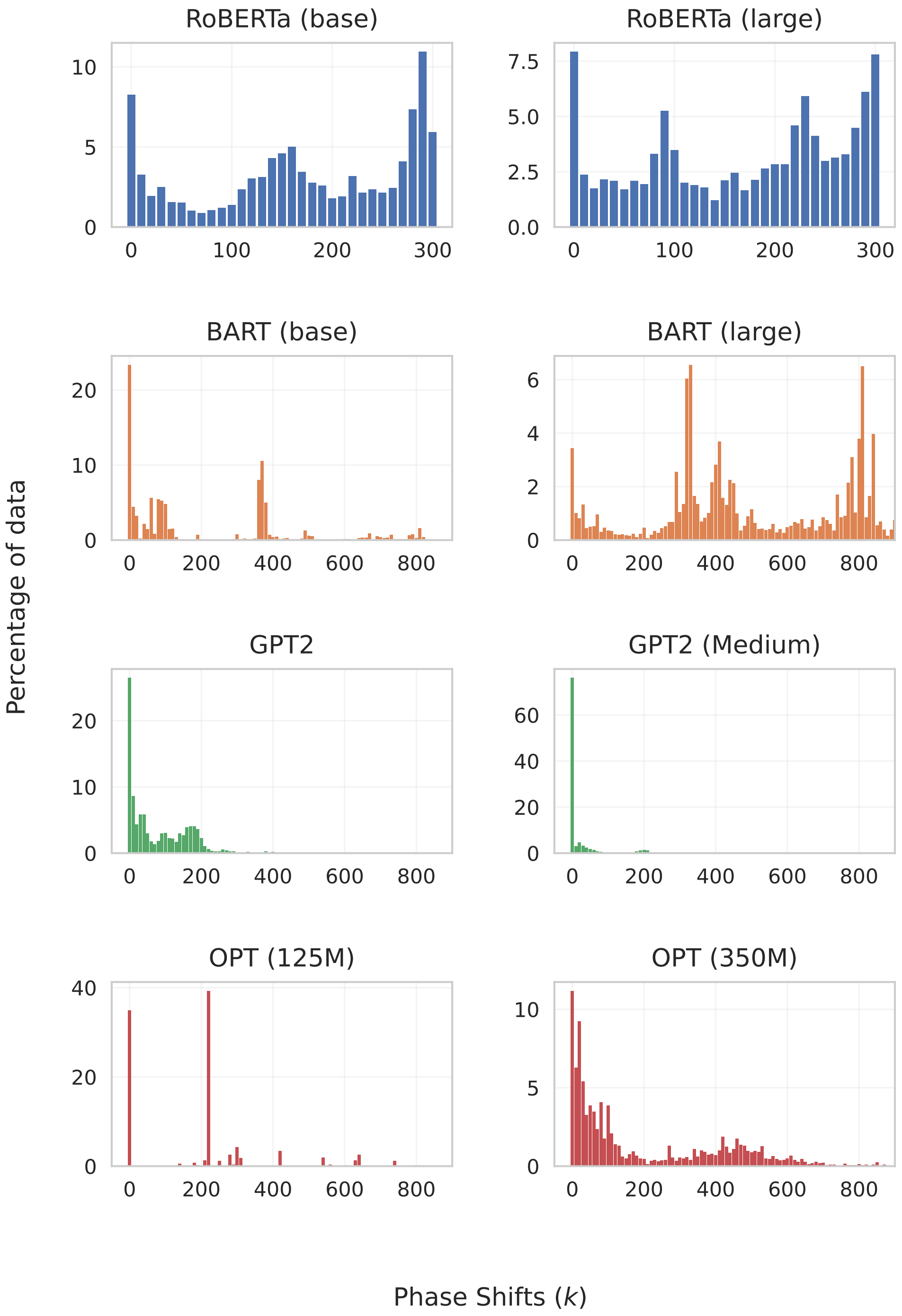


Figure 6.9 Distribution of sentences having the lowest perplexities for each phase shift

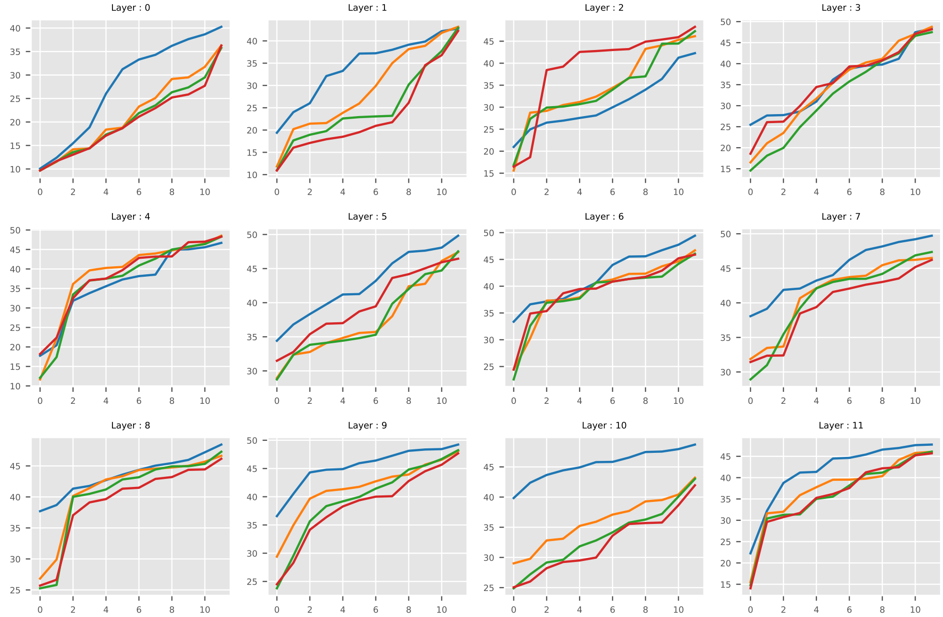


Figure 6.10 Attention globality distributions of GPT2 across different heads (sorted according to value) and averaged over all layers and 5000 data points. Blue curve stands for the no phase shift condition, and orange, green and red curves represent $k = 100, 200$ and 300 respectively.

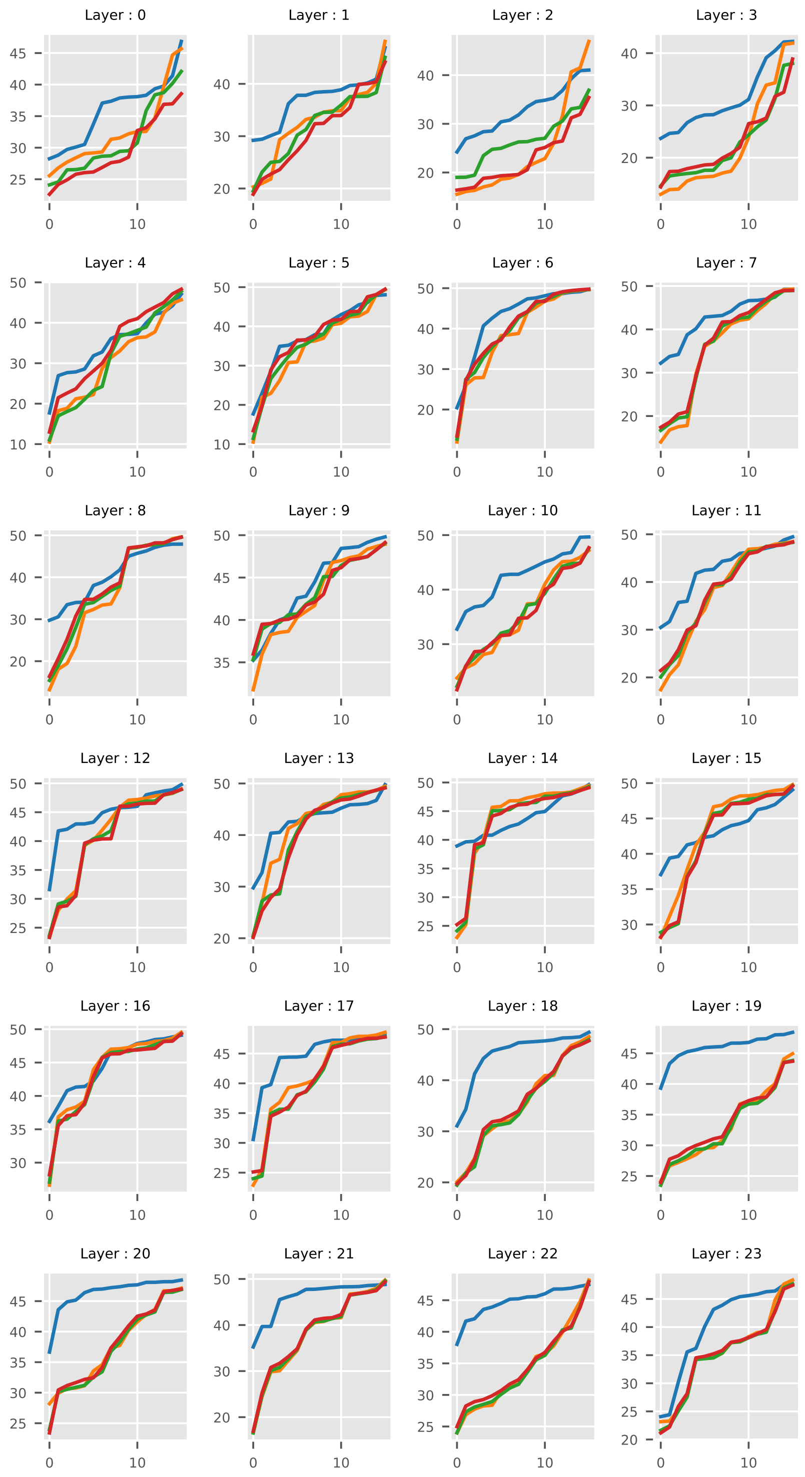


Figure 6.11 Attention globality distributions of GPT2-Medium across different heads (sorted according to value) and averaged over all layers and 5000 data points. Blue curve stands for the no phase shift condition, and orange, green and red curves represent $k = 100, 200$ and 300 respectively.

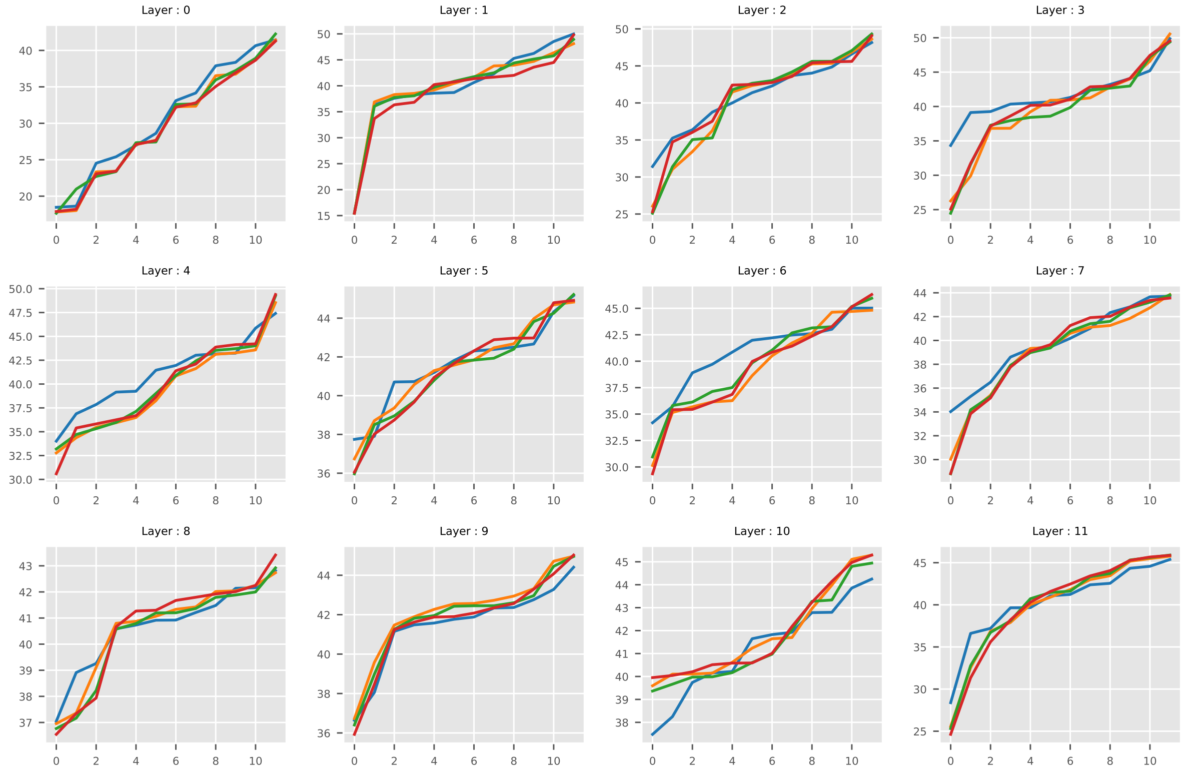


Figure 6.12 Attention globality distributions of RoBERTa (base) across different heads (sorted according to value) and averaged over all layers and 5000 data points. Blue curve stands for the no phase shift condition, and orange, green and red curves represent $k = 100, 200$ and 300 respectively.

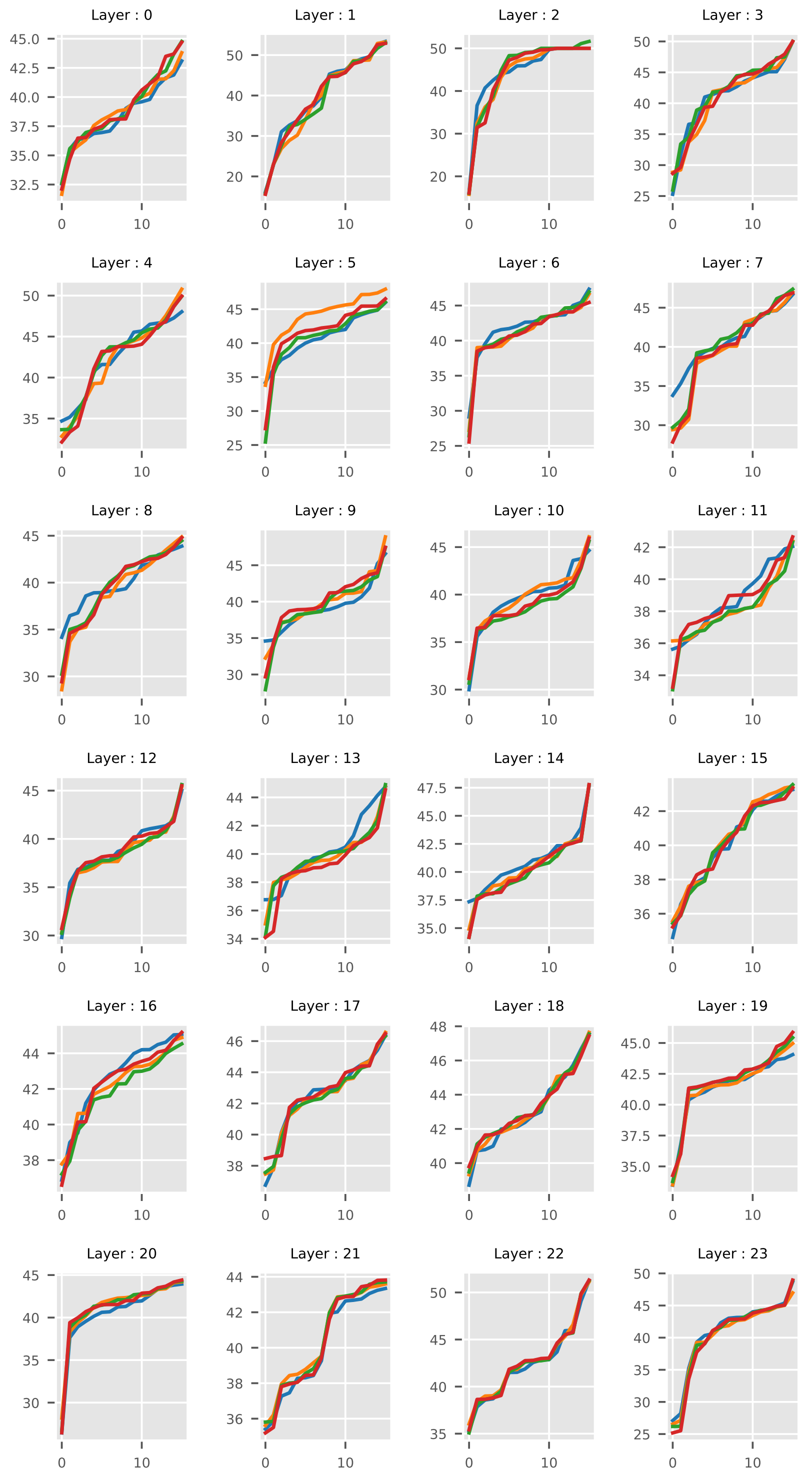


Figure 6.13 Attention globality distributions of RoBERTa (large) across different heads (sorted according to value) and averaged over all layers and 5000 data points. Blue curve stands for the no phase shift condition, and orange, green and red curves represent $k = 100, 200$ and 300 respectively.

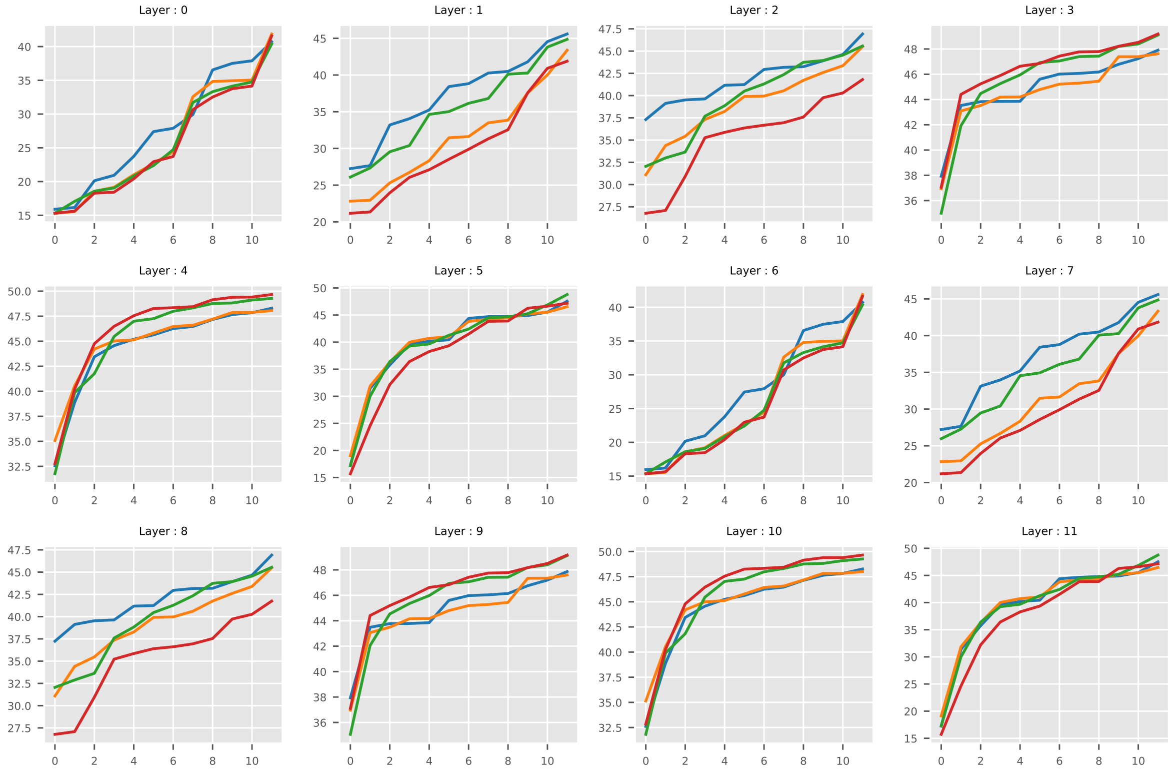


Figure 6.14 Attention globality distributions of BART (base) across different heads (sorted according to value) and averaged over all layers and 5000 data points. Blue curve stands for the no phase shift condition, and orange, green and red curves represent $k = 100, 200$ and 300 respectively.

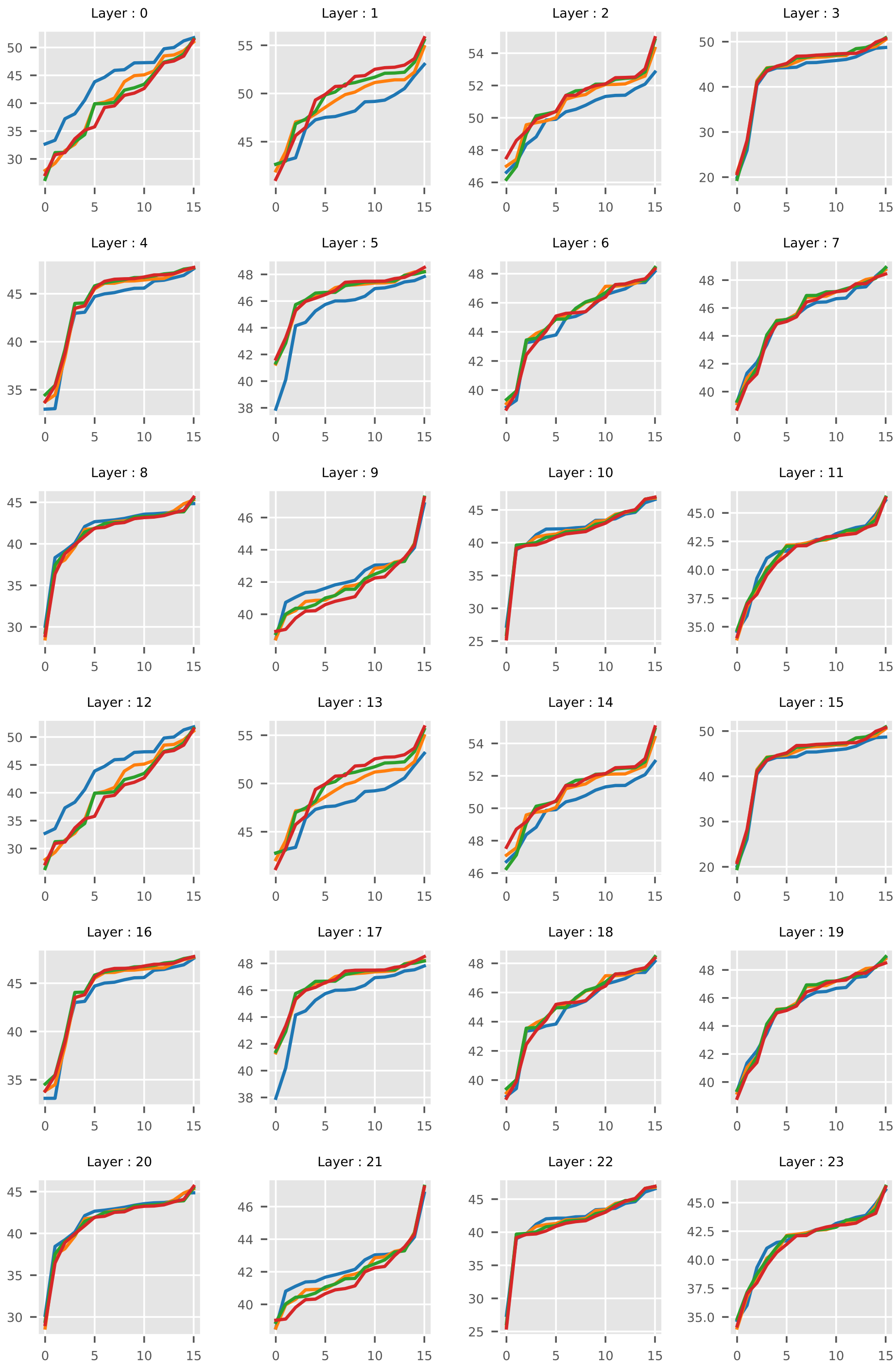


Figure 6.15 Attention globality distributions of BART (large) across different heads (sorted according to value) and averaged over all layers and 5000 data points. Blue curve stands for the no phase shift condition, and orange, green and red curves represent $k = 100, 200$ and 300 respectively.

Chapter 7

Conclusion

In this thesis, we have used *systematicity* as a tool to inspect and investigate the semantic and syntactic reasoning capabilities of state-of-the-art NLU models. In NLP, we witness a trend of ever-increasing size of large language models of the Transformer family, following the laws of scale [Kaplan et al., 2020]. These models tend to saturate established benchmarks quickly, even surpassing human-level performance [Kiela et al., 2021], and project a notion of human-level reasoning capability. However, in this thesis we re-evaluate those notions using the principles of systematicity. We observe in our results that such models fail to display robust, systematic human-like behaviors while processing natural language, which can cause severe issues in production settings. My work in this thesis attempts to highlight these issues; provides a mechanism to understand the black-box nature of the Transformer-family large language models; and allow us to evaluate their human-like generalization capability. The findings of this thesis will hopefully help the community to develop more robust and human-like NLU models in the future.

7.1 Summary

To summarize the major findings in this thesis:

- **Generalization via composition is hard, depends on the structure of input.** In Chapter §3 we observe that NLU models fail short of generalizing to systematic compositions of known rules. This highlights the models inability to reason by stitching known components, which is a key requirement for modular systems. We also observe the models to be highly sensitive to unwanted noise in the input, which a human can easily circumvent. Finally, we observe that one of the key hurdles for the model to achieve systematic generalization is its inability to comprehend the variability of syntax of natural language.

- **Models face word-order insensitivity issues in inference and pre-training.** In Chapter §4, we further find evidence of the poor syntactic encoding of state-of-the-art NLU models, by applying the notions of systematicity to word-order. Specifically, we find these models are surprisingly *in*-sensitive to drastic perturbations of word-order, which is undoubtedly one of the basic components needed for syntax understanding. We observe for downstream tasks such as MNLI [Williams et al., 2018c], a state-of-the-art model can still perform optimally (or sometimes even better) on inputs where word-order is scrambled, making the input devoid of the original semantic information. Furthermore, in Chapter §5, we observe the same models achieve near optimal downstream performance when pre-trained from scratch on word-order shuffled data. Thus, we observe word-order being a powerful tool to investigate the systematicity of model reasoning capability.

- **Syntax representation of large language models is considerably weaker than we thought.** Following the results of Chapter §4 and Chapter §5, one might be quick to conclude that Transformer-family of models have no syntactic abilities at all. However, that is not the case. Our results highlights a key aspect of the internal workings of Transformer-family of models: they do not require the same notions of classic syntax processing pipelines (pos-tagging, named-entity recognition, parsing, etc) as we thought them to have in order to achieve good downstream performance. Rather, their high performance can be explained by their distributional nature: ability to learn the n-gram statistics of tokens from the training copora. Our results can also highlight that fact that our current methods of evaluating syntax in these models is outdated, and we need better understanding of syntax evaluation mechanisms, such that they are co-related with the downstream task results.

- **A probable cause of weak syntax encoding is due to its poor understanding of relative positions.** Finally, in Chapter §6 we attempt to understand the reason models are weak in their syntax encoding. We observe that it can be partially explained by the poor relative position encoding scheme employed by the Transformer-family of models. Due to their lack of proper relative position understanding, these models are prone to different sentence processing behaviors when subjected to understanding a sentence within a context in isolation. We provide a thorough analysis and conjecture that the positional encoding schemes employed by these models should be revisited closely in order to improve their syntax representation abilities.

7.2 Limitations

In this section, we discuss the limitations of the research presented in this thesis as a whole. The goal of this thesis is to raise systematicity issues with the language processing mechanisms employed by NLU models. While our results raise significant number of questions, they are not exhaustive. Also, systematicity leads us to explore one specific subsection of the broad out-of-domain generalization literature, which could have many other potential avenues to investigate the inner workings of these models.

The biggest limitation with the dataset introduced in Chapter §3 stems due to its semi-synthetic nature. While synthetic and semi-synthetic datasets are useful in evaluating the model in a diagnostic setting, the dataset is not an ideal candidate to evaluate model performance in a real world setting, as we do not expect models to ever encounter such complex puzzles. Some of the natural language templates curated from Amazon Mechanical Turk might have undesired noise in them, which is unfortunately hard to eradicate in a crowd-sourced data annotation setting. Finally, at the time of publication of our paper [Sinha et al., 2019], we were only beginning to explore pre-trained Transformer family of models, hence we only included a small subset of it (BERT). It would be interesting to explore followup studies with modern Transformer-family of models (GPT2 [Radford et al., 2019a], RoBERTa [Liu et al., 2019b], DeBERTa [He et al., 2020]) to see if they still have such systematicity issues. Furthermore, post publication of this paper a new trend emerged in NLU, which consists of learning *in-context* from given examples (or prompts) using a massive language model, such as GPT3 or OPT. It would be further interesting to compare length generalization based systematicity issues with such in-context learning modes, which could perhaps be less amenable to the problem as they do not require any parameter updates (however there doesn't exist such a study to date, and it is an open question).

In our word-order related works (Chapter §4 and Chapter §5), one criticism we received is on the method of application of permutation *pre*-tokenization. Ravishankar et al. [2022] show that for some model families, the unnatural acceptability reduces significantly if the permutation is performed *post*-tokenization, i.e. directly on the BPE tokens. While the nature of experiments run by Ravishankar et al. [2022] is not directly comparable to our setting (they use a subset of tasks, pre-train a much smaller network on a significantly less amount of data), their results also corroborate ours. Post-tokenization can be viewed as sub-unigram permutations on individual tokens. Since the model never gets to learn those sub-unigrams in isolation, the permutation acceptability is also drastically lower, which is similar to our results from pre-training in Chapter §5. In a real-world scenario, it is much more probable for a model to encounter out-of-position words rather than tokens, which highlights the need for systematically investigating the root cause of permutation acceptance.

Our primary model type for exploration in Chapter §5 has been masked language models (MLM). It is conceivable to expect the same effect from Autoregresive language models, however it is an open question worthy of further inspection. Haviv et al. [2022] observe a surprising fact that Autoregressive models can perform optimally *even without position embeddings*. This highlights that Autoregressive models might have an internal representation of positions which is learned indirectly due to the causal language modelling objective. Thus, it could be a possibility that when trained on shuffled sentences these Autoregressive models face a position representation mismatch between train (shuffled) and inference (natural) word orders. Thus, it would be an interesting followup experiment to observe the same effects in such classes of models, to complete our understanding of the distributional nature of Transformer family of models.

Finally, our experiments on sub-sentence representation in Chapter §6 is solely limited to models having absolute position embeddings. We do not focus on the relative position embeddings [Shaw et al., 2018, Raffel et al., 2020] (RPE) as our method of phase-shift analysis is not applicable to those classes of models. RPEs employ a window based position information computation on the fly, which does not require it to store embeddings uniquely for each position. Thus, a phase shift in RPE would not change the sentence processing pipeline, as the model recomputes the position information based on the shifted window. Thus, we need different tools to study the relative position encoding of RPE than the one proposed in this work, which remains an open question.

7.3 Future Work

The work presented in this thesis opens up several exciting prospects for future work. A non-exhaustive set of ideas for future work based on top of our results is listed here.

Mutually exclusive learning A few years ago, Manning [2015] encouraged NLP to consider "the details of human language, how it is learned, processed, and how it changes, rather than just chasing state-of-the-art numbers on a benchmark task." In this thesis, we expand upon this view, and suggest one particular future direction: we should train models not only to do well on clean test data, but also be *consistent* in their understanding of corrupt, malformed or irrelevant data. For instance, a systematic model should avoid processing a malformed, corrupt or noisy input - instead it should provide mechanisms for early detection and exit. This concept is studied under the realm of *mutual exclusivity*, which has been shown to be a recurrent problem of neural networks [Gandhi and Lake, 2020]. A good future direction to alleviate the issues of word-order insensitivity would be to design word-order exclusive models, which will provide an early exit mechanism when they encounter corrupt input.

Leveraging distributional overlap to measure model performance Our results in Chapter §5 highlight the distributional characteristics which the large language models employ to achieve good model performance. The results suggests that the Transformer-family of models utilize their massive amount of parameters to memorize the statistics of n-gram distributions during pre-training. Recent results [Feldman and Zhang, 2020, Carlini et al., 2021, 2022] also corroborate the claim that models with larger amount of parameters memorize longer contiguous stretches of the input text. Future work can leverage this fact to devise tailored evaluation datasets for each model, such that the test distribution has the least amount of overlap with the n-gram distributions learned by the model.

Developing privacy sensitive NLU models The results from Chapter §4 and Chapter §5 also open an exciting future direction to develop privacy sensitive NLU models. Recent results increasingly highlight a growing concern for developing privacy sensitive models. Transformer models are shown to memorize large amount of training data, which can be extracted via prompting methods [Carlini et al., 2021, 2022], sensitive token information can be gained by attack vectors in embedding space [Song and Raghunathan, 2020], or by carefully crafted adversarial attacks [Henderson et al., 2018]. Current privacy preserving techniques typically involve extensive computation, and result in sub-optimal downstream task performance [Jayaraman and Evans, 2019]. Given our results, leveraging word-order to pre-train Transformer-family models can be a powerful alternative to develop privacy sensitive NLU models. Any prompting based attack vector will not be able to extract the same sentences used during training, as the input is itself shuffled and corrupted. A shuffled input would also result in retrieving sensitive information harder, as the context required to extract the same information is not in its original position in the training data. The resulting model would not lose downstream performance considerably, as the model only requires the distributional statistic to perform optimally (§5.4.1).

Bibliography

Anne Abeille. Lexical and syntactic rules in a Tree Adjoining Grammar. In *28th Annual Meeting of the Association for Computational Linguistics*, pages 292–298, Pittsburgh, Pennsylvania, USA, June 1990. Association for Computational Linguistics. doi: 10.3115/981823.981860. URL https://www.aclweb.org/anthology/P90-1037.

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. *arXiv preprint arXiv:1606.07356*, 2016.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *ICLR 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HJ4-rAVtl.

Matteo Alleman, Jonathan Mamou, Miguel A Del Rio, Hanlin Tang, Yoon Kim, and SueYeon Chung. Syntactic perturbations reveal representational correlates of hierarchical phrase structure in pretrained language models. *arXiv preprint arXiv:2104.07578*, 2021. URL https://arxiv.org/abs/2104.07578.

Aixiu An, Peng Qian, Ethan Wilcox, and Roger Levy. Representation of constituents in neural language models: Coordination phrase as a case study. 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 2888–2899, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1287. URL https://www.aclweb.org/anthology/D19-1287.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giri Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Mona T. Diab, Zornitsa Kozareva, and Ves Stoyanov. Efficient large scale language modeling with mixtures of experts. *CoRR*, abs/2112.10684, 2021. URL https://arxiv.org/abs/2112.10684.

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

Alan D Baddeley, Graham J Hitch, and Richard J Allen. Working memory and binding in sentence recall. *Journal of Memory and Language*, 2009. URL https://doi.org/10.1016/j.jml.2009.05.004.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkezXnA9YX.

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. Learning to few-shot learn across diverse natural language classification tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5108–5123, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi:

10.18653/v1/2020.coling-main.448. URL https://aclanthology.org/2020.coling-main.448.

Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008. doi: 10.1162/coli.2008.34.1.1. URL https://aclanthology.org/J08-1001.

Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 113–120, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics. URL https://aclanthology.org/N04-1015.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and alternatives. *arXiv preprint arXiv:2102.12452*, 2021.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020. URL https://arxiv.org/abs/2004.05150.

Douglas K Bemis and Liina Pylkkänen. Basic linguistic composition recruits the left anterior temporal lobe and left angular gyrus during both listening and reading. *Cerebral Cortex*, 2013.

Emily M. Bender and Alexander Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.463. URL https://www.aclweb.org/anthology/2020.acl-main.463.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In *TAC*, 2009. URL https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.232.1231&rep=rep1&type=pdf.

Jean-Phillipe Bernardy and Shalom Lappin. Using deep neural networks to learn syntactic agreement. In *Linguistic Issues in Language Technology, Volume 15, 2017*. CSLI Publications, 2017. URL https://www.aclweb.org/anthology/2017.lilt-15.3.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. English web treebank. *Linguistic Data Consortium, Philadelphia, PA*, 2012. URL https://catalog.ldc.upenn.edu/LDC2012T13.

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6239.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL https://doi.org/10.5281/zenodo.5297715. If you use this software, please cite it using these metadata.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Gregory Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Martin Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-neox-20b: An open-source autoregressive language model. In *Challenges & Perspectives in Creating Large Language Models*, 2022. URL https://openreview.net/forum?id=HL7IhzS8W5.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *JMLR*, 2003. URL https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://www.aclweb.org/anthology/D15-1075.

Joan Bresnan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler. *Lexical-functional syntax*. John Wiley & Sons, 2015.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference* *on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting Training Data from Large Language Models, June 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying Memorization Across Neural Language Models. *arXiv:2202.07646 [cs]*, February 2022.

James McKeen Cattell. The time it takes to see and name objects. *Mind*, os-XI(41):63–65, 01 1886. ISSN 0026-4423. doi: 10.1093/mind/os-XI.41.63. URL https://doi.org/10.1093/mind/os-XI.41.63.

Rui Chaves. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 1–11, New York, New York, January 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.scil-1.1.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1152. URL https://www.aclweb.org/anthology/P17-1152.

Gennaro Chierchia and Sally McConnell-Ginet. *Meaning and grammar: An Introduction to Semantics*. Cambridge, Ma: MIT Press, 1990.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

Noam Chomsky. *Syntactic structures*. Walter de Gruyter, 1957.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL `https://arxiv.org/abs/2204.02311`.

Grzegorz Chrupała and Afra Alishahi. Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2952–2962, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1283. URL `https://www.aclweb.org/anthology/P19-1283`.

Guglielmo Cinque. *Adverbs and functional heads: A cross-linguistic perspective*. Oxford University Press on Demand, 1999.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019b. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://www.aclweb.org/anthology/W19-4828.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020a.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *IJCAI*, 2020b.

Louis Clouatre, Prasanna Parthasarathi, Amal Zouaq, and Sarath Chandar. Local structure matters most: Perturbation study in NLU. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3712–3731, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-acl.293.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008. URL https://dl.acm.org/doi/10.1145/1390156.1390177.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45, 2003. URL https://www.aclweb.org/anthology/W03-0906.

Alexis Conneau and Douwe Kiela. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1269.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1070. URL https://www.aclweb.org/anthology/D17-1070.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018a. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties, 2018b.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.49. URL https://aclanthology.org/2021.emnlp-main.49.

Jillian Da Costa and Rui Chaves. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 12–21, New York, New York, January 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.scil-1.2.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*. Springer, 2006.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Syg-YfWCW.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. Evaluating compositionality in sentence embeddings. In *Proceedings of Annual Meeting of the Cognitive Science Society*, 2018. URL https://arxiv.org/abs/1802.04302.

Forrest Davis and Marten van Schijndel. Recurrent neural network language models always learn English-like relative clause attachment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1979–1990, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.179. URL https://www.aclweb.org/anthology/2020.acl-main.179.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124, 2019.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990. URL https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1097-4571(199009)41:6%3C391::AID-ASI1%3E3.0.CO;2-9.

Dorottya Demszky, Kelvin Guu, and Percy Liang. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*, 2018. URL https://arxiv.org/pdf/1809.02922.pdf.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://www.aclweb.org/anthology/I05-5002.pdf.

Pedro Domingos and Matthew Richardson. Markov logic networks. *Mach. Learn.*, 62 (1-2):107–136, 2006. ISSN 0885-6125.

Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017,* *Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Hk95PK9le.

Matthew S Dryer. The Greenbergian word order correlations. *Language*, pages 81–138, 1992.

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. Position information in transformers: An overview, 2021. URL https://arxiv.org/abs/2102.11090.

Sašo Džeroski. Relational data mining. In *Data Mining and Knowledge Discovery Handbook*, pages 887–911. Springer, 2009.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA, August 2018a. Association for Computational Linguistics. URL https://aclanthology.org/C18-1055.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia, July 2018b. Association for Computational Linguistics. doi: 10.18653/v1/P18-2006. URL https://aclanthology.org/P18-2006.

Allyson Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48, 2020. doi: 10.1162/tacl_a_00298. URL https://www.aclweb.org/anthology/2020.tacl-1.3.

Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M. Bender. Towards linguistically generalizable NLP systems: A workshop and shared task. In *Proceedings of* *the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 1–10, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5401. URL `https://aclanthology.org/W17-5401`.

Robert M Fano. *The transmission of information*. Massachusetts Institute of Technology, Research Laboratory of Electronics ..., 1949.

Zichu Fei, Qi Zhang, Tao Gui, Di Liang, Sirui Wang, Wei Wu, and Xuanjing Huang. CQG: A simple and effective controlled generation framework for multi-hop question generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6896–6906, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.acl-long.475`.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

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

Jonathan Frankle, David J Schwab, and Ari S Morcos. Training batchnorm and only batchnorm: On the expressive power of random features in cnns. *arXiv preprint arXiv:2003.00152*, 2020. URL `https://arxiv.org/abs/2003.00152`.

Gottlob Frege. Sense and reference. *The philosophical review*, 1948.

Herve Gallaire and Jack Minker. *Logic and Data Bases*. Perseus Publishing, 1978.

Kanishk Gandhi and Brenden M Lake. Mutual exclusivity as a challenge for deep neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14182–14192. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/a378383b89e6719e15cd1aa45478627c-Paper.pdf.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2021. URL https://doi.org/10.5281/zenodo.5371628.

Jon Gauthier and Roger Levy. Linking artificial and human neural representations of language. 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 529–539, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1050. URL https://aclanthology.org/D19-1050.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.10. URL https://www.aclweb.org/anthology/2020.acl-demos.10.

Dedre Gentner and Cecile Toupin. Systematicity and surface similarity in the development of analogy. *Cognitive science*, 10(3):277–300, 1986.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June 2007a. Association for Computational Linguistics. URL https://aclanthology.org/W07-1401.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9, 2007b. URL https://www.aclweb.org/anthology/W07-1401.pdf.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5426. URL https://aclanthology.org/W18-5426.

Goran Glavaš and Ivan Vulić. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.270.

Oren Glickman, Ido Dagan, and Moshe Koppel. Web based probabilistic textual entailment. 2005.

Yoav Goldberg. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*, 2019. URL https://arxiv.org/abs/1901.05287.

Nicolas Gontier, Koustuv Sinha, Siva Reddy, and Chris Pal. Measuring systematic generalization in neural proof generation with transformers. *Advances in Neural Information Processing Systems*, 33:22231–22242, 2020.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1958–1969, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.177. URL https://aclanthology.org/2020.acl-main.177.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada, 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1052.

Joseph Greenberg. Some universals of grammar with particular reference to the order of meaningful elements. *In J. Greenberg, ed., Universals of Language. 73-113. Cambridge, MA.*, 1963. URL https://wals.info/refdb/record/Greenberg-1963.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 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, June 2018a. Association for Computational Linguistics. doi: 10.18653/v1/N18-1108. URL https://www.aclweb.org/anthology/N18-1108.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. Colorless green recurrent networks dream hierarchically. *arXiv:1803.11138 [cs]*, March 2018b. URL http://arxiv.org/abs/1803.11138.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. BERT & family eat word salad: Experiments with text understanding. *AAAI*, 2021. URL https://arxiv.org/abs/2101.03453.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://www.aclweb.org/anthology/N18-2017.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.8552&rep=rep1&type=pdf.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.659. URL https://aclanthology.org/2020.acl-main.659.

Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. Embedding logical queries on knowledge graphs. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2026–2037. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7473-embedding-logical-queries-on-knowledge-graphs.pdf.

Kyoung-Soo Han, Young-In Song, and Hae-Chang Rim. Probabilistic model for definitional question answering. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 212–219, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148210. URL http://doi.acm.org/10.1145/1148170.1148210.

Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954. URL https://www.tandfonline.com/doi/pdf/10.1080/00437956.1954.11659520.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. Transformer Language Models without Positional Encodings Still Learn Positional Information. *ArXiv preprint*, abs/2203.16634, 2022. URL https://arxiv.org/abs/2203.16634.

Robert Hawkins, Takateru Yamakoshi, Thomas Griffiths, and Adele Goldberg. Investigating representations of verb bias in neural language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4653–4663, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.376. URL https://www.aclweb.org/anthology/2020.emnlp-main.376.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

Irene Heim and Angelika Kratzer. *Semantics in generative grammar*. Blackwell Oxford, 1998.

Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Genevieve Fried, Ryan Lowe, and Joelle Pineau. Ethical challenges in data-driven dialogue systems. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 123–129, 2018.

Aurelie Herbelot. How to stop worrying about compositionality. *The Gradient*, 2020.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015.

Jack Hessel and Alexandra Schofield. How effective is BERT without word ordering? implications for language understanding and data privacy. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 204–211, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.27. URL https://aclanthology.org/2021.acl-short.27.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://www.aclweb.org/anthology/N19-1419.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

Douglas R Hofstadter. *Gödel, escher, bach*. Vintage Books New York, 1980.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL https://doi.org/10.5281/zenodo.1212303.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence Moss. OCNLI: Original Chinese Natural Language Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3512–3526, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.314. URL https://www.aclweb.org/anthology/2020.findings-emnlp.314.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 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, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.158. URL https://www.aclweb.org/anthology/2020.acl-main.158.

Drew Arad Hudson and Christopher D. Manning. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1Euwz-Rb.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926, 2018.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality Decomposed: How do Neural Networks Generalise? *jair*, 67:757–795, April 2020. ISSN 1076-9757. doi: 10.1613/jair.1.11674.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://www.aclweb.org/anthology/P19-1356.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356.

Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912, 2019.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition.

In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8690–8705, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.768. URL https://www.aclweb.org/anthology/2020.acl-main.768.

Robin Jia and Percy Liang. 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, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://aclanthology.org/D17-1215.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

Martin Joos. Description of language design. *The Journal of the Acoustical Society of America*, 22(6):701–707, 1950.

Jaap Jumelet and Dieuwke Hupkes. 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, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5424. URL https://aclanthology.org/W18-5424.

Jaap Jumelet, Willem Zuidema, and Dieuwke Hupkes. Analysing neural language models: Contextual decomposition reveals default reasoning in number and gender assignment. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 1–11, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1001. URL https://aclanthology.org/K19-1001.

Jaap Jumelet, Milica Denic, Jakub Szymanik, Dieuwke Hupkes, and Shane Steinert-Threlkeld. Language models use monotonicity to assess NPI licensing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4958–4969, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.439. URL https://aclanthology.org/2021.findings-acl.439.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Ronald M Kaplan and Joan Bresnan. Formal system for grammatical representation. *Formal Issues in Lexical-Functional Grammar*, 1995.

Dimitri Kartsaklis, Mohammad Taher Pilehvar, and Nigel Collier. Mapping text to knowledge graph entities using multi-sense lstms. *arXiv preprint arXiv:1808.07724*, 2018.

Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.698. URL https://aclanthology.org/2020.acl-main.698.

Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without Natural Images. In *Proceedings of the Asian Conference on Computer Vision*, 2020. URL https://openaccess.thecvf.com/content/ACCV2020/papers/Kataoka_Pre-training_without_Natural_Images_ACCV_2020_paper.pdf.

Divyansh Kaushik and Zachary C Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, 2018.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-Deep Networks: Understanding and mitigating network overthinking. In *Proceedings of the 2019 International Conference on Machine Learning (ICML)*, Long Beach, CA, Jun 2019. URL https://arxiv.org/abs/1810.07052.

Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pretraining. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=09-528y2Fgf.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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 252–262, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1023. URL https://aclanthology.org/N18-1023.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi

Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.324. URL `https://aclanthology.org/2021.naacl-main.324`.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL `https://aclanthology.org/D14-1181`.

Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1340. URL `https://aclanthology.org/P19-1340`.

Shun Kiyono, Sosuke Kobayashi, Jun Suzuki, and Kentaro Inui. SHAPE: Shifted absolute position embedding for transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3309–3321, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.266. URL `https://aclanthology.org/2021.emnlp-main.266`.

Jordan Kodner and Nitish Gupta. Overestimation of syntactic representation in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1757–1762, Online, July 2020. Association for

Computational Linguistics. doi: 10.18653/v1/2020.acl-main.160. URL https://www.aclweb.org/anthology/2020.acl-main.160.

Robert Kowalski. *Logic for problem solving*, volume 7. Ediciones Díaz de Santos, 1979.

Angelika Kratzer and Irene Heim. *Semantics in generative grammar*, volume 1185. Blackwell Oxford, 1998.

Philippe Laban, Luke Dai, Lucas Bandarkar, and Marti A. Hearst. Can transformer models measure coherence in text: Re-thinking the shuffle test. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1058–1064, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.134. URL https://aclanthology.org/2021.acl-short.134.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*, 2018. URL http://proceedings.mlr.press/v80/lake18a/lake18a.pdf.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. The emergence of number and syntax units in LSTM language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 11–20, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1002. URL https://aclanthology.org/N19-1002.

Yair Lakretz, Dieuwke Hupkes, Alessandra Vergallito, Marco Marelli, Marco Baroni, and Stanislas Dehaene. Mechanisms for handling nested dependencies in neural-network language models and humans. *Cognition*, page 104699, 2021. ISSN 0010-

0277. doi: https://doi.org/10.1016/j.cognition.2021.104699. URL https://www.sciencedirect.com/science/article/pii/S0010027721001189.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997. URL https://psycnet.apa.org/record/1997-03612-001.

Hector J. Levesque, Ernest Davis, and L. Morgenstern. The winograd schema challenge. In *KR*, 2011. URL http://commonsensereasoning.org/2011/papers/Levesque.pdf.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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, 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Peiguang Li, Hongfeng Yu, Wenkai Zhang, Guangluan Xu, and Xian Sun. SA-NLI: A supervised attention based framework for natural language inference. *Neurocomputing*, 2020. URL https://doi.org/10.1016/j.neucom.2020.03.092.

Yongjie Lin, Yi Chern Tan, and Robert Frank. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4825. URL https://www.aclweb.org/anthology/W19-4825.

Tal Linzen and Marco Baroni. Syntactic structure from deep learning. *Annual Review of Linguistics*, 2021. URL https://doi.org/10.1146/annurev-linguistics-032020-051035.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016. doi: 10.1162/tacl_a_00115. URL https://www.aclweb.org/anthology/Q16-1037.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019a. URL http://arxiv.org/abs/1907.11692.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv*, 2019b. URL https://arxiv.org/abs/1907.11692.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8: 726–742, 2020.

Hans Peter Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317, 1957.

Ziyang Luo, Artur Kulmizev, and Xiaoxi Mao. Positional artefacts propagate through masked language model embeddings. 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 5312–5327, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.413. URL https://aclanthology.org/2021.acl-long.413.

Bill MacCartney and Christopher D. Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/W07-1431.

Bill MacCartney and Christopher D Manning. An extended model of natural logic. In *Proceedings of the Eighth International Conference on Computational Semantics*, IWCS-8 '09, pages 140–156, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 9789074029346.

Nikolay Malkin, Sameera Lanka, Pranav Goel, and Nebojsa Jojic. Studying word order through iterative shuffling. In *EMNLP*, 2021.

Christopher D Manning. Computational linguistics and deep learning. *Computational Linguistics*, 2015.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1907367117. URL https://www.pnas.org/content/117/48/30046.

Gary F Marcus. Rethinking eliminative connectionism. *Cognitive psychology*, 37(3):243–282, 1998.

Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1151. URL https://www.aclweb.org/anthology/D18-1151.

Brian W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et biophysica acta*, 405 2:442–51, 1975. doi: https://doi.org/10.1016/0005-2795(75)90109-9. URL https://www.sciencedirect.com/science/article/pii/0005279575901099.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. *arXiv:1902.01007 [cs]*, June 2019a. URL http://arxiv.org/abs/1902.01007.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and*

Interpreting Neural Networks for NLP, pages 217–227, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.21. URL `https://www.aclweb.org/anthology/2020.blackboxnlp-1.21`.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 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, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL `https://www.aclweb.org/anthology/P19-1334`.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a. URL `http://arxiv.org/abs/1301.3781`.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013b. URL `https://arxiv.org/pdf/1310.4546.pdf`.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. Parlai: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84, 2017.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782, 2013.

Pasquale Minervini and Sebastian Riedel. Adversarially regularising neural NLI models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 65–74, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-1007. URL https://aclanthology.org/K18-1007.

Pasquale Minervini, Sebastian Riedel, Pontus Stenetorp, Edward Grefenstette, and Tim Rocktäschel. Learning reasoning strategies in end-to-end differentiable proving. In *International Conference on Machine Learning*, pages 6938–6949. PMLR, 2020.

Kanishka Misra. minicons: Enabling flexible behavioral and representational analyses of transformer language models. *ArXiv preprint*, abs/2203.13112, 2022. URL https://arxiv.org/abs/2203.13112.

Kanishka Misra, Allyson Ettinger, and Julia Rayz. Exploring BERT's sensitivity to lexical cues using tests from semantic priming. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4625–4635, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.415. URL https://aclanthology.org/2020.findings-emnlp.415.

Francis Mollica, Matthew Siegelman, Evgeniia Diachek, Steven T Piantadosi, Zachary Mineroff, Richard Futrell, Hope Kean, Peng Qian, and Evelina Fedorenko. Composition is the core driver of the language-selective network. *Neurobiology of Language*, 1(1):104–134, 2020. URL https://direct.mit.edu/nol/article/1/1/104/10024/Composition-is-the-Core-Driver-of-the-Language.

Richard Montague. Universal grammar. *Theoria*, 1970.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL-HLT*, pages 839–849, 2016.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1198.

Aakanksha Naik, Abhilasha Ravichander, Carolyn Rose, and Eduard Hovy. Exploring numeracy in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3374–3380, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1329. URL https://www.aclweb.org/anthology/P19-1329.

Nikita Nangia and Samuel R. Bowman. Human vs. muppet: A conservative estimate of human performance on the GLUE benchmark. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4566–4575, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1449. URL https://www.aclweb.org/anthology/P19-1449.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. Compositional vector space models for knowledge base completion. *arXiv preprint arXiv:1504.06662*, 2015.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 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, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL https://www.aclweb.org/anthology/2020.acl-main.441.

Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.251. URL https://aclanthology.org/2022.acl-long.251.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://www.aclweb.org/anthology/N19-4009.

Isabel Papadimitriou and Dan Jurafsky. Learning Music Helps You Read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6829–6839, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.554. URL https://aclanthology.org/2020.emnlp-main.554.

Isabel Papadimitriou, Ethan A. Chi, Richard Futrell, and Kyle Mahowald. Deep subjecthood: Higher-order grammatical features in multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2522–2532, Online, April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.eacl-main.215.

Prasanna Parthasarathi, Koustuv Sinha, Joelle Pineau, and Adina Williams. Sometimes we want ungrammatical translations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3205–3227, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.275. URL https://aclanthology.org/2021.findings-emnlp.275.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. Rissanen data analysis: Examining dataset characteristics via description length. In *ICML*, 2021a.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. Rissanen Data Analysis: Examining Dataset Characteristics via Description Length. In *Proceedings of the Thirty-eighth International Conference on Machine Learning (ICML)*, March 2021b.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational

Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Thang M Pham, Trung Bui, Long Mai, and Anh Nguyen. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? *arXiv preprint arXiv:2012.15180*, 2020. URL https://arxiv.org/abs/2012.15180.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning 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 1267–1273, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1128. URL https://aclanthology.org/N19-1128.

Tiago Pimentel, Naomi Saphra, Adina Williams, and Ryan Cotterell. Pareto probing: Trading off accuracy for complexity. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3138–3153, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.254. URL https://aclanthology.org/2020.emnlp-main.254.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.420. URL https://aclanthology.org/2020.acl-main.420.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-2023. URL https://www.aclweb.org/anthology/S18-2023.

Carl Pollard and Ivan A Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994. URL https://web.stanford.edu/group/cslipublications/cslipublications/site/0226674479.shtml.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1007. URL https://www.aclweb.org/anthology/K19-1007.

Ofir Press, Noah A. Smith, and Mike Lewis. Shortformer: Better language modeling using shorter inputs. 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 5493–5505, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.427. URL https://aclanthology.org/2021.acl-long.427.

Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

Geoffrey K Pullum, Barbara C Scholz, et al. Recursion and the infinitude claim. *Recursion in human language*, 104:113–38, 2010.

Liina Pylkkänen, Douglas K Bemis, and Estibaliz Blanco Elorrieta. Building phrases in language production: An meg study of simple composition. *Cognition*, 2014.

J R Quinlan. Learning logical definitions from relations. *Mach. Learn.*, 5(3):239–266, August 1990. ISSN 0885-6125. doi: 10.1007/BF00117105.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019a. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019b. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140): 1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021. URL https://openreview.net/forum?id=Gl8FHfMVTZu.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. Can LSTM learn to capture agreement? the case of Basque. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 98–107, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5412. URL https://www.aclweb.org/anthology/W18-5412.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. Studying the inductive biases of RNNs with synthetic variations of natural languages. 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 3532–3542, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1356. URL https://www.aclweb.org/anthology/N19-1356.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language*

Learning (CoNLL), pages 349–361, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1033. URL https://www.aclweb.org/anthology/K19-1033.

Vinit Ravishankar, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. Multilingual ELMo and the effects of corpus sampling. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 378–384, Reykjavik, Iceland (Online), May 31–2 June 2021. Linköping University Electronic Press, Sweden. URL https://aclanthology.org/2021.nodalida-main.41.

Vinit Ravishankar, Mostafa Abdou, Artur Kulmizev, and Anders Søgaard. Word order does matter (and shuffled language models know it). *arXiv preprint arXiv:2203.10995*, 2022.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, 2013.

Jorma Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information theory*, 30(4):629–636, 1984. URL https://ieeexplore.ieee.org/document/1056936.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and

Phil Blunsom. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*, September 2015.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95, 2011.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349. URL https://aclanthology.org/2020.tacl-1.54.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6399.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Masked Language Model Scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.240.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://arxiv.org/abs/1910.01108.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. Relational recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 7310–7321, 2018.

Yves Schabes, Anne Abeille, and Aravind K. Joshi. Parsing strategies with 'lexicalized' grammars: Application to Tree Adjoining Grammars. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*, 1988. URL https://www.aclweb.org/anthology/C88-2121.

Eckart Scheerer. Early german approaches to experimental reading research: The contributions of wilhelm wundt and ernst meumann. *Psychological Research*, 1981.

Christian Scheible and Hinrich Schütze. Cutting recursive autoencoder trees. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations (ICLR) Scottsdale, Arizona, USA, May 2-4, 2013, Conference Track Proceedings*, 2013. URL https://arxiv.org/pdf/1301.2811.pdf.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, pages 3776–3784, 2016.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301, 2017.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL https://aclanthology.org/N18-2074.

Sheng Shen, Alexei Baevski, Ari Morcos, Kurt Keutzer, Michael Auli, and Douwe Kiela. Reservoir transformers. 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 4294–4309, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.331. URL https://aclanthology.org/2021.acl-long.331.

Natalia Silveira, Timothy Dozat, Marie-Catherine De Marneffe, Samuel R Bowman, Miriam Connor, John Bauer, and Christopher D Manning. A gold standard dependency corpus for english. In *LREC*, pages 2897–2904. Citeseer, 2014. URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/1089_Paper.pdf.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. CLUTRR: A diagnostic benchmark for inductive reasoning from text. 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 4506–4515, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1458. URL https://aclanthology.org/D19-1458.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.230. URL https://aclanthology.org/2021.emnlp-main.230.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. UnNatural Language Inference. 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 7329–7346, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.569. URL https://aclanthology.org/2021.acl-long.569.

Koustuv Sinha, Amirhossein Kazemnejad, Siva Reddy, Joelle Pineau, Dieuwke Hupkes, and Adina Williams. The curious case of positional embeddings. *Empirical Methods of Natural Language Processing (EMNLP) Findings*, 2022.

Paul Smolensky. The constituent structure of connectionist mental states: A reply to fodor and pylyshyn. In *Connectionism and the philosophy of mind*, pages 281–308. Springer, 1991.

Joshua Snell and Jonathan Grainger. The sentence superiority effect revisited. *Cognition*, 2017.

Joshua Snell and Jonathan Grainger. Word position coding in reading is noisy. *Psychonomic bulletin & review*, 26(2):609–615, 2019.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 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, 2013. URL https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf.

Shagun Sodhani, Sarath Chandar, and Yoshua. Bengio. On Training Recurrent Neural Networks for Lifelong Learning. *arXiv e-prints*, November 2018.

Congzheng Song and Ananth Raghunathan. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 377–390, 2020.

Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gur, Zenghui Yan, and Xifeng Yan. On generating characteristic-rich question sets for qa evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 562–572, 2016.

Zoltan Szabó. The case for compositionality. *The Oxford handbook of compositionality*, 64: 80, 2012.

Whitney Tabor. *Syntactic innovation: A connectionist model*. PhD thesis, 1994. URL https://www.proquest.com/openview/0a8f7e8a71e058b12053b545ca857fb2/1.

Ronen Tamari, Kyle Richardson, Noam Kahlon, Aviad Sar-shalom, Nelson F. Liu, Reut Tsarfaty, and Dafna Shahaf. Dyna-bAbI: unlocking bAbI's potential with dynamic synthetic benchmarking. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 101–122, Seattle, Washington, July 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.starsem-1.9.

Ajinkya Tejankar, Bichen Wu, Saining Xie, Madian Khabsa, Hamed Pirsiavash, and

Hamed Firooz. A fistful of words: Learning transferable visual models from bag-of-words supervision. *ArXiv*, abs/2112.13884, 2021.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL `https://www.aclweb.org/anthology/P19-1452`.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.303. URL `https://aclanthology.org/2021.emnlp-main.303`.

Hiroshi Toyota. Changes in the constraints of semantic and syntactic congruity on memory across three age groups. *Perceptual and Motor Skills*, 2001. URL `https://pubmed.ncbi.nlm.nih.gov/11453195/`.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. *arXiv preprint*, pages 1–12, 2016. doi: 10.1016/B978-0-12-800077-9/00020-7.

Masatoshi Tsuchiya. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL `https://www.aclweb.org/anthology/L18-1239`.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018. URL https://arxiv.org/pdf/1711.10925.pdf.

Johan Van Benthem. *Language in Action: categories, lambdas and dynamic logic*. MIT Press, 1995.

Johan Van Benthem et al. *A brief history of natural logic*. LondonCollege Publications9781904987444, 2008.

Marten van Schijndel and Tal Linzen. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1499. URL https://www.aclweb.org/anthology/D18-1499.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5831–5837, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1592. URL https://www.aclweb.org/anthology/D19-1592.

Lucy Vanderwende and William B Dolan. What syntax can contribute in the entailment task. In *Machine Learning Challenges Workshop*. Springer, 2005. URL https://dl.acm.org/doi/10.1007/11736790_11.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.14. URL https://aclanthology.org/2020.emnlp-main.14.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1221. URL https://www.aclweb.org/anthology/D19-1221.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://www.aclweb.org/anthology/W18-5446.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, 2019. URL https://papers.nips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Ben Wang. Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. On position embeddings in BERT. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=onxoVA9FxMw.

Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*, 2016.

Yu-An Wang and Yun-Nung Chen. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6840–6849, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.555. URL https://aclanthology.org/2020.emnlp-main.555.

Z. Wang, L. Li, D. D. Zeng, and Y. Chen. Attention-based multi-hop reasoning for knowledge graph. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 211–213, Nov 2018. doi: 10.1109/ISI.2018.8587330.

Alex Warstadt and Samuel R Bowman. Can neural networks acquire a structural bias from raw linguistic data? In *Proceedings of the 42nd Annual Virtual Meeting of the Cognitive Science Society*, 2020. URL https://arxiv.org/abs/2007.06761.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 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, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1286. URL https://www.aclweb.org/anthology/D19-1286.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019b. URL https://arxiv.org/pdf/1805.12471.pdf.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, March 2019c. doi: 10.1162/tacl_a_00290. URL https://www.aclweb.org/anthology/Q19-1040.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7: 625–641, 2019d. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020a. doi: 10.1162/tacl_a_00321. URL https://aclanthology.org/2020.tacl-1.25.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020b. doi: 10.1162/tacl_a_00321. URL https://www.aclweb.org/anthology/2020.tacl-1.25.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online, November 2020c. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.16. URL https://aclanthology.org/2020.emnlp-main.16.

Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States, July 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.naacl-main.167.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association of Computational Linguistics*, 6:287–302, 2018.

Yun Wen, Joshua Snell, and Jonathan Grainger. Parallel, cascaded, interactive processing of words during sentence reading. *Cognition*, 2019.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards AI-Complete question answering: A set of prerequisite toy tasks. 2015. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2014.09.131.

Aaron Steven White, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Lexicosyntactic inference in neural models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4717–4724, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1501. URL https://www.aclweb.org/anthology/D18-1501.

John Wieting and Douwe Kiela. No training required: Exploring random encoders for sentence classification. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=BkgPajAcY7.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5423. URL https://www.aclweb.org/anthology/W18-5423.

Adina Williams, Andrew Drozdov, and Samuel R. Bowman. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics*, 6:253–267, 2018a. doi: 10.1162/tacl_a_00019. URL https://www.aclweb.org/anthology/Q18-1019.

Adina Williams, Andrew Drozdov, and Samuel R Bowman. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association* *for Computational Linguistics*, 6:253–267, 2018b. URL https://www.aclweb.org/anthology/Q18-1019/.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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, June 2018c. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. Harcourt, Brace & Company, Inc., 1922.

Thomas Wolf. Some additional experiments extending the tech report "Assessing BERT's Syntactic Abilities" by Yoav Goldberg. page 7, 2019a. URL https://huggingface.co/bert-syntax/extending-bert-syntax.pdf.

Thomas Wolf. Some additional experiments extending the tech report" assessing berts syntactic abilities" by yoav goldberg. Technical report, HuggingFace, 2019b. URL https://huggingface.co/bert-syntax/extending-bert-syntax.pdf.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *EMNLP: System Demonstrations*, 2020a.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings* *of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.383. URL https://www.aclweb.org/anthology/2020.acl-main.383.

Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. Exploring randomly wired neural networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1284–1293, 2019. URL https://arxiv.org/pdf/1904.01569.pdf.

Wenhan Xiong, Thien Hoang, and William Yang Wang. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573, 2017.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. One-shot relational learning for knowledge graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1980–1990, 2018.

Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. SyGNS: A systematic generalization testbed based on natural language semantics. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 103–119, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.10. URL https://aclanthology.org/2021.findings-acl.10.

Lang Yu and Allyson Ettinger. On the interplay between fine-tuning and composition in transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2279–2293, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.201. URL https://aclanthology.org/2021.findings-acl.201.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. On the paradox of learning to reason from data. *ArXiv*, abs/2205.11502, 2022a.

Kelly Zhang and Samuel Bowman. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5448. URL https://www.aclweb.org/anthology/W18-5448.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, 2018.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models. *ArXiv*, abs/2205.01068, 2022b. doi: 10.48550/ARXIV.2205.01068. URL https://arxiv.org/abs/2205.01068.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample BERT fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, March 2021. URL https://openreview.net/forum?id=cO1IH43yUF.

Yu Zhang, Zhenghua Li, and Min Zhang. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.302. URL https://aclanthology.org/2020.acl-main.302.

Yuan Zhang, Jason Baldridge, and Luheng He. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1131. URL https://aclanthology.org/N19-1131.

Han Zhao, Zhengdong Lu, and Pascal Poupart. Self-adaptive hierarchical sentence model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 4069–4076, 2015. URL https://arxiv.org/abs/1504.05070.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei.

Bert loses patience: Fast and robust inference with early exit. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18330–18341. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d4dd111a4fd973394238aca5c05bebe3-Paper.pdf.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015. URL https://arxiv.org/pdf/1506.06724.pdf.

Acronyms

APE Absolute Position Encodings. 130–132, 134, 135, 141, 142, 146, 150–153

MDL Minimum Description Length. xiv, 87, 114, 115

MLM Masked Language Models. 20, 21, 84, 89–92, 95, 102, 107, 112, 124, 126, 127, 132, 141, 152, 153, 165

NLU Natural Language Understanding. xxiii–xxv, xxvii, 2–5, 12, 13, 22, 24, 25, 28–30, 42, 49–51, 54–61, 73, 85–87, 121, 161–164, 167

RNN Recurrent Neural Networks. xii, 14, 16, 19, 24, 82, 83, 85, 87