FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Michael Hanna

Investigating Large Language Models' Representations Of Plurality Through Probing Interventions

Institute of Formal and Applied Linguistics

Supervisor of the master thesis:	RNDr. David Mareček, Ph.D.
Study programme:	Computer Science
Study branch:	Computational Linguistics

Prague 2022

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

Sincere thanks to my supervisors, Prof. Roberto Zamparelli at the University of Trento and Prof. David Mareček at Charles University, for all of their help and support despite my sometimes-erratic work schedule and deadline challenges. Thanks also to Profs. Ondřej Bojar and Raffaella Bernardi, who helped and supported me through other research projects and papers. Many thanks to the many friends I have made through LCT, especially those who have come to visit from afar, and those who have journeyed places with me. A big thanks to my family for their support through my long absences from home, and their willingness to come quite a ways to visit. Special thanks to my godmother Yvonne, who has kindly hosted me in her home in Amsterdam, and provided me with good company when I needed it the most! Finally, thanks to the EM LCT program (and the EU) for their financial support for this degree.

Title: Investigating Large Language Models' Representations Of Plurality Through Probing Interventions

Author: Michael Hanna

Institute: Institute of Formal and Applied Linguistics

Supervisor: RNDr. David Mareček, Ph.D., Institute of Formal and Applied Linguistics

Abstract: Large language models (LLMs) have become ubiquitous in natural language processing, but how exactly they process their input and arrive at good downstream task performance is still poorly understood. While much work has been done using probing to examine LLM internals, or behavioral studies, to determine LLMs' linguistic capabilities, these techniques are too weak to allow us to draw conclusions how LLMs process language. In this paper, I use both probing and causal intervention methods to investigate the question of subject-verb agreement with respect to the subject's plurality. I find that while probing reveals that subject plurality information is distributed throughout a sentence, causal interventions suggest that only information stored in linguistically relevant tokens is used. Probing interventions suggest that some but not all probes capture information in a way that reflects LLMs' usage thereof.

Keywords: Interpretability, Probing, Natural Language Processing, Computational Linguistics

Contents

1. Introduction

Since the inception of artificial intelligence research, the field has aimed to construct systems whose language abilities match those of humans. Much effort has been expended in the engineering of such systems, and with notable success: in the past five years, models have become better at translating between languages [Vaswani et al., 2017], answering questions [Devlin et al., 2019], and generating text [Brown et al., 2020]. Large language models (LLMs) in particular have driven progress in this field, with impressive few-shot generalization abilities [Chowdhery et al., 2022], and successes on simple reasoning tasks [Wei et al., 2022].

Still, these successes are not a good way to determine whether or not such systems have acquired human-level language abilities. While many tasks or benchmarks might seem to require language faculties, their implementations are often flawed, leaving them solvable by simple heuristics alone [McCoy et al., 2019, Sen and Saffari, 2020]. Moreover, simple metrics like task accuracy do not give the fine-grained information needed to judge models' language abilities. As an alternative to benchmarks, past work has investigated this question from primarily one of two perspectives. Studies taking a behavioral approach use carefully crafted datasets to elicit model behavior in specific scenarios, uncovering how models process various linguistic phenomena. In contrast, studies taking an internal or structural approach look inside models, examining representations for traces of linguistically-relevant information that models have learned.

Unfortunately, there exists a significant gap between these two types of studies. Behavioral studies can tell us concretely how models perform on certain tasks, providing us with strong negative evidence when a model fails on a task. However, if a model succeeds, they cannot tell us how it performs the task of interest, or causally, what underlying rules are encoded in it. On the other hand, internal studies allow us to look into the internal computations and representations of the model. But, many internal analyses only search for linguistic information within model representations, without ensuring that this information is actually used and causally contributes to the functioning of the model.

In this thesis, I work to bridge the gap between these types of analyses. To do so, I apply and combine existing interventions to a simple question: large language models' processing of subject nouns' plurality in the context of verb agreement. I make the following contributions:

- I show that probes find subject plurality information in all words of a sentence, not only the subject as expected.
- I use geometric probing interventions to reveal that only some of these probes capture subject plurality information that LLMs actually use.
- I furthermore apply swap interventions to give an upper bound on geometric probing interventions' effect, and show that our geometric interventions were effective.
- I show that when processing subject-verb agreement, LLMs use only plurality information encoded in the positions of the subject and words agreeing with it, even though that information is available elsewhere.

2. Background

In this chapter, I provide background information on the technical and theoretical underpinnings of this project. I will start by unpacking the problem of word representations in natural language processing (NLP). Next, I will describe neural networks, the dominant machine learning model used in NLP. I will then describe large language models (LLMs), the specific type of neural network that has driven recent advances in NLP. Following that, I describe techniques used to evaluate LLMs linguistic knowledge. Finally, I will introduce causal interventions in pre-trained language models, which allow us to draw connections between LLMs' internal representations and external behavior.

2.1 Word Representations in NLP

The question of how to create meaningful representations of words, sentences, and texts is one of the core issues at the heart of computational linguistics and NLP. Naturally, machines can represent words as strings, but this only tells us the written form of a word. In order to perform even relatively simple tasks like sentiment analysis, we need word representations that capture word syntax and semantics. Moreover, we need representations that can be implemented and manipulated computationally.

Various discrete, hand-crafted word representations have emerged to meet these needs; some are still in common use. WordNet [Fellbaum, 1998], based in psycholinguistic research, represents words as nodes in a graph. Each word is connected to other words by various semantic relations: words in the same "synset" are the same class of word; words can be hypernyms (supercategories) or hyponyms (subcategories) of other words. FrameNet, in contrast, collects "frames" in which verbs are used, as well as the different roles taken on by their arguments [Baker et al., 1998]. These are just two of many repositories of word information, in addition to the myriad other word representations proposed by linguists but not fully implemented.

These representations are valuable, but labor-intensive; they require some human annotation. Instead, it would be ideal to automatically generate meaningful representations of language from text. In order to build these representations, many early works in this direction relied on the distributional hypothesis: words with similar meanings have similar distributional properties throughout a corpus. A common approach was thus to represent a word w as a co-occurrence vector, where each entry in the vector was the number of times w appeared in the proximity of another given word in the corpus. Similar approaches use instead the number of times w appeared in a specific document, or some normalized / otherwise transformed version of this.

This style of word representation has various benefits. In addition to being automatically generated, these representations are real-valued vectors, which can be used as input for machine learning models. Moreover, real-valued vectors allow for a variety of operations to be performed on them, which correspond to real-world linguistic properties. For example, the distance between two word vectors can be interpreted as their similarity; the nearest neighbors of a word can be interpreted as those that are the most semantically similar.

Based on this, these word vectors could be employed for various tasks. Schütze [1992] uses semi-contextual learned word vectors to perform word sense disambiguation: many instances of a polysemous word are organized into clusters via a clustering algorithm, and each cluster is manually labeled with a sense. Each new instance of that word is assigned the sense of the cluster nearest to it. Landauer and Dumais [1997] create word representations by performing singular vector decomposition on document-frequency vectors of words. They then use these representations to identify a word's synonym from a list of candidates; the candidate whose representation had the highest cosine similarity with the target's representation was chosen as the synonym.

These approaches were succeeded by word representations with larger vocabularies, useful for a wider variety of tasks. Glove [Pennington et al., 2014] vectors, created using co-occurrence statistics of billions of tokens, were trained to predict co-occurrence ratios. Word2Vec [Mikolov et al., 2013] vectors are similarly trained on large data, and consist of the weights of a linear model used to predict which tokens co-occur with a given input token.

Both of these models could perform well on pre-existing tasks (like finding similar words via a nearest neighbors search) and on new tasks as well. Attracting great attention was the "analogy" task: models could solve analogies of the form u is to v as w is to "____" using simple addition and subtraction. The answer could be computed as the nearest neighbor of the expression $w + (v - u)$. Famously, word2vec could compute that $woman + (king - man) = queen$, i.e. "man" is to "king" as "woman" is to "queen".

Successes of this sort generated great enthusiasm for this style of word representations, and their potential to encode linguistic relations via vector-space geometry. Further work aimed to make these vector spaces more compositional, for example by modeling adjective-noun composition [Baroni and Zamparelli, 2010, Fyshe et al., 2015], or even phrase and sentence composition more generally [Socher et al., 2012]. Other work attempted to unify these vector representations with lexical resources like WordNet [Yu and Dredze, 2014, Faruqui et al., 2015]. In sum, there was significant interest in these vector representations of words, both for their capability to represent semantically meaningful information about words, and for their practical usefulness within NLP.

Despite the popularity of this style of representation learning at its peak, it is no longer the dominant mode of representation learning in NLP. Rather, deep learning models trained on vast corpora, using complex architectures and objectives, automatically learn word representations while performing other tasks.

Unlike the representations discussed here, deep models' word representations do not necessarily conform to linguistic intuition regarding nearest neighbors or analogies. Instead these representations, which play a crucial role in allowing deep learning models to achieve high performance on linguistic tasks, are opaque. They do not obviously encode any linguistic information, and do not always allow for vector-space geometry as discussed above. Thus, exact information encoded in these representations is still an open question; indeed it is a major focus of this paper. In the following section, I discuss deep neural networks in NLP, noting also the ways in which these networks generate these representations.

2.2 Neural Networks in NLP

In this section, I provide a very brief overview of neural networks in NLP. I wish to highlight the most important architectures, as well as their salient characteristics with respect to language processing and representation learning specifically.

2.2.1 Feed-forward Networks

Neural networks are a class of models that has, over the past decade, come to revolutionize the field of NLP. Their most basic form, a feedforward neural network (FFN), derives from the perceptron [Rosenblatt, 1958]. It consists of a model that repeatedly performs linear transformations and element-wise non-linear functions on its input. More formally, let **FFN** be an L-layer FFN, with weights and biases $W_1, \ldots, W_L$ and $b_1, \ldots, b_L$, and activation function σ. We define f, run on input $\mathbf{x} \in \mathbb{R}^n$ as follows:

```
$\underline{\mathbf{FFN}(\mathbf{x})}$
$\mathbf{x}_1 \leftarrow \mathbf{x}$
for $\ell$ in $1, 2, \ldots, L-1$ :
    $\mathbf{x}_{\ell+1} \leftarrow \sigma\left(W_\ell^\top \mathbf{x}_\ell + b_\ell\right)$
$\mathbf{x}_{L+1} \leftarrow W_L^\top \mathbf{x}_L + b_L$
Output $\mathbf{x}_{L+1}$
```

For such an approach to work in NLP, we must convert our input (often in text form) into real-valued features. While there are many ways to do this, one possibility is to convert each word of the text into a word vector, as described in the prior section. One early work that does this is Bengio et al. [2003], who train a FFN to predict, given a sequence of n words, the following word. Their model takes in input by converting each word in the length-n sequence into a learned embedding vector, concatenating these vectors, and using this as input to the FFN. The FFN is trained to output a distribution over words in the vocabulary that maximizes the probability of the true next word in the sequence.

More generally, FFNs saw limited use even prior to the widespread popularization of neural networks for NLP [Rumelhart and McClelland, 1986]. However, FFNs have key flaws that affect their ability to process language: they have no way of processing arbitrary-length sequences, such as sentences. Moreover, they have no way to encode positional information about their input. Finally, they scale poorly, as their weight matrices grow with the length of the input sequence. As a result of these flaws, networks with sequential processing abilities were the first to grow in popularity for NLP.

2.2.2 Recurrent Neural Networks

As both computing resources and interest in neural networks for NLP grew, recurrent neural networks (RNNs) [Rumelhart and McClelland, 1987] became a popular architecture in NLP. RNNs maintain an internal hidden state, updating it as they process each input in an arbitrary-length sequence. We can formally define an RNN **RNN**, run on an length-T input $\mathbf{x}^1, \ldots, \mathbf{x}^T$, with an initial hidden state $\mathbf{h}^0$, activation function σ, and weights W_h, W_x, W_y, b_h, b_y as follows:

```
$\mathbf{RNN}(\mathbf{x}^1, \ldots, \mathbf{x}^T)$
for $t$ in $1, 2, \ldots, T$ :
    $\mathbf{h}^t \leftarrow \sigma\left(W_h^\top \mathbf{h}^{t-1} + W_x^\top \mathbf{x}^t + b_h\right)$
    $\mathbf{y}^t \leftarrow W_y^\top \mathbf{h}^t + b_y$
    Output $\mathbf{y}^t$
```

We can see that RNNs output a hidden representation for each word in the sequence. This makes RNNs suitable for tasks such as part of speech tagging [Perez-Ortiz and Forcada, 2001], among other one-to-one sequence-to-sequence tasks. However, another powerful aspect of RNNs is that they form representations of entire sequences. The hidden output corresponding to the final token contains information from all prior tokens, and can serve as a fixed-length representation of the entire sequence. Based on this, RNNs saw use in tasks like language modeling [Mikolov et al., 2010], natural language inference [Wang and Jiang, 2016], dependency parsing [Kiperwasser and Goldberg, 2016].

Moreover, RNNs allowed for new generative approaches such as teacher forcing: newly-predicted tokens could be fed directly into the RNN in order to generate following tokens. Additionally, one RNN could generate a representation of an input sequence, which would then act as the initial hidden state of a second RNN that generated a sequence over a different vocabulary. These new sequence-to-sequence approaches allowed for the use of RNNs in tasks like machine translation [Sutskever et al., 2014] and dialogue generation [Vinyals and Le, 2015].

Despite their versatility, RNNs too have significant flaws. Even LSTMs, RNNs designed to handle long sequences [Hochreiter and Schmidhuber, 1997], forget information over long timespans. Moreover, because of their sequential nature, RNNs are difficult to parallelize compared to other neural network architectures, leading to slower training. However, since they re-use weight matrices across timesteps, they avoid the scaling issues of FFNs; furthermore, their sequential processing neatly matches human left-to-right language processing.

2.2.3 Transformers

The most popular current architecture in NLP, the Transformer [Vaswani et al., 2017], solves many of the RNN's problems. The fundamental mechanism of the Transformer is attention: given a sequence of tokens as input, the transformer creates a new representation for each token by taking a convex combination of other token representation's values. Crucially, it chooses this combination by attending to which other tokens contain useful information for the representation of the token in question. Formally, we can define an attention module, with input $\mathbf{x}^1, \ldots, \mathbf{x}^T$, and weights W_Q, W_K, W_V (of dimensionality d_k) as follows:

```
$\mathbf{attention}(\mathbf{x}^1, \ldots, \mathbf{x}^T)$
$Q \leftarrow \text{stack}\left(W_Q^\top \mathbf{x}^1, \ldots, W_Q^\top \mathbf{x}^T\right)$
$K \leftarrow \text{stack}\left(W_K^\top \mathbf{x}^1, \ldots, W_K^\top \mathbf{x}^T\right)$
$V \leftarrow \text{stack}\left(W_V^\top \mathbf{x}^1, \ldots, W_V^\top \mathbf{x}^T\right)$
$\mathbf{x}'^1, \ldots, \mathbf{x}'^T \leftarrow \text{unstack}\left(\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} V\right)\right)$
Output $\mathbf{x}'^1, \ldots, \mathbf{x}'^T$
```

These attention blocks are normally combined into multi-headed attention blocks, where multiple independent attention heads act on the same input. Their outputs are concatenated and projected into a smaller dimension. So, a transformer block, with input $\mathbf{x}^{1:T} = \mathbf{x}^1, \ldots, \mathbf{x}^T$ can be defined thus:

```
$\textbf{Transformer}(\mathbf{x}^{1:T})$
$\mathbf{a}^{1:T} = \text{Multi-head-attention}(\mathbf{x}^{1:T})$
$\mathbf{y}^{1:T} = \text{LayerNorm}(\mathbf{x}^{1:T} + \mathbf{a}^{1:T})$
$\mathbf{z}^{1:T} = \text{Linear}(\mathbf{y}^{1:T})$
$\mathbf{o}^{1:T} = \text{LayerNorm}(\mathbf{y}^{1:T} + \mathbf{z}^{1:T})$
Output $\mathbf{o}^{1:T}$
```

Note that Linear denotes a learned transformation $\text{Linear}(\mathbf{x}) = W^\top \mathbf{x} + \mathbf{b}$, and LayerNorm is layer normalization [Ba et al., 2016]. Because transformers process all tokens at once, they are easily parallelizable and are well-suited to training on GPUs. This computational efficiency allows many transformers to be stacked on each other, each processing the output of the last, to create very deep networks.

In NLP, such stacks of transformers generally take in words (or tokens) as input: each word is then replaced with a learned word representation. Additionally, positional information is added, as unlike RNNs, transformers do not process words sequentially and thus lack any inherent position information. Given this input, each transformer layer outputs a new representation ($\mathbf{o}^{1:T}$) of the input data, until the final layer is reached.

The transformer approach has reshaped the NLP landscape. Although it initially found success in machine translation, its greatest success has been the creation of large language models, pretrained models that can be adapted to a wide variety of tasks [Devlin et al., 2019, Radford et al., 2019, Raffel et al., 2020]. These models' technical details and usage are detailed in the following section.

From a linguistic standpoint, the way in which transformers generate word representations both aligns with and breaks from how representations should work in humans. For example, the fact that a transformers' word representations rely on words' context is consonant with linguistic reality. After all, the representation of a word should not be static, as with early word embeddings: for example, polysemous words have multiple meanings, and their representations should reflect the meaning that they have in a given context. Other words, such as pronouns, have extremely fluid meanings that depend on the referent of the pronoun in the sentence in which appears.

However, other aspects of the transformer are peculiar. In transformers' attention, any token can attend to and incorporate information from any other token, including later tokens. This is a clear break with human language processing. Even in transformer setups that prevent tokens from attending to future tokens, the fact that processing is parallel, rather than sequential, differs from human processing, unlike RNNs. We will see later how this causes the presence of unexpected information in the internal representations of transformers.

2.3 Large Language Models

Large language models (LLMs), the object of study in this thesis, are currently the dominant modeling paradigm within NLP. Their wide success on diverse metrics of natural language understanding, as well as their ability to generate grammatical and often coherent text has led to diverse opinions on their underlying language faculties. In this section, I give an overview of the most popular LLMs, as well as particular successes and failures that have influenced the discourse surrounding LLMs' language abilities.

To start, let us define large language models. First, they are *large*, at least according to the standards of the day: they contain at least one hundred million [Devlin et al., 2019] but up to hundreds of billions [Chowdhery et al., 2022] of parameters. Second, they are (neural) *language models*: they produce probability distributions over series of words. In practice, this means language models are trained to take in a series of words, and produce a probability distribution over possible following words. Alternatively, LLMs are masked language models, which are fed sentences where some words were randomly replaced by a mask token, and tasked with predicting the masked word at each masked position.

They also share other features, beyond those specified in the name. They are *pretrained* on massive amounts of data, and then fine-tuned on a specific task. They generate *contextual* representations of their input tokens, rather than relying on static word representations. Finally, they tend to make use of *Transformers*. Note that precursors to large language models deviated slightly from this definition: ELMo [Peters et al., 2018] used RNNs, and was intended for use without fine-tuning; CoVE [McCann et al., 2017] pretrained on machine translation.

Most LLMs, however, build on the foundation set by BERT [Devlin et al., 2019]. BERT is a transformer-based masked language model trained on billions of tokens (more architectural details provided in Section 2.4). Upon release, it set new state-of-the-art (SotA) performance on the GLUE dataset [Wang et al., 2018], which measures natural language understanding (NLU) in NLP models. The dataset includes question answering, NLI, linguistic acceptability, and sentiment analysis tasks; thus, it seemed to some to offer evidence of good language understanding by BERT.

Soon after BERT's release, many similar LLMs proliferated, improving on BERT's training procedure [Liu et al., 2019b, Zhang et al., 2019, Lewis et al., 2020], shrinking BERT [Sanh et al., 2019, Lan et al., 2020], and adapting BERT for use with individual non-English languages [Martin et al., 2020, Sido et al., 2021]. Many of these models also set SotA performances on challenging datasets, such as GLUE's more difficult successor, SuperGLUE [Wang et al., 2019].

At the same time, LLMs based on traditional, rather than masked, language modeling, also became popular, starting with GPT-2 [Radford et al., 2019]. Being trained as language models, these models were also generative models, able to produce strings of text in response to input. Moreover, these models' sizes quickly increased by orders of magnitude [Raffel et al., 2020, Brown et al., 2020, Chowdhery et al., 2022]. Curiously, at large sizes, these LLMs showed a remarkable ability to perform new tasks with few to even zero examples provided, and no training, via a prompting methodology. Simply by providing LLMs with labeled examples of a task as input, the LLM could provide a label to a new example with surprising accuracy. More work quickly arose finding new, better ways to prompt LLMs and perform tasks in a few-shot fashion. This, too led to excitement around the linguistic abilities of LLMs.

In sum, LLMs represented a new paradigm in NLP that significantly improved

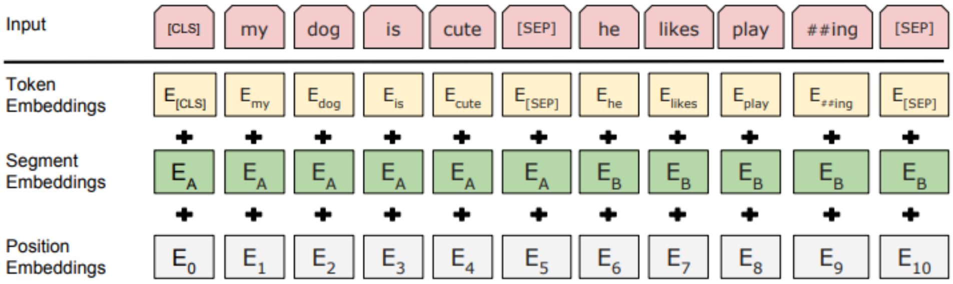

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
$E_{[CLS]}$ E_{my} E_{dog} E_{is} E_{cute} $E_{[SEP]}$ E_{he} E_{likes} E_{play} $E_{##ing}$ $E_{[SEP]}$
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B E_B
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Figure 2.1: BERT's embeddings (from Devlin et al. [2019])

model performance across a wide range of NLU and NLP tasks. This rapidly changed the NLP landscape, and generated significant "hype" regarding LLMs, and their ability to process language. However, coarse-grained metrics such as accuracy on NLU tasks are not sufficient to measure LLMs' language abilities; indeed a wide body of work [Rogers et al., 2020] has emerged to analyze this very question, covered in the following section.

2.4 A Close Look at BERT's Architecture

In order to understand the issues that arise when analyzing LLMs with the methods detailed in the next section, we will need a detailed understanding of their functioning. In this section, we use BERT as a case study, explaining in detail its architecture and pre-training, walking through an example as BERT processes it.

Fundamentally, BERT is a masked language model, pre-trained on the English Wikipedia and OpenBooks [Zhu et al., 2015] corpora. Its pretraining process works as follows. First, two sentences are selected from the dataset; the second either directly follows the first, or is a random other sentence, with equal probability. Then, the two sentences are tokenized with the WordPiece tokenizer [Wu et al., 2016], which prepends a [CLS] token to the input and separates sentences by appending a [SEP] token. It also splits words into tokens which are often smaller then the original word; this allows BERT to learn morphological information sub-word associated with these sub-word tokens. Then, at training time, 15% of token indices are randomly selected to be masked. Of these, 80% are replaced with a special [MASK] token, 10% with a random token, and 10% are left unchanged; the model's objective will be to predict the original tokens at these indices.

Having created the input, we now see how BERT processes it. First, BERT converts the tokenized input into embeddings (Figure 2.1). Each token is represented by the sum of a position embedding, which indicates the token's index, a segment embedding, which indicates which sentence the token is in, and a learned token embedding, like the word embeddings discussed earlier. Next, the tokens are processed by a series of transformer layers—in BERT-base, there are 12 transformer layers with 12 heads each and a hidden size of 768, while in BERT-large, there are 24 layers, each with 16 heads and a hidden size of 1024. At each layer and token position, BERT creates a new token representation, combining information from all token representations from the prior layer.

Finally, in its last layer, BERT uses a linear layer to project each token representation into logits for each word in the vocabulary; applying softmax yields probabilities for each word. During pre-training, BERT is trained (using cross-entropy loss) to predict the correct original token at each of the masked positions in the input. It is also trained to predict whether the second sentence in the input originally followed the first sentence; this is called "next sentence prediction". In fine-tuning, this final layer is removed and replaced with a task-specific layer.

2.5 Evaluating the Language Abilities of Large Language Models

Given the immense fanfare regarding LLMs performance on NLP tasks: how do we evaluate the degree to which LLMs capture linguistic competence? The phrasing of this question is difficult and broad, because the task of defining linguistic competence in LLMs is complex. Broadly, past work has sought out such competence in two ways. The first category of work examines LLMs' internal representations of words, sentences, and other linguistic units, for evidence that they represent linguistic structures or phenomena in ways that reflect our knowledge of them. The second category of work examines LLMs' external behavior in response to linguistic inputs, and compares it to expected or human behavior on those inputs. In this section, I provide an overview of both lines of work, with the aim of unifying them in following sections.

2.5.1 Evaluation of Internal Representations

One method for evaluating LLMs' capture of language competence is the analysis of internal representations. All of these representations can be analyzed, for linguistic information.

One of the predominant methods for the analysis of LLMs' internal representations is probing (see Belinkov [2022] for a longer overview). Although this term occasionally refers to certain techniques used to analyze model behavior, we will use it to refer to the use of auxiliary models to determine the information content of LLMs' internal representations. Specifically, in a probing regime, we have representations of some linguistic unit (say, words), which we believe should encode some attribute (say, the words' part of speech). We then train a classifier to predict the attribute from the representation, e.g. we train a classifier to predict a word's part of speech, from its representation.

Formally, let $\mathbf{X}$ be a set of instances of some linguistic unit $\mathcal{X}$, i.e. a set of words, phrases, sentences. Let $f : \mathcal{X} \to \mathcal{Y}$ be a function mapping from that linguistic unit to a set $\mathcal{Y}$ of attributes. Then, let $\phi : \mathcal{X} \to \mathbb{R}^n$ be a function producing real-valued representations of that linguistic unit. In probing, we train a classifier $h : \mathbb{R}^n \to \mathcal{Y}$ to predict a word's attribute from its representation.

Concretely, we can imagine that $\mathbf{X}$ is a set of words (in context), and f maps from a word to its part of speech (POS). Then, let ϕ be the word's representation from a neural network, and h be a linear classifier. Then, if h successfully learns to map from neural network representations to their POS even on unseen data, we might conclude that the network encodes POS in its representations.

Such probing techniques were used prior to the advent of LLMs to probe RNN-based machine translations models [Shi et al., 2016], [Adi et al., 2016], and sentence encoders Conneau et al. [2018] for qualities ranging from tense and subject / object number to sentence length and parse tree depth. Note that there has a significant diversity of probes, even relatively recently: while some authors use only linear probes [Bisazza and Tump, 2018, Liu et al., 2019a], others use FFNs with one or two hidden layers [Belinkov et al., 2017, Ettinger et al., 2018].

Naturally, probing has also become part of the interpretability arsenal levied against BERT-family models. Probing work has found evidence for dependency and constituency structure [Hewitt and Manning, 2019, Rosa and Mareček, 2019], as well as named entities [Liu et al., 2019a], encoded in BERT's representations. Tenney et al. [2019b,a] perform a pair of large studies which find evidence that BERT representations encode information about a myriad of properties, from POS to semantic role, at various levels in the model. Ultimately, they conclude that BERT rediscovers parts of the classical NLP pipeline—a significant finding for a model known to be opaque.

However, probing has also received a significant amount of criticism. One line of criticism focuses on whether probing actually achieves its goal of extracting information from representations. Hewitt and Liang [2019] investigate whether powerful probes, such as FFNs with one or two hidden layers, learn to memorize labels, rather than simply extracting linguistic information encoded in the representations. This is undesirable: if a probe capable of memorizing labels succeeds on a probing task, we cannot determine if the probe extracted a meaningful attribute encoded in the input representations, or if the probe just learned to map from representations to labels.

Hewitt and Liang [2019] propose testing for memorization via control tasks, which mirror the underlying task (e.g. POS tagging or dependency parsing) but have random / arbitrary labels. If a probe can learn to map from words' representations to these arbitrary labels, the probe is likely too strong; the probe can learn arbitrary mappings without relying on information from the input representations, which surely encodes nothing relevant to the label, except word identity. Using this test, they find that more powerful networks like FFNs exhibit memorization, especially when the task is simple. However, more complex tasks require the use of more powerful probes, lest the probe be unable to extract information that is encoded there. Later work [Voita and Titov, 2020, Pimentel et al., 2020] attempts to quantify model complexity an incorporate this into probe evaluation, in order to offset this problem.

Some methods avoid this criticism by avoiding probes that involve learning extra parameters. Wu et al. [2020] use a parameter-free probe to recover syntactic trees from BERT, while Wiedemann et al. [2019] show that BERT representations naturally embed polysemous words into clusters. Other methods eschew the analysis of individual representations entirely, turning to other components of BERT, such as its attention heads. Such studies have found that some heads attend to dependencies and coreferring expressions, and contribute to subject-verb agreement processing [Clark et al., 2019, Htut et al., 2019, Lin et al., 2019].

Unfortunately, one major flaw plagues many of these methods, probing especially. Although internal representation analysis has shown promising results about language processing in LLMs, one thing is still unclear: do LLMs actually use the linguistic information encoded in their representations? That is, does the fact that a word's representation encodes POS such that it is extractable via a probe, mean that LLMs rely on POS to make their predictions? Or is such information tangential to LLM processing; does it go encoded but unused?

It may well be the case that we can find linguistic information encoded in deep neural network, but it is not used. Ravichander et al. [2021] show that models trained on natural language inference learn to encode features such as tense or subject number, even when these features are not task-relevant (and thus not used). This calls into question the value of purely internal analyses. While it is significant and interesting that linguistic information is encoded by LLMs, if this encoded information does not affect LLM behavior, is it of interest to either linguists or practitioners? The question of why LLMs learn this information without using it might be interesting, but actually probing for this information would not be a fruitful path to understanding language processing in LLMs.

2.5.2 Behavioral Analyses

Behavioral analyses provide an excellent complement to internal representation analysis. We refer to as behavioral analysis of LLMs any technique that analyzes what a LLM has learned based on its outputs on specific inputs. This definition is very broad, and includes even typical fine-tuning techniques used with LLMs. Indeed, fine-tuning LLMs and evaluating them on benchmarking datasets is a form of behavioral analysis; it is simply a coarse-grained evaluation.

However, such coarse-grained evaluations can yield misleading conclusions. While initial testing of BERT on benchmarking datasets suggested that it had defeated challenging datasets, datasets are frequently found to have flaws [Poliak et al., 2018, Gururangan et al., 2018], and new datasets released that are challenging for machines, but not for humans. Behavioral analysis can help us determine whether LLMs are solving these tasks or relying on heuristics or dataset artifacts. For example, McCoy et al. [2019] find that fine-tuned BERT relies heavily on lexical overlap between premise and hypothesis to solve NLI.

Still, behavioral analysis on fine-tuned models can only tell us what fine-tuned models have learned; perhaps the underlying LLMs do not share the same flaws as their fine-tuned counterparts. Fortunately, bare LLMs have also been the object of much study. In general, such studies use the fact that LLMs are language models, or masked language models, to extract behavioral information from LLMs. In the case of standard language models, studies provide the LM with a prompt and must continue the given phrase; in the case of masked language models, models receive prompts containing [MASK] tokens and must fill in the blank. Their performance is assessed based on their response to these tasks.

For example, Ettinger [2020] uses this methodology to test BERT on a variety of psycholinguistic suites originally used for human analysis, finding that BERT differs from humans in its processing of negation and semantic role. Bacon and Regier [2019] use prompting to show that BERT's agreement abilities grow worse with increasing distance between agreeing words; Pandia and Ettinger [2021] find that distractors also inhibit these abilities. Warstadt et al. [2019] use prompting in a test suite demonstrating BERT's ability to process negative polarity items.

Some behavioral studies go so far as to re-pretrain LLMs to understand their learning and behavior. Liu et al. [2021] re-train RoBERTa, testing it intermittently for linguistic generalization. They find that linguistic knowledge is acquired early in training, in contrast to commonsense or reasoning faculties, which are learned later or not at all. Wei et al. [2021] examine LLMs' ability to conjugate verbs of varying frequencies by altering verb frequencies in the training dataset and re-training BERT. They find that BERT is often able to correctly conjugate subject-verb pairs never seen in training; however, BERT still display a bias towards frequent verb forms. Sinha et al. [2021] re-train BERT on English sentences whose words are randomly ordered. Surprisingly, BERT's performance remains high when tested on tasks with normally ordered input, suggesting that BERT is insensitive to word order.

These analyses all share a common thread: they are all informative with respect to how the LLM analyzed behaves. However, none of these analyses can tell us how LLMs are actually performing processing. Although some work is being done on the formal properties of transformer-based models [Merrill et al., 2021], without strong formal constraints on the underlying programs computed thereby, behavioral analyses cannot tell us how LLMs process language. External behavior can correspond to many different internal programs causing that behavior; thus we cannot distinguish through behavior alone how processing occurs.

This is particularly problematic for positive results from behavioral studies. Even if we observe that a model exhibits the correct behavior on a linguistic task, we still cannot conclude for sure that the model is actually processing language using the same underlying rules as humans. Although the model might perform correctly in a given scenario, the underlying program encoded within the models weights could still be distinct.

This underlying flaw is difficult to compensate for while remaining within the realm of behavioral analyses. While one can perform ever-more behavioral analyses to ensure that the model performs correctly in a wide variety of scenarios, there could always be another dataset, another task, that reveals incorrect behavior in our model. If we are to gain confidence in our models' performance, we must know what is happening inside them. That is, we must combine internal representation analysis with external behavior analysis.

3. Unifying Internal and External Analyses of LLMs

If, as claimed in the prior section, both internal and external analyses are necessary to understand the functioning of LLMs, how can these two modes of analysis be combined? In this section, I lay out the foundation of a framework that unifies these two modes of analysis into one. The fundamental principle is very simple: first, we observe the model's behavior on a task of interest. Then, we run the model again, on the same data, but we perform an intervention on its internal representations, i.e. we change the flow of information through the model. Finally, we observe its behavior again.

This sort of analysis allows us to extract causal information from these models. Rather than just offering post-hoc analyses of what could have caused model behaviors, we can test hypotheses about how our models translate from input to output. If we believe our model internally operates in a certain fashion, we can perform an intervention on the model's internals; if the model's behavior changes as predicted by our hypothesis, we gain evidence for it. In this section we provide an overview of such intervention techniques, and explain the techniques we use in this thesis.

3.1 Probing Interventions

Probing interventions are an internal and external analysis combining probe with behavioral analyses. Probing interventions aim to go beyond basic probing analyses: instead of merely proving that a given type of information is encoded in a model's representations, they prove also that this information is used.

Probing interventions do this via a series of steps. First, they evaluate a model on a behavioral task of interest. Second, they train probes to extract information from a model's representations as in standard probing analyses. Third, they run the model once more on the first task. However, they remove or alter the information that was probed for from the model's representations; thus, if that information was important to the model's performance of the task, the model's task performance should change.

This third step is the piece that allows us to extract causal information. Imagine that we hypothesize that a certain type of linguistic information is both encoded and used by a model. We can then choose a behavioral task to which the probed-for information is key, assuming our hypothesis is correct. When we run our model on this task, but remove or alter the information of interest, we should see very significant and predictable affects on model performance. That is, this intervention should cause a change in model behavior, but only if the linguistic information probed for is both encoded and used.

How, then, can we perform such interventions? Returning to the formalism from Section 2.5.1, let $\mathcal{X}$ be all possible instances of a given linguistic unit (e.g. words, phrases, etc.), $\mathcal{Y}$ be all possible values of some linguistic property, e.g. all the different parts of speech. $f : \mathcal{X} \to \mathcal{Y}$ is a function assigning to each instance of a given unit the value of the property of interest. Then, we have a dataset $\mathbf{X} \subset \mathcal{X}$, and representation function $\phi : \mathcal{X} \to \mathbb{R}^n$. In this intervention scenario, ϕ yields the representations given by running a LLM H on the input $\mathbf{x}$. Then, we learn a classifier $h : \mathbb{R}^n \to \mathcal{Y}$ that predicts a unit's property given its representation, on examples $(\phi(x), f(x))$ drawn from $\mathbf{X}$. Given this formalism as a jumping-off point, we can arrive at a variety of different probing interventions.

Amnesic Probing: A Projection Intervention One approach to probing interventions is to remove entirely any information regarding the attribute that the probe seeks to predict from the LLM's representations. That is, given a probe that predicts POS, one can remove POS information. Once that information is removed, one can observe the LLM's behavior to see if it relied on that information, as found by the probe.

This can be done as follows. Let our probe h be a binary linear classifier parameterized as $h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$. Imagine that we are running H on an input $\mathbf{x}$ that contains a linguistic unit whose representation we wish to train. Then consider the stage of the computation where we have just computed the representations of layer ℓ, including $\phi(\mathbf{x})$. We then replace the representations originally corresponding to $\phi(\mathbf{x})$, with $\text{proj}(\phi(\mathbf{x}), \mathbf{w})$, and continue computation. Note that here, $\text{proj}(\phi(\mathbf{x}), \mathbf{w})$ indicates the projection of $\phi(\mathbf{x})$ onto the plane whose normal vector is $\mathbf{w}$.

They key operation here is the projection, which removes information about the probed-for characteristic from the model's interpretation. This projection works because h is a binary linear classifier; thus its weights define a plane bisecting $\mathbb{R}^n$. Points on one side of the plane are assigned one class, and points on the other side, the other class. Projecting the $\phi(\mathbf{x})$ onto this plane means that, as determined by h, there is no information in the projected point about its class; h should assign equal probability to each class.

Thus, if H is using the linguistic information as probed for by h, this projection technique will remove this information that H relies on, changing its behavior. However, since we are performing a projection, we will in theory preserve all information not relevant to the probing task. See Figure 3.1 for an example of this technique using a probe trained on plurality.

Elazar et al. [2021] introduce amnesic probing, a slightly more advanced version of the aforeproposed technique. Instead of just projecting the representations onto the decision boundary of one probe, they use another technique, iterative nullspace projection (INLP) [Ravfogel et al., 2020]. In this technique, probes are trained to detect a certain attribute in a dataset of representations; then, the representations are projected onto their nullspace as described above, such that (with respect to the probes), no information about that attribute remains in the representations. This process is repeated to ensure that no linearly extractable information regarding that attribute remains in the dataset. Using amnesic probing of POS and dependency labels, Elazar et al. [2021] show, as discussed previously, that diagnostic probing task performance is not necessarily related to task performance.

AlterRep: A Reflection Intervention Other geometric techniques beyond projection are also possible. Ravfogel et al. [2021] go beyond amnesic probing to propose, AlterRep, which uses other geometric operations and works as follows.

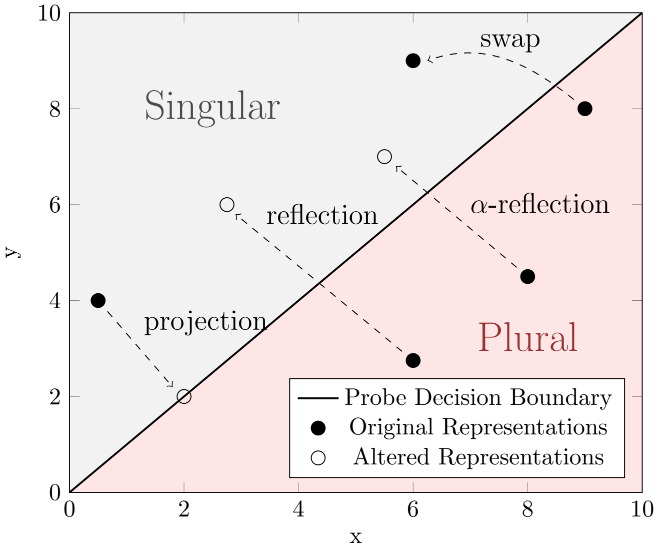


Figure 3.1: Hypothetical probing interventions in $\mathbb{R}^2$, where the probe was trained to classify nouns as singular vs. plural. In **projection**, all plurality information is removed. In **reflection**, the representation's plurality is flipped. In α**-reflection**, the representation's plurality is just barely pushed over the decision boundary. Finally, in **swap**, the representation is replaced with a corresponding representation on the other side of the decision boundary.

Denote the component of $\phi(\mathbf{x})$ orthogonal to $\mathbf{w}$ as $\phi(\mathbf{x})_{\mathbf{w},\perp}$. Instead of replacing $\phi(\mathbf{x})$ with $\text{proj}(\phi(\mathbf{x}), \mathbf{w}) = \phi(\mathbf{x}) - \phi(\mathbf{x})_{\mathbf{w},\perp}$, AlterRep instead replaces $\phi(\mathbf{x})$ with its reflection across $\mathbf{w}$'s plane, $\phi(\mathbf{x}) - 2\phi(\mathbf{x})_{\mathbf{w},\perp}$. This moves the representation across the decision boundary, changing the class assigned to it, and ideally changing the behavior of H accordingly. We can also replace $\phi(\mathbf{x})$ with $\phi(\mathbf{x}) - (1+\alpha)\phi(\mathbf{x})_{\mathbf{w},\perp}$ for some small α. This still changes the label assigned, but changes the representation less than reflection ($\alpha = 1$) does. See Figure 3.1 for an image of this process.

Overall, we see that geometric interventions are quite powerful, but come with constraints. There is a wide array of linear transformations with respect to the linear probe that can be applied to a model's internal representations in order to change the information contained within. However, this technique works most smoothly if the information being probed for is encoded linearly, and as a binary task. Still, this technique is one of few that allows us to make controlled changes to model internals and observe their results.

Ravfogel et al. [2021] apply AlterRep to the question of relative clauses. Specifically, the quality probed for (and intervened upon) is "is this word in a relative clause?". Having trained probes on this task, the authors then test a model on a dataset of the form "The skater that the officers love [MASK] happy", i.e. sentences that contain both a subject of a given number, and an object in the relative clause, with a differing number.

Generally, despite the proximity of the object in the relative clause ("officers") to the verb, BERT and other such LLMs can correctly predict that the verb should be conjugated according to the number of the subject [Goldberg, 2019]. However, here, the authors intervene upon the "in the relative clause" quality of the word "officers". They predict that by doing so, the model will predict the masked verb as being conjugated for “officers”, rather than “skater”. Indeed, the results of the experiment indicate that this intervention has mild effects, primarily when the intervention is performed upon the middle layers of the model.

Gradient-based Interventions While we focus on these geometric methods, similar techniques have also been proposed. Giulianelli et al. [2018] propose a gradient-based method for probing interventions. In this method, the goal is still to change H's output by changing its internal representations with probes. However, instead of changing the representation to another class by performing geometric operations on it, we instead choose a class that we want the representation to belong to.

For example, given a sentence like “The boy [MASK] the teacher,” we might wish to change the representation of the token $\mathbf{x} =$“boy” to belong to the class $y = 1$, “plural”. We do so by computing $\nabla_{\mathbf{x}} = \frac{d(-\ln(h(\mathbf{x})))}{d\mathbf{x}}$, and then setting $\mathbf{x} = \mathbf{x} - \alpha \nabla_{\mathbf{x}}$ for some positive α. That is, we compute the gradient of the probability assigned by the probe to class y with respect to the representation, and update the representation to make y more likely. We note that this method is very similar to the preceding ones, assuming we are using a binary classifier to act on a point that is not of class y. While the magnitude of the gradient will differ based on the point chosen, the direction will be identical to $\phi(\mathbf{x})_{\mathbf{w},\perp}$. Giulianelli et al. [2018] use this method in the way opposite to the previous proposals: to correct errors made by LSTMs in long-term number agreement.

3.2 Other Causal Interventions

Interchange Interventions Geiger et al. [2020] and Geiger et al. [2021] introduce swap, or “interchange” interventions as a method of probing models' computational structure and behavior. This type of intervention is much simpler than probing interventions, and works as follows. First, we specify a hypothesis about how our model of interest performs its task. For example, we might have a LLM that performs sentiment analysis, and we might hypothesize that it is sensitive to negation at lower layers.

Then, given an example of a text-based task with a label, we first create a counterfactual example by slightly altering the text, such that the label changes. For example, we might have “The movie was very good” (positive) and “The movie was not good” (negative). We run the model on the counterfactual example, storing its internal representations. Then, we run the model on the original example; however, we swap relevant tokens' representations with those from the counterfactual example. In this case, we would replace the representation of “very” with that of “not” at a given layer, and expect the model to predict “negative”, but only at lower layers, according to our hypothesis. We then observe the model's true predictions and draw conclusions about its behavior. To visualize examples based on plurality, see Figures 3.1 and 3.2.

How does this work concretely? Geiger et al. [2020] study NLI, and hypothesize that the model stores the specific entailment relation between premise and hypothesis in the [SEP] token that separates them in the input. Thus, by replacing the [SEP] token representation of one NLI example, with the [SEP] token representation of another example containing a different entailment relation,

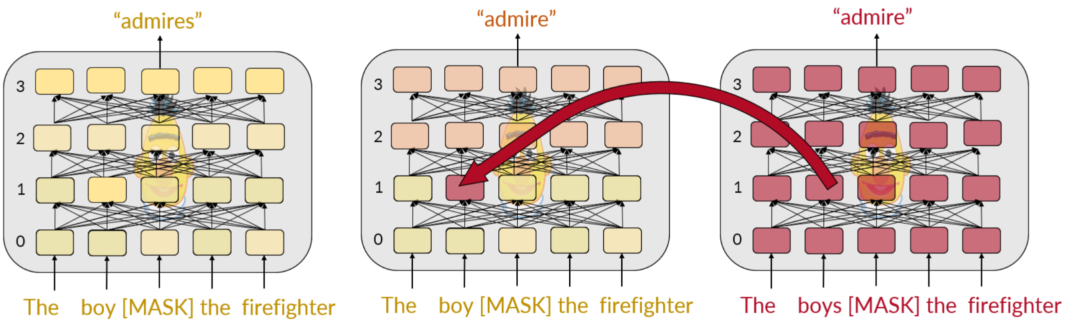

"admires"
"admire"
"admire"
3
2
1
0
The boy [MASK] the firefighter
The boy [MASK] the firefighter
The boys [MASK] the firefighter

Figure 3.2: An example of swapping. In the unaltered case, when "boy" is singular, BERT predicts "admires" as the masked token; when the subject is plural "boys", BERT correctly predicts "admire". In the center, we see the swap scenario: the layer-1 representation of "boy" is replaced with that of "boys" in the same context, from the same layer. This changes BERT's output to "admire," indicating that the plurality of the representation of "boy" causally relates to the conjugation of the output verb.

the model's output should be changed. They find that in some subsets of their dataset, this hypothesis does hold; however, it does not work for all examples.

Geiger et al. [2021] expand this method to study NLI using a slightly more complex dataset and hypothesized causal model. Their hypothesized model is the same model used to artificially generate the dataset, which consists of simple sentences exhibiting phenomena like quantification and negation. They attempt to find alignments between their hypothesized model and BERT, their model of study. They do so by performing swap interventions on representations in BERT they believe correspond to nodes in the computational graph specified by their hypothesized model. They find that BERT's computation of object noun phrases aligns with the predictions of their model.

Causal Tracing Interventions Similar to the prior method, Meng et al. [2022] introduce causal tracing interventions as a way to track where models store factual information. They propose to locate where models (they investigate GPT-2) store factual information via a simple intervention that works as follows. First, run the model on a factual query, and store its activations. Next, run the model again on the same query, corrupting some of the input embeddings; this will change the output to the query. Then working with the same corrupted input, "restore" some of the model activations by replacing them with activations generated by the uncorrupted query. At some point, this will restore the original query's output. The minimal set of neurons whose activations must be restored in order to recover the original output is the set of neurons in which the relevant information is stored.

This method is nearly identical to the preceding one, with the difference that instead of working with two different examples, we examine only one example and its corrupted counterpart. This method also allow for the examination of the activations of multiple neurons, although the authors ultimately focus on one specific midlayer neuron per input for the purpose of editing factual information.

3.3 Our Methodology

As reviewed in the prior two sections, methodologies combining internal and behavioral analyses are rather new and have seen only limited applications. In this paper, we show how applying multiple probing interventions, in a new but familiar domain, yields us new insights into both the effectiveness of probing techniques and LLMs' language processing. Concretely, we do this by applying both probing interventions and other causal interventions to the same question: subject-verb agreement in LLMs.

Applying both of these techniques is useful: while probing interventions can be shown to have *an* effect, the magnitude of their effect is hard to evaluate without context. Consider the case of a probe trained to predict noun plurality, which we use in a reflection intervention to change the plurality of a verb's subject. We measure if the predicted verb agrees or disagrees with the original subject's plurality. If a probing intervention increases model disagreement to 25%, does this indicate that our probe has learned a functionally-relevant feature? Or that our probe did not learn one, as error did not rise above 50%?

By also applying a swap intervention, where we replace the subject's representation with a representation with the opposite plurality, we can provide an upper bound on probe performance. That is, given a sentence like "The dog [MASK] the turkey.", we can compare the effects of applying a reflection on the representation of "dog", and replacing the representation of "dog" with that of "dogs", at various layers. The closer in magnitude the effects of the former technique to those of the second, the better.

This unification of causal intervention methodologies is one that was not possible in the domains explored by other papers. For example, Ravfogel et al. [2021] apply probing interventions to change whether BERT believes a noun is in a clause, but that domain of intervention isn't well suited for swap intervention. This is because there is no sentence neatly corresponding to the counterfactual scenario in which all other words in the sentence is the same, but the noun is not in the clause. Our specific choice of domain, in contrast, allows us to use multiple methodologies for clearer insights.

4. Investigating Plurality Information in LLM Representations

Having introduced a framework for using causal methods to determine whether LLMs truly use the information that probes find in them, we must now return to our question: to what degree does BERT's use of the information encoded in its representations correspond to linguistic intuition? To answer this question, we choose a type of information possibly encoded in BERT's representations, and investigate it using the aforementioned methodology.

The phenomenon that the rest of this thesis will focus on is *number*, specifically on the plurality of nouns when they act as the subject of verbs. This is an appealing attribute of study because it is binary; thus we can easily probe for it and apply the probing interventions discussed earlier. Moreover, the plurality of nouns, when they act as the subject of present-tense verbs, directly triggers agreement effects. Specifically, the 3rd-person singular conjugation of most verbs in the present tense ends in "-s", distinct from all the other persons. This means that by altering the number of a masked present-tense verb's subject, we can also change the form of the verb that should be predicted for the mask token.

There do exist other phenomena in English that would be appropriate targets for study with this methodology. For example, one could probe for whether a noun takes the indefinite article "a" or "an"; then, masking the indefinite article, use probing interventions on the noun, to see if BERT's predicted article is altered. One could also probe for verb tense, and then try to produce an effect using that.

Regrettably, eliciting effects beyond agreement in a way that meets the requirements of this method (a binary phenomenon, whose effects can be seen using just one mask token) is challenging. Many interesting non-agreement phenomena have more than 2 classes, or have sentence-level effects that cannot be captured in one mask token. We note, however, that in languages with richer morphology than English, that have a greater variety and complexity of agreement phenomena (gender, vowel harmony, etc.), which could be probed using this methodology. Due to time constraints, we leave this for future work.

Having established both the methodology and phenomenon of interest, we now proceed to the main purpose of this thesis: investigating LLMs' processing of subject-verb agreement, and examining the distinct perspectives yielded by different interpretability methods. We do this using the following steps, which follow the evolution of probing through the various methodologies aforediscussed.

1. Probing: We first probe the representations of nouns that are the subject of a verb, attempting to extract the noun's plurality (singular / plural).

2. Probing Intervention: Next, we use a variety of probing interventions to demonstrate that LLMs do use the plurality information contained within subjects' representations in order to make predictions.

3. Swapping Intervention: Then, we compare the effects of probing interventions to those of swapping interventions, which we posit as an upper bound

Dataset	Sentence
Brown	It urged that the city take steps to remedy this problem.
Diagnostic	The exile taught the martyrs.

Table 4.1: Example sentences from datasets used in the initial probing experiment.

on the effects of any such intervention.

4. Diffused Information Probing: In order to ensure the that BERT is only using plurality information encoded where we expect it, we check if BERT is also encoding information about noun subjects' plurality in other words besides the noun subject.

5. Diffused Information Probing Intervention: We again use probing and swapping interventions to demonstrate that although plurality information is distributed across sentences, LLMs only use the information contained within relevant parts of the sentence.

6. Confirming Hypotheses with a More Complex Datasets: We show that the above findings also generalize to slightly more complex datasets than the very simple synthetic dataset used in the initial studies.

4.1 Probing for Plurality

In this experiment, we use simple, standard probing techniques on two datasets and 6 models to show that nominal plurality information can indeed be found in LLMs' representations.

Dataset We run this experiment on two datasets. The first is a subset of the Brown corpus, which consists of well-annotated sentences; each word is annotated with part of speech, including plurality information for nouns. To train the probes on the dataset, we first iterate over each sentence in a random subset, and generate representations of the sentence using each model of interest. For each subject noun in the sentence, we save its representation, along with its plurality as a label. If the noun is split into multiple tokens by the tokenizer, we generate multiple (token, label) pairs, considering each a different example. We do this until we have 8000 examples.

The second dataset is a synthetic diagnostic dataset from Klafka and Ettinger [2020]. It consists of examples that follow a simple template: "The [SUBJ] [VERB-PAST] the [OBJ]."; each example is labeled with the plurality of the subject. The subject and object of the sentence are either singular or plural nouns; all nouns are regular and describe animate humans, such as professions. The verb, always in the past tense, is a transitive verb that is grammatically correct but often unusual or uncommon given its arguments. Compared to the Brown dataset, this dataset is artificial and very simple; however, this simple format has significant benefits.

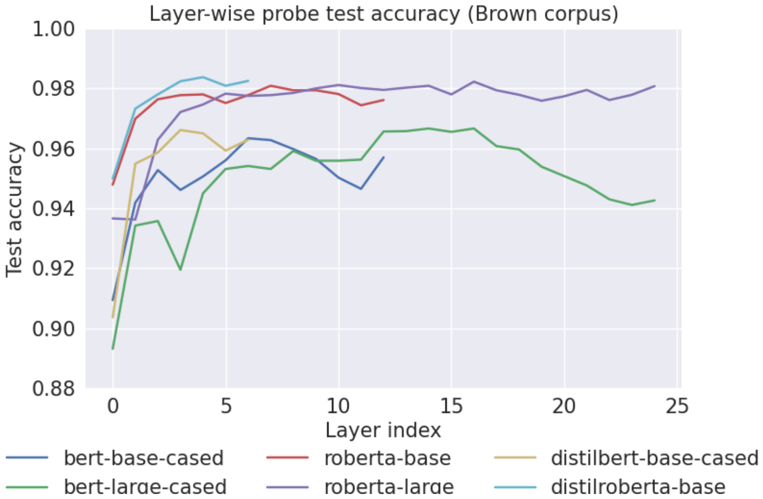


Figure 4.1: Test accuracy by layer for probes trained to predict plurality from subject representations from the Brown corpus.

The first dataset has 8000 examples (train: 6000 / valid: 1000 / test: 1000), 5874 singular and 2126 plural. Thus, this dataset is rather imbalanced, but we will see that the model learns both classes well at all layers of LLM representations in spite of this. The second dataset has 6000 examples (train: 4000 / test: 1000), 2505 singular and 2495 plural. We split out 500 training examples to use as validation examples, since none were provided as part of the dataset. Examples from both can be found in Table 4.1.

Experiment The representations discussed in the prior section were generated by the following models: `bert-base-cased`, `bert-large-cased` [Devlin et al., 2019], `distilbert-base-cased` [Sanh et al., 2019], `roberta-base`, `roberta-large`, `distilroberta-base` [Liu et al., 2019b]. For all models, we use the Huggingface `transformers` [Wolf et al., 2020] implementations and checkpoints. Although these model choices are not especially relevant to this section, in following sections, we will observe different trends for each model.

We train probes for each layer of each model, leading to a total of 80 probes trained per dataset. Each probe is a Linear layer with bias (an affine transformation) which inputs one output with a sigmoid activation function; in essence, we train logistic regressors. We train these regressors using batched gradient descent. While we recognize the potential for more exact or efficient parameter estimation using e.g. maximum likelihood estimation, the results demonstrate that the probes have effectively learned to perform the task. All experiments are implemented in Python using PyTorch [Paszke et al., 2019], PyTorch Lightning [Falcon and The PyTorch Lightning Team, 2019], and HuggingFace `datasets` [Lhoest et al., 2021]. Further details are available in Appendix A.1.

Results & Discussion Figure 4.1 and Figure 4.2 report the test accuracy of probes on the plurality prediction task, by model and by layer, for the Brown and diagnostic datasets respectively. Note that a direct layer-by-layer model

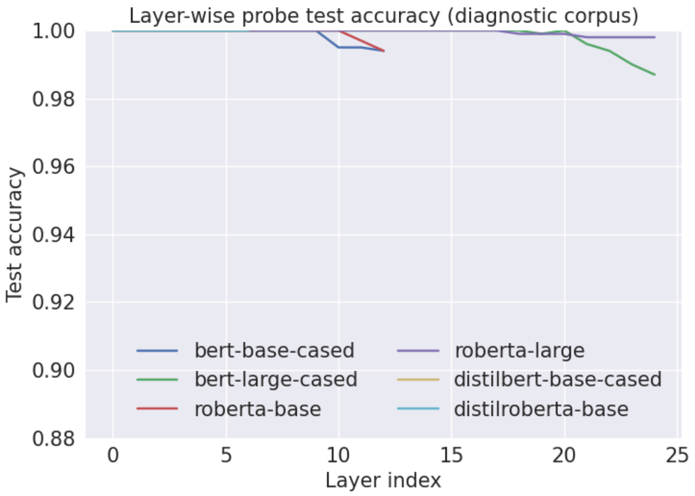


Figure 4.2: Test accuracy by layer for probes trained to predict plurality from subject representation from the diagnostic corpus.

comparison is not possible: models have different depths, so comparing a specific layer number across models will not make sense. Rather, it is more informative to look at e.g. the early or late layers of each model.

Test accuracies are consistently strong across models and layers. Some trends do emerge: in the Brown corpus, most models have slightly weaker performance in the first (embedding) layer, and their accuracy experiences a sharp jump in the second layer. Performance peaks in the middle layers, and later dips slightly.

Overall, performance is much higher on the diagnostic dataset, above 98% in all cases. This is not surprising, because this dataset is much less diverse: its subjects are uniformly human nouns, participating in very simple sentences. This lack of diversity likely makes this task easier to learn. It is still the case, however, that performance dips slightly in later layers.

This high performance indicates that probes were able to learn to extract plurality information from nouns' representations. A traditional account of probing might stop here, taking these results to indicate that the models encode plurality in nouns' representations. But do they actually use the information as detected by the probes? Or are the probes detecting or memorizing functionally irrelevant information? In the next section we target the question of functional relevance.

4.2 Targeted Probing Intervention

In this experiment, we use probing interventions to intervene on models' representations of plural nouns acting as verb subjects, in order to test that the information encoded in these representations is actually used. This requires a very specific setup: we must have a verb that should be conjugated in the present tense; ideally, the subject should be only one token in length as well. Moreover, we must be aware if there exist other traces of the noun's plurality within the verb phrase. Consider the case that the determiner reflects the noun's plurality (e.g. “this” or “these”), or the noun is part of a coordinated noun phrase connected by "and". This will affect the outcome of our intervention: other clues besides the subject noun's plurality will indicate to the model the true plurality of the subject, and thus the correct conjugation of the verb.

It is difficult to ensure that these conditions are fulfilled with natural data; the listed cases that might make an example unsuitable are just a subset of those too numerous to implement in a rule-based fashion. Unfortunately this means that we must focus initially on the synthetic second dataset; we expand later, however, to more complex synthetic datasets.

Dataset Small changes are needed to adapt the diagnostic dataset into a form compatible with these probing interventions. First, we mask out the verb in the sentence. When masking out the verb, we always replace the verb with one mask token. This is for two reasons: first, if there are multiple mask tokens, evaluating model performance is much more difficult. Given n mask tokens, MLMs emit n probability distributions over tokens in the vocabulary. These cannot be easily converted into a distribution over words that span n tokens. Second, it is not important in our setup that the model be able to predict the original verb that was masked. As described in the methodology section, we observe the LLMs' predictions of the masked token to determine the effectiveness of our intervention. There are many present tense verbs that fit into one masked token; observing the probability mass assigned to each conjugation (ending either in "-s" or not.) suffices to characterize LLM behavior on a given example.

We must also ensure that the examples to elicit present-tense verbs when we run the LLM on the examples. If the LLM produces mostly past-tense verbs, for example, we will not be able to observe whether the LLM believes the subject to be singular or plural, as past-tense verbs tend to be identical for all persons. Diagnostic experiments showed that LLMs indeed tended to produce past-tense verbs on this dataset; to solve this, we add the word "nowadays" to the end of each sentence, producing examples like "The king [MASK] the lawyers nowadays". After this addition, more probability mass is assigned to present-tense verbs.

Experiment We apply the probing intervention earlier described on the diagnostic dataset. For each probe, we extract its weights and biases. Then, for each LLM, and each layer in the LLM, we iterate over the examples in the dataset. We feed the example into the LLM, and apply the chosen intervention to the representation of the example's subject. If the subject consists of multiple tokens, we apply an intervention to each token. Then, we record the masked word probabilities generated by the LLM.

We experiment with 3 interventions: projection, reflection, and α-reflection ($\alpha = 0.05$). We also record a baseline for each LLM where no intervention was performed. To evaluate the probabilities produced by the LLM for each intervention, we use the method from Ravfogel et al. [2021]. That is, we organize each word in the LLM's vocabulary into one of three sets. These are *agree*: present tense verbs that agree in number with the original subject; *disagree*: present tense verbs that disagree with the original subject; and *other*: those words that are not present-tense verbs. Then for each set, we sum the probability assigned to the words therein, which we denote P_{agree}, $P_{disagree}$ and P_{other}. In total, these sum to 1. Note that LLM vocabularies include sub-word tokens that

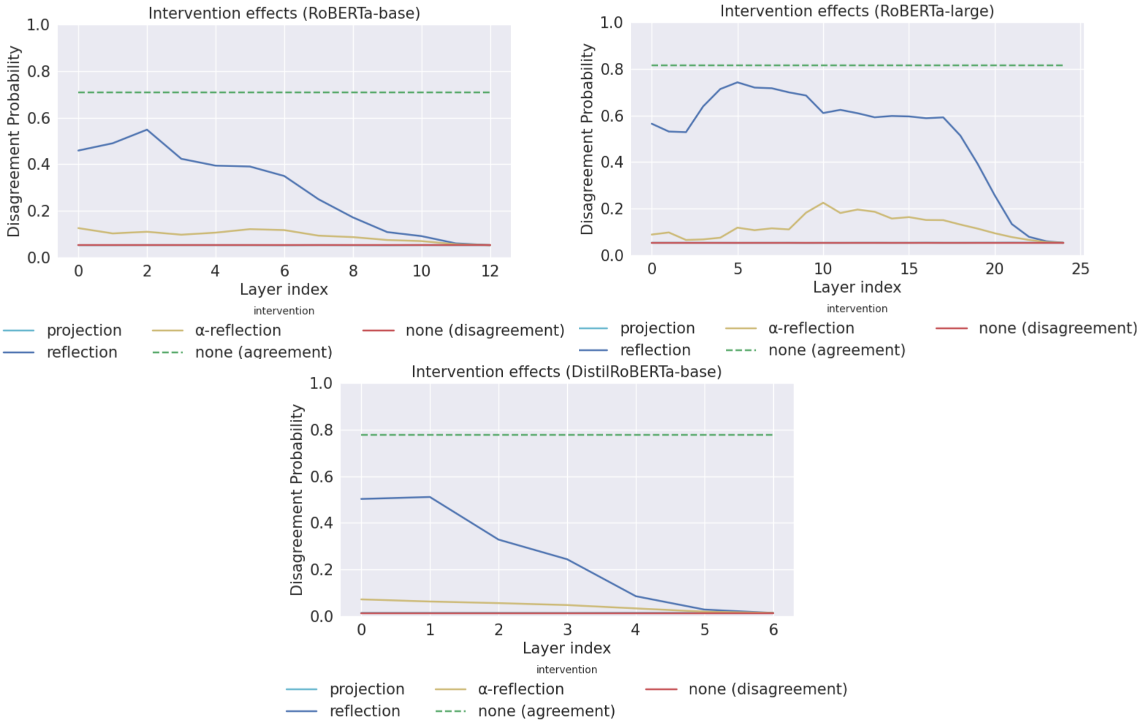


Figure 4.3: Effects of geometric probing interventions on RoBERTa models. All lines represent disagreement, except for the dashed green line, which is baseline agreement when no intervention is performed.

are word fragments; these are categorized as *other*.

Predictions A successful projection intervention will remove all plurality information as used by the LLM from the subject noun's representation. Consequently, we expect that the LLM should predict verbs that agree and disagree with the original subject with equal probability, i.e. we should have $P_{disagree} \approx P_{agree}$. In contrast, successful reflection and α-reflection interventions will trick the LLM into believing its subject has a different number, thus increasing $P_{disagree}$ and decreasing P_{agree}. Finally, if in any case P_{agree} remains high, or only P_{other} increases, we can infer that the model still believes in the original subject number, or that the representation has been otherwise corrupted.

Results In Figure 4.3, we report the results of the interventions on three different RoBERTa-based models. Disagreement ($P_{disagree}$) when projection, reflection, and alpha reflection are performed are displayed in light blue, dark blue, and yellow respectively. Projection effects are always near 0, so in practice, any blue on the graphs is dark blue (reflection). Additionally, dashed green shows the baseline *agreement* when no intervention was performed, while red shows baseline disagreement.

We include baseline agreement because this provides a reasonable upper bound on intervention performance; interventions should not be expected to engender more disagreement than there was agreement initially. Note that an increase in disagreement does not necessarily entail a decrease in agreement, because it is also possible that P_{other} decreased. In practice, we observe that agreement does decrease when disagreement increases, and for that reason plot only disagreements along with the two baselines.

The projection intervention is broadly ineffective. The effect of this intervention is almost nothing: disagreement stays entirely at baseline (near 0). Given the success of other interventions, this is somewhat surprising: if reflection and even alpha have some effect, projecting onto the boundary learned by the probe should have some effect. However, we must also consider the possibility that the space around the learned linear boundary is not typically used by the LLMs; most words fall either on the singular or plural side of the boundary. Thus, representations in the boundary area yield unusual effects.

The reflection intervention's effects differ between models. It is most effective in the early layers of the models, making disagreeing forms even more probable than agreeing forms in the first few layers of the distilled and base models. In contrast, in the large model, the effects last longer, all the way until the last fifth of model layers, and the probabilities assigned to disagreeing verbs rise almost as high as the original agreement baseline.

These results indicate that our intervention was effective in the reflection case. Flipping the embeddings over the probes' decision boundary changed the model's behavior in a way that directly corresponds to the information probed for. However, the α-reflection intervention is less effective. Although it slightly increases disagreement in early and middle layers, disagreement levels never come close to the probability assigned to agreeing forms. This indicates that that the probes have not precisely found the decision boundary: while embeddings reflected ($\alpha = 1$) across the decision boundary had a great effect, embeddings just barely pushed over the boundary ($\alpha = 0.05$) produce very little effect. Future experiments could interpolate over values of α to find a value reflecting the true decision boundary.

4.3 Targeted Swap Intervention

The first intervention did produce changes in model behavior; however, the magnitude of these changes is mixed: large in large models, and smaller in other models. While the interventions worked successfully in early-mid layers, they are not as successful in later layers. What are the reasons behind these trends? This question is difficult to answer using the probing intervention methodology from the previous section. It is possible that these trends are an inevitable consequence of how LLMs process their input. However, it is equally possible that something is wrong with our probes, and they do not capture relevant plurality information in later layers; this would also lead to lower performance in later layers.

In order to determine which of these is the case, we need a causal intervention method that does not rely on auxiliary models. Thus, we employ the interchange intervention methodology [Geiger et al., 2021]. In this strategy, we do not project the representation of the subject onto a plurality probe's null space or reflect it across the decision boundary. Instead, we replace the subject's representation with an in-context representation of the same subject, with the opposite plurality.

For example, a swap intervention might replace the representation of "king" in "The king likes the queen." with the representation of "kings" in "The kings like the queen." With this methodology, we know that the replacement representation truly represents the plural noun in the correct context. This intervention acts as a sort of upper bound on the performance of reflection interventions. Reflection interventions yield opposite-plurality subject representations, but these are surely no better representations of the opposite-plurality subject than those taken directly from the LLM. Thus, we expect effect sizes to be larger for the latter than the former

Fortunately, such swap interventions are simple to implement. We augment the dataset used in the prior experiment by adding the opposite-plurality version of each sentence in the dataset, so that each sentence in the dataset is available with the subject in the singular, and with the subject in the plural. We eliminate those sentences in which the opposite-plurality subject has a different number of tokens than the original subject, as this makes swapping impossible. Then, for each example (pair of sentences) in the dataset, we first run the model on the opposite-plurality sentence, and store the subject representation (at all layers). Then, for each layer, we run the model on the original sentence, but replace the subject with its opposite-plurality representation at the specified layer.

Other experimental details remain the same: we evaluate by calculating P_{agree} and $P_{disagree}$, and run these experiments on the same models as used before. Since there is no training in this intervention, we only evaluate, on the test set.

Predictions If the LLM is truly using only the plurality information contained in the subject noun representation to determine which verb form to use, then swapping the subject's representation with the representation of the same word with the opposite plurality should have a drastic effect. Specifically, $P_{disagree}$ should rise far above P_{agree} in almost all layers, excluding the final layer, which is used for generating the logits for predictions.

If the swap interventions succeed, then the probing interventions' small effects in later layers are due to probe issues: if swap interventions could engender significant effects in later layers, and probing interventions could not, then the probes' decision boundaries may be at fault. Of course, it is possible that no linear decision boundary exists that separates singular and plural subject representations in later layers, in which case it is not probe training but probe architecture / hypothesis class that is at fault.

Alternatively, it is possible that the swap intervention will show similar trends to the probing interventions, failing at higher layers. In this case, the failure of the probes is likely not due to probe issues. Instead, this trend likely reflects underlying model processing: we would conclude that in later layers, models simply do not rely on the subject representation when generating subject-verb agreement predictions.

Results We report the results for the swap experiment on the same 3 models as in the prior experiment, in Figure 4.4. Baseline agreement and disagreement are in dashed green and red respectively; swap is in purple, and reflection is provided in blue for comparison. The trends of the interventions' effects are strikingly similar. In small models, the disagreement peaks early and dissipates in later layers. In the large model, disagreement remains high even when the intervention is performed later, shrinking only at the very last layers of the model. In no model

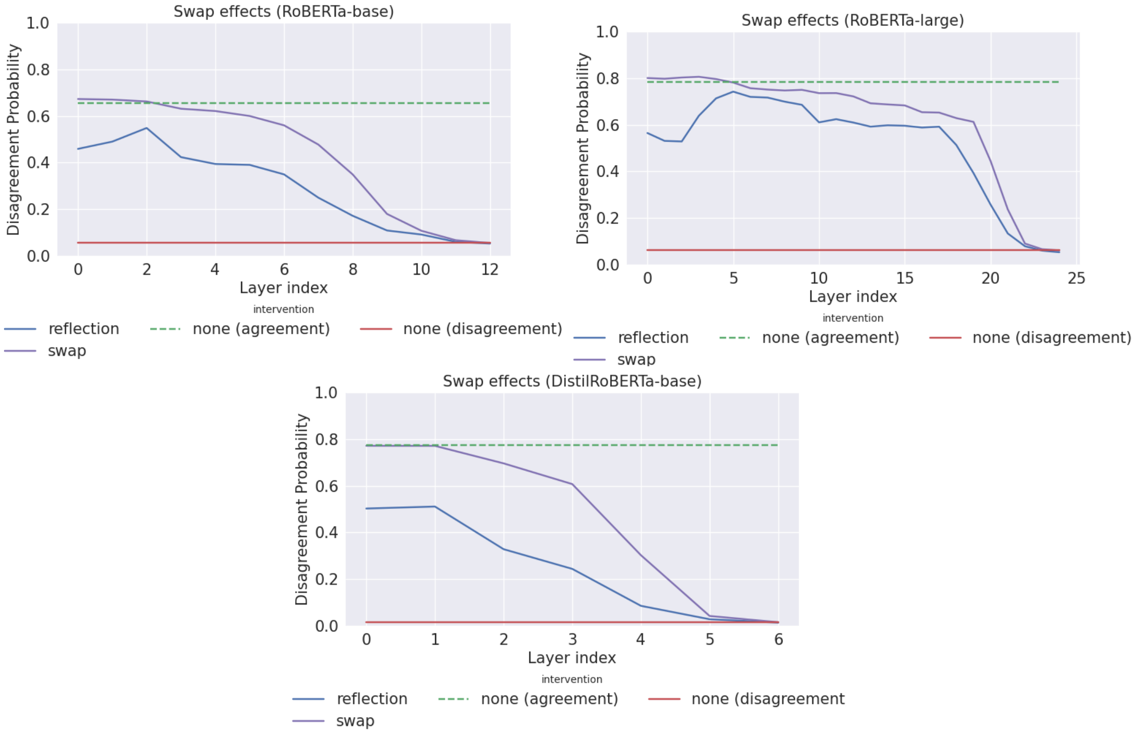


Figure 4.4: Effects of swap on various RoBERTa models

is the swap intervention successful where the probing intervention failed entirely

These results indicate that the probing interventions' ineffectiveness in later model layers did not reflect a limitation with probe architecture or training. Rather, it seems more likely that the model does not use subject plurality information in the subject token(s) at all in later layers, despite using it heavily in the early-mid layers. We conclude that that probing interventions' small effect size in late layers is due to LLM processing effects, rather than probe issues.

Despite this, there are some noticeable effect size differences between the two interventions. The effects of the swap intervention are more pronounced than those of the reflection intervention, especially in smaller models. For the very first layer, this makes sense: performing the swap intervention is identical to inputting the same sentence with an opposite-number subject. Correspondingly, the disagreement created by swap interventions in the first layer is at or above baseline agreement.

Still, reasons for the higher performance of swap than reflection in general are unclear. In all likelihood, probes do not learn exactly the right linear decision boundary, or such a decision boundary does not exactly exist. While past work has found clustering of e.g. word senses in LLMs' embedding spaces [Wiedemann et al., 2019], LLMs need not rely on a clean, linearly-separable encoding of noun number, even if probes can find a relatively accurate decision boundary. This hypothesis also agrees with our finding that those types of probing interventions that rely heavily on the exact decision boundary being correct, are ineffective.

Regardless of why exactly swap interventions outperform reflections, we can conclude from these experiments that while our reflection interventions were effective, they were still not maximally effective. Considering that the disagreement generated by our swap intervention is an upper bound on intervention effectiveness, there remains some room for improvement in our reflection interventions.

4.4 Probing for Diffused Information

Both the reflection experiment and the swapping experiment were effective at changing our models' predictions. However, one thing is curious: both interventions were more effective earlier in the models' layers; at later layers, even swapping did not change models' predictions. How can this be?

In the foregoing sections, we naively assumed that models stored and relied on subject plurality information exclusively in the subject itself. But this may not be the case: Klafka and Ettinger [2020] show, using the diagnostic dataset we have been using, that subject plurality information is recoverable from all tokens in the sentence, at least in the two latest layers of BERT. This provides a possible solution to our mystery regarding interventions' poor performance in later layers. Perhaps BERT is using the information stored in these other tokens to reconstruct the true plurality of the subject, and making its predictions based on that.

We can test this hypothesis using the suite of methods developed in the preceding sections. While Klafka and Ettinger [2020] show that the information is present in the last two layers, we will probe for this information in all layers. We will again probe the diagnostic dataset, and do so using linear probes. Note that this experiment will test for the presence of relevant information; to test if this information is used, we will again need causal interventions.

Experiment Consider that each sentence in the dataset has the form "ART1 SUBJ VERB ART2 OBJ nowadays", where ART1 and ART2 are both "the". Then, the implementation of this experiment is similar to the first phase of the probing experiment. As before, we train probes for every model and model layer. However, instead of only training one probe for every model / layer, to predict the plurality of the subject from the representation of the subject (SUBJ), we train 5 probes, which predict the plurality of the subject from each of [ART1, SUBJ, VERB, ART2, OBJ]. So, one probe might be trained to predict SUBJ's plurality given a representation of OBJ from RoBERTa's first layer. See Figure 4.5 for a visual explanation. Thus, for a model like `roberta-base` with 13 layers, we train 65 probes: one for each of the 5 words of interest, for each of the 13 layers. We train and use linear probes as done previously, and use representations from the same models as in prior experiments.

Predictions The hypothesis we are investigating is that in later layers, LLMs use diffused information to reconstruct the subject noun's plurality, even when we perform an intervention that alters its representation. If this is the case, we predict that diffused information should be recoverable using probing from later layers (as long as it is encoded in a linearly extractable fashion); indeed, we know that this is the case for the last two layers. However, the layers from which it is extractable could yield important insights. If this information is extractable only at later layers, this supports the hypothesis that LLMs use it for subject plurality

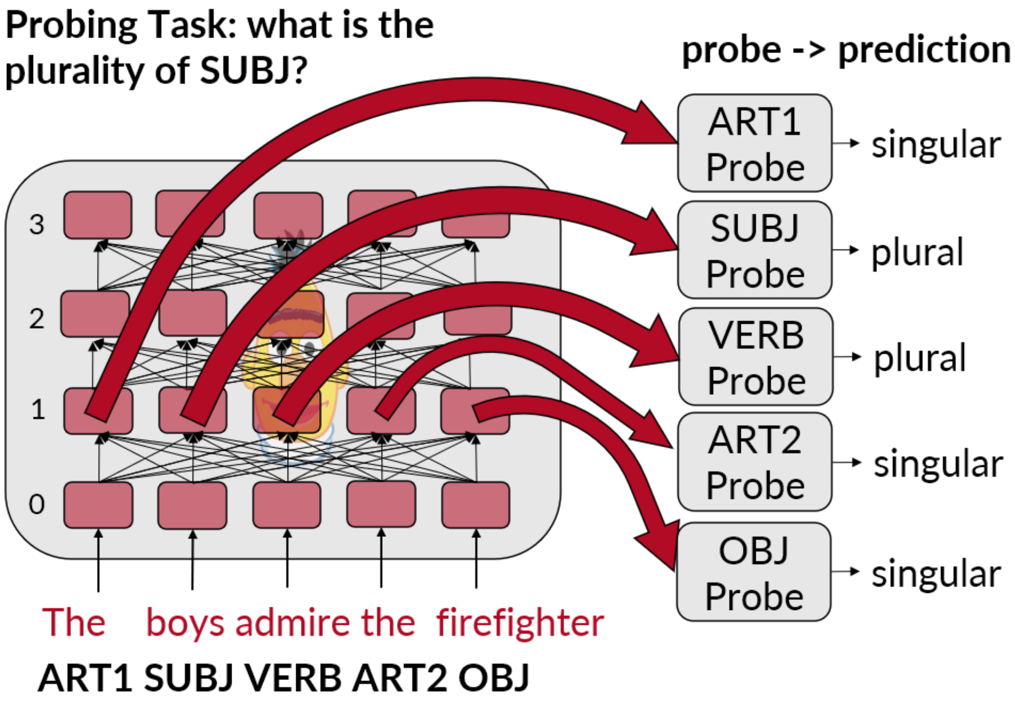


Figure 4.5: A visual representation of the probing setup, in which all words' representations are probed for subject plurality information at layer 1. Note that the [CLS] and [SEP] tokens, as well as the "nowadays" at the end of the sentence are omitted for brevity, and are not probed.

reconstruction at that point. However, if it is extractable at all layers, then we are left with another question: if this information is used as hypothesized, why is it used only in later layers?

Note also that the extractability of this information should vary somewhat from word to word. As shown previously, the subject representation contains subject plurality information, but the verb as well is an interesting case. Because the verb is conjugated according to the subject's plurality, the plurality of the subject noun should definitely be extractable from its representations. At the same time, during probing interventions, the verb is masked, so the token itself should contribute no information in that scenario. In contrast to these, ART1, ART2, and OBJ have (at the surface level) no indication of the subject's plurality; thus, if subject plurality information is extractable from them, we will consider this to indicate that information diffusion has occurred.

Results We report in Figure 4.6 the test accuracy by model of probes trained to predict subject plurality from non-subject words. For all words and models, performance is highest in later layers. However, how deep into the model performance improves varies depending on the word probed. While the subject's article (ART1) has consistently high accuracy beyond the earliest model layers, this is not true for other words. Both the verb and object article (ART2) reach perfect accuracy only later in the model, and not even in all later layers. The object

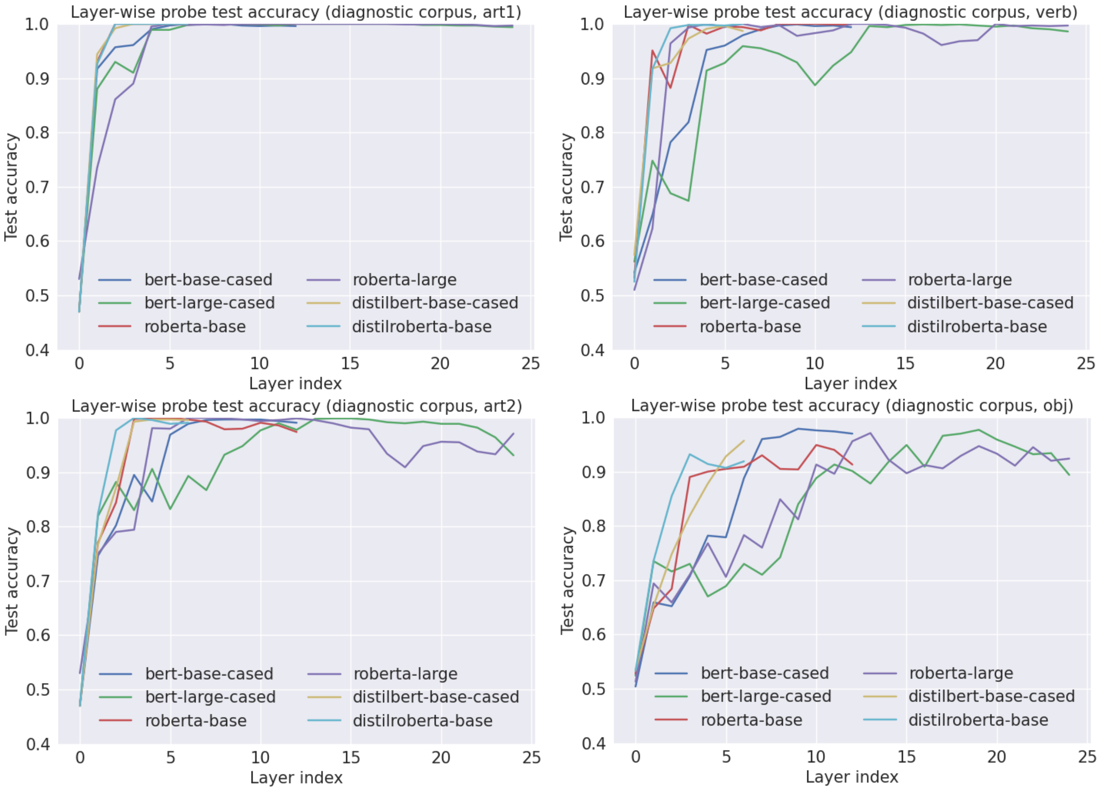


Figure 4.6: Test accuracy of probes predicting subject plurality from non-subject words in the sentence.

representation has more mixed performance, reaching its highest accuracies in the latter half of model layers, and not reaching the same accuracies as other words (although its maximum is still high, at 90%).

All words show chance-level (50%) accuracy when trained on the first layer's representations. This makes sense for non-verb representations, as the first layer of the model is the embedding layer, where no attention-based cross-token information mixing has taken place. It is somewhat curious, though, that the verb representations also have chance-level accuracy, as the verb conjugation should indicate the number of the subject. It is possible that the verb's conjugation is not directly encoded in its word embedding, and is only made available upon processing by transformer layers.

On the whole, evidence from this experiment does not strongly support the idea that the model is reconstructing the original subject's plurality in the upper layers only: the information necessary for this reconstruction is present in the lower layers too, even if it is more available in higher layers. Why, then, is it easier to change model predictions in the lower layers than in the upper layers? The answer must lie in the representations of the words that are not the subject. After all, in the swap experiment, these are the only point of difference from the case in which the sentence truly has a subject with the other plurality. To answer this question, we can apply the same intervention-based methodologies as before. This time, however, we will apply them to the non-subject words in the sentence.

4.5 Broad Probing Intervention

In the last experiment, we found evidence that plurality information, even for one specific token in the input, is in fact widely spread across input tokens. However, applying a reflection to the subject of a sentence is effective in early layers, despite the presence of information in other tokens revealing the true plurality of the subject. In this experiment, we apply probing interventions to each word type in the sentence, independently and together. Through this, we hope to reveal which word types play a role in the LLMs' understanding of plurality (as revealed by their predictions).

Experiment Having trained probes to predict subject plurality from each of the word type's representations, we now use probes to remove that information from each of the model's layers. We test removing the information from each word individually, and then some combinations: ART1 and SUBJ; SUBJ and VERB; ART1, SUBJ and VERB; ART2 and OBJ; and finally all word types. Note that since we use the LLM's prediction of the masked word token to judge model performance, we first mask out VERB, which was not masked in the probe training phase. We employ two different interventions on these representations: reflection and swap. We forgo the other two interventions due to their ineffectiveness.

Predictions If the model is reconstructing subject plurality information in higher layers from other words in the sentence, we expect that eliminating subject plurality from other words in the sentence will yield a sharp increase in disagreement, as opposed to when we eliminate this information from only the subject noun. But, it is not necessarily the case that the model uses subject plurality information from all of words in the sentence, even though they all contain this information. What are some possible hypotheses, and their corresponding predictions?

- The model uses subject plurality information only within the NP (DP) of which the subject is the head: in this scenario, the determiner, subject, and any modifiers.
- The model uses subject plurality information only within the subject and words that agree with it: in this scenario, the subject and verb should contain used subject plurality information.
- The model uses all subject plurality information distributed throughout the sentence, and is not organized in any way.

Results In Figure 4.7, we report the results of performing interventions on non-subject words; again, we report only roberta-base results as a representative sample of the models investigated. We find varying effects of performing these interventions on non-subject words. Intervening on the subject's article in produces only small effects: while it does make both interventions more effective in middle layers, it does not increase its early-layer effectiveness or make the interventions effective in later layers.

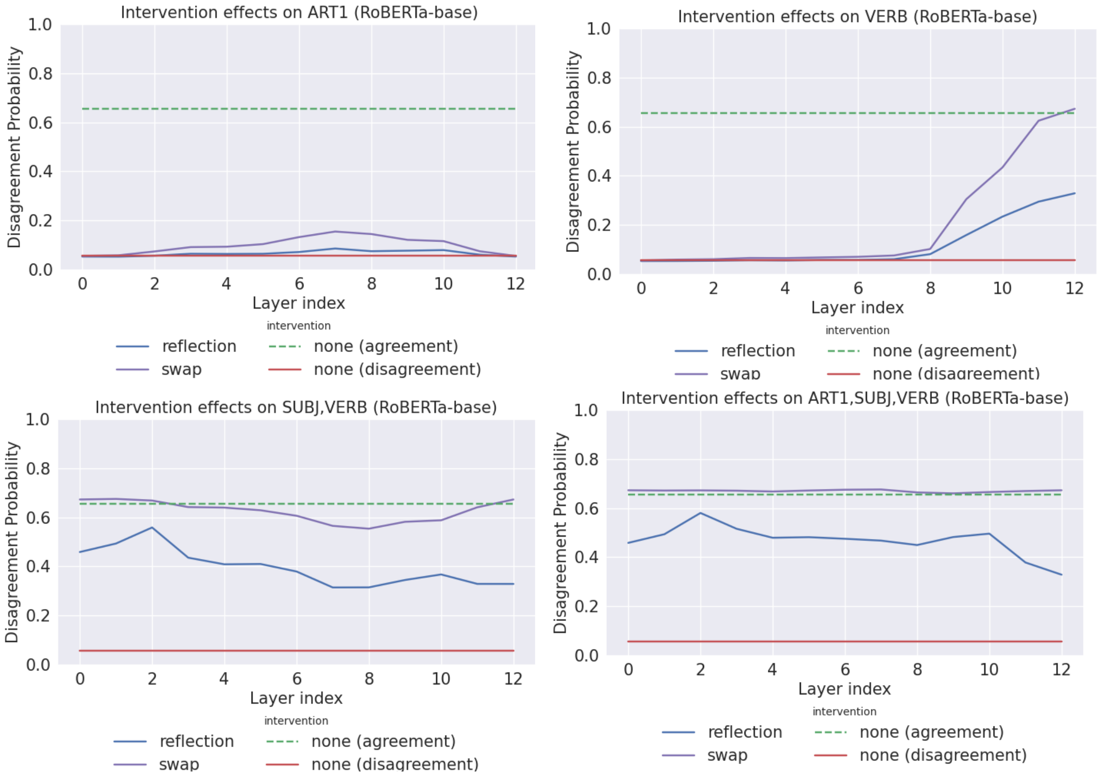


Figure 4.7: Results of the reflection and swap interventions performed on multiple words in the sentence (roberta-base).

In contrast, intervening on the verb is very effective in later layers, where intervening on the subject was less effective. Thus, it appears that the model shifts its attention from the subject to the verb over the course of its layers. Intuitively, this makes sense. When we say that we intervene on the verb, we are actually not intervening on an actual verb, but on the position of the verb, whose actual form is hidden by the [MASK] token. Because the verb is masked, in early layers, the model has no way of knowing the plurality that the verb should have. However, in later layers, the model is building up a representation of the verb, until the very last layer, where it is forced to actually predict a distribution over words in the vocabulary. Thus, in these layers, verb interventions have greater effects; naturally, in the very last layer, swapping the verb yields disagreement as high as the original agreement.

When we combine the SUBJ and VERB interventions, we find that swap performance is nearly as high as the original disagreement, and reflection is not far behind. in the middle layers, there is a slight dip in performance, but intervening on the subject article as well eliminates this. Thus, we arrive at a story of how the model processes subject-verb agreement: in early layers, it focuses on the subject, while in later layers it focuses on the verb, with some article effects in middle layers. What, though, of the object and its article? We found that intervening on these, separately or in tandem, had no effect at all.

Based on our current data, which shows that the subject and the verb are most important when performing these interventions, the most likely hypothesis is that plurality is stored in the subject and other agreeing words. However, this dataset is too simple to differentiate between this hypothesis and others that

Dataset	Sentence
Basic	The actors write the stories nowadays.
Adjective	The old oncologist hardly likes the mountain nowadays.
This / That	Those sad guests rarely tour the cafes nowadays.

Table 4.2: Example sentences from datasets generated using the BLiMP generator

are less linguistically well-motivated. For example, it could simply be the case that subject plurality information is stored in the subject noun and the following word. This hypothesis is supported by our results, but does not exemplify a neat account of linguistic processing in LLMs that we desire.

How can we arrive at such insights? To verify any one specific hypothesis, we would need very complex and structured data to ensure that our specific hypothesis about LLM processing is correct. The hypothesis space of possible techniques LLMs use to process even a relatively simple phenomenon such as noun-verb agreement is contains many hypotheses, both linguistically well-motivated and otherwise. Despite these complications, in the next section, we take some first steps towards ensuring that LLMs only subject plurality information in certain words for linguistically-motivated reasons, and not just because of those words' positions relative to the subject, as postulated as an alternative hypothesis.

4.6 Confirming Hypotheses with a More Complex Dataset

In this final experimental section, we take first steps at differentiating our chosen hypothesis, that LLMs use encoded plurality information in the subject and words that agree with it, from the hypothesis that LLMs use encoded plurality information in the subject and words nearby it. To do so, we create datasets more amenable to differentiating these two, in which the subject is not next to the words agreeing with it, and where an additional word agrees with the subject. Then, we perform causal interventions as done previously.

Dataset Creation We create our dataset using the dataset creation tool used to make the BLiMP dataset [Warstadt et al., 2020], which we re-implemented using the Python package `pandas` [Reback et al., 2020]. This tool enables the random sampling of relatively simple sentences from a vocabulary of 614 nouns and 2731 verbs, as well as the imposition of constraints upon these sentences. The tool also imposes constraints on which words may be composed together, to ensure that the resulting sentence is syntactically and semantically valid, if not especially plausible. For example, the verb "write" will only be combined with an animate subject, and an object that is a document.

Using this tool, we create three additional datasets that we use to ensure the validity of our previous findings. The vocabulary is limited such that every word type (SUBJ, VERB, OBJ, etc.) consists of one word only, as opposed to e.g. multi-word compounds such as "college student". Additionally, verbs are limited to simple transitive verbs not requiring any prepositions or other complements besides a direct object.

Each dataset consists of 6000 sentences labeled with its subject's number, as well as each sentence's opposite-plurality counterpart, for use with the swap intervention. We organize these sentences into splits (train: 4000 / valid: 1000 / test: 1000). We create the following datasets:

1. A dataset near identical to the initial diagnostic dataset. That is, this dataset consists of sentences of the form "The SUBJ VERB the OBJ nowadays." We create this dataset in order to ensure that this methodology is robust to differences in vocabulary, as this tool's vocabulary is wider than that of the diagnostic dataset. For example, unlike in the first dataset, the nouns are not all humans, and the verbs vary correspondingly.

2. A dataset similar to the initial diagnostic dataset, but with adjectives in between the determiners and subjects, and adverbs between the subjects and verbs. That is, sentences take the form "The [ADJ] [SUBJ] [ADV] [VERB-PRESENT] the [OBJ] nowadays". With this dataset, we will see examine whether models use plurality information in the verb and subject's article only because they are close to the subject.

3. A dataset building on the prior dataset by also including number agreement in the article of the sentence. It does so by using the determiners "this/these" and "that/those' for the subject'. So, sentences take the form "[This/These/That/Those] [ADJ] [SUBJ] [ADV] [VERB-PRESENT] the [OBJ] nowadays". By introducing another source of information regarding the plurality of the subject noun, we aim to see if LLMs are truly focusing on words that are relevant to the plurality of the subject, i.e. those that agree with it in number.

Examples from each dataset can be found in Table 4.2.

Experiments For each of the three datasets described above, we train probes to predict plurality each of the word types. That is, just as before, a different probe is trained on each word of the sentence ([ART1, SUBJ, VERB, ART2, OBJ, ADJ, ADV]). As before, we use these probes to perform the reflection intervention on each layer of a variety of models (the same selection as before). We also use the opposite-plurality sentences to perform the swap intervention on all layers of tested models. We continue to measure the effect of the interventions using the disagreement engendered in the model compared to the no-intervention setting.

Due to the increasing length of sentences, we cannot intervene on all combinations of words; this scales combinatorially. Instead, we intervene on all words individually, and then also on all words together. Then, we select combinations of words that are of relevance (specifically, ART1, ADJ, SUBJ, ADV, and VERB), and intervene on those. In particular, we focus on the combination of SUBJ and VERB (as these must agree), as well as ART1, SUBJ, and VERB for the same reason in the third dataset. We generally omit ART2 and OBJ from our interventions due to their observed unimportance in prior experiments.

The remaining experimental details (probe architecture and training details, and models used) remain the same as in prior experiments.

Predictions With these experiments, we aim to show that LLMs use the information stored in the subject and words that agree with it. How will this manifest in each of the datasets?

1. When we perform interventions while processing sentences from the first dataset, results should be largely the same as in prior experiments, as the dataset is also largely the same. So, we should see interventions have greater effects when performed on the subject and verb, less of an effect when performed on ART1 (which, linguistically, is associated with the subject), and no effect when performed on the object and its article. If these intervention techniques are not robust to vocabulary shifts, and work only on the extremely simple diagnostic dataset, we expect that interventions may yield no effects at all.

2. Interventions using the second dataset should show largely the same trend: targeting the subject and verb should yield the most significant effects, even if they are further separated, because they are relevant to the subject's plurality. In contrast as the adjective and the verb do not agree with the subject, intervening on them should have no effect. An alternative hypothesis might posit that LLMs use plurality information from words near the subject: were this the case, we would expect large effects from interventions on the adjective and adverb due to their proximity to the subject.

3. Interventions using the third dataset should show strong intervention effects not just in the subject and verb but in ART1 as well. By our hypothesis, as the subject's article agrees with it in this dataset, the article should be a source of plurality information on which the model relies to choose the proper verb conjugation. If this hypothesis is false, we expect that the intervening on ART1 will continue to be ineffective.

Results For all three datasets, probe training was successful, with all probes achieving high ($\approx$ 90% or above) on all word types during training. Having succeeded in the first step, we report intervention results by dataset:

1. We report the results of the first experiment, which is similar to the prior experiment, but with a slightly more diverse dataset, in Figure 4.8. We find that results are largely the same: intervening on the subject is effective, more so in the early-mid layers. However, in the later layers, there is a precipitous drop in efficacy. Intervening on the verb as well increases disagreement to the greatest extent, as seen in previous experiments.

   As before, intervening on the subject's article has very little effect, and intervening on ART2 and OBJ have no effect at all. The results overall are quite similar to those of the prior experiment, although the decrease in SUBJ intervention effectiveness is somewhat more sudden than the gentle decrease observed in prior experiments. Still, the broad similarity between results indicates, as per our predictions, that this finding was not affected by small changes to the data distribution.

2. The results of the second experiment, which added an intervening adjective and adverb into each sentence of the dataset are in Figure 4.9. We can

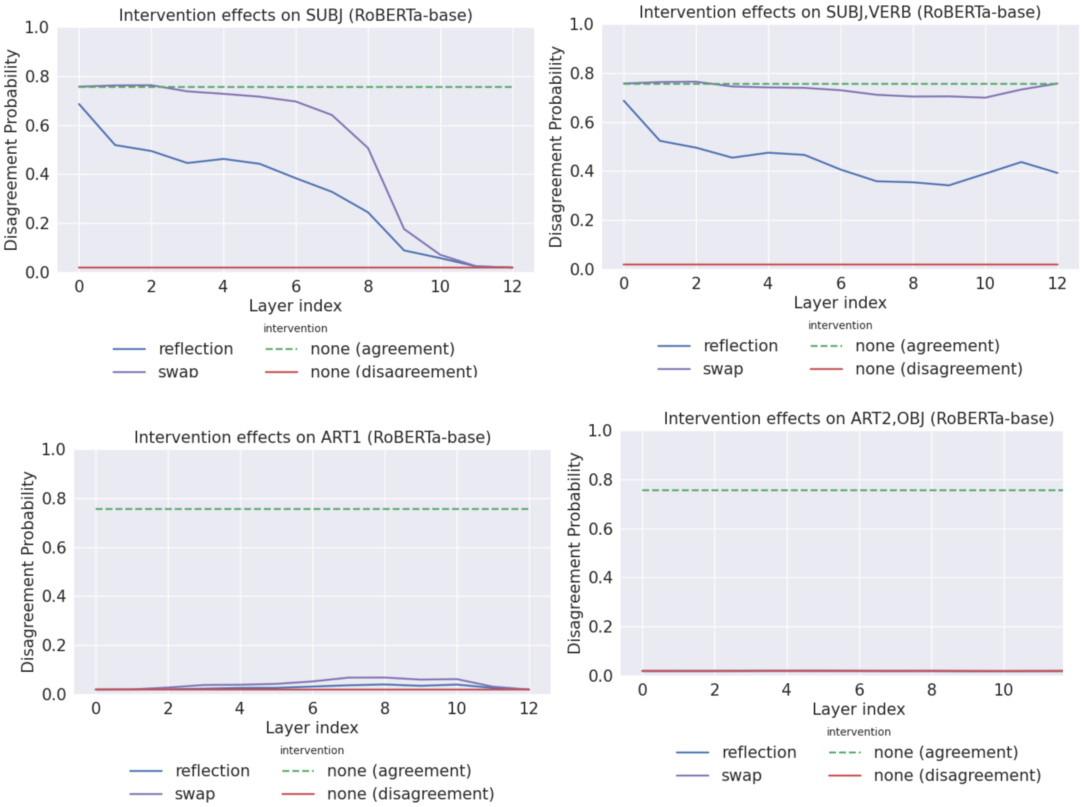


Figure 4.8: Results of the reflection and swap interventions performed on the basic BLiMP dataset (roberta-base).

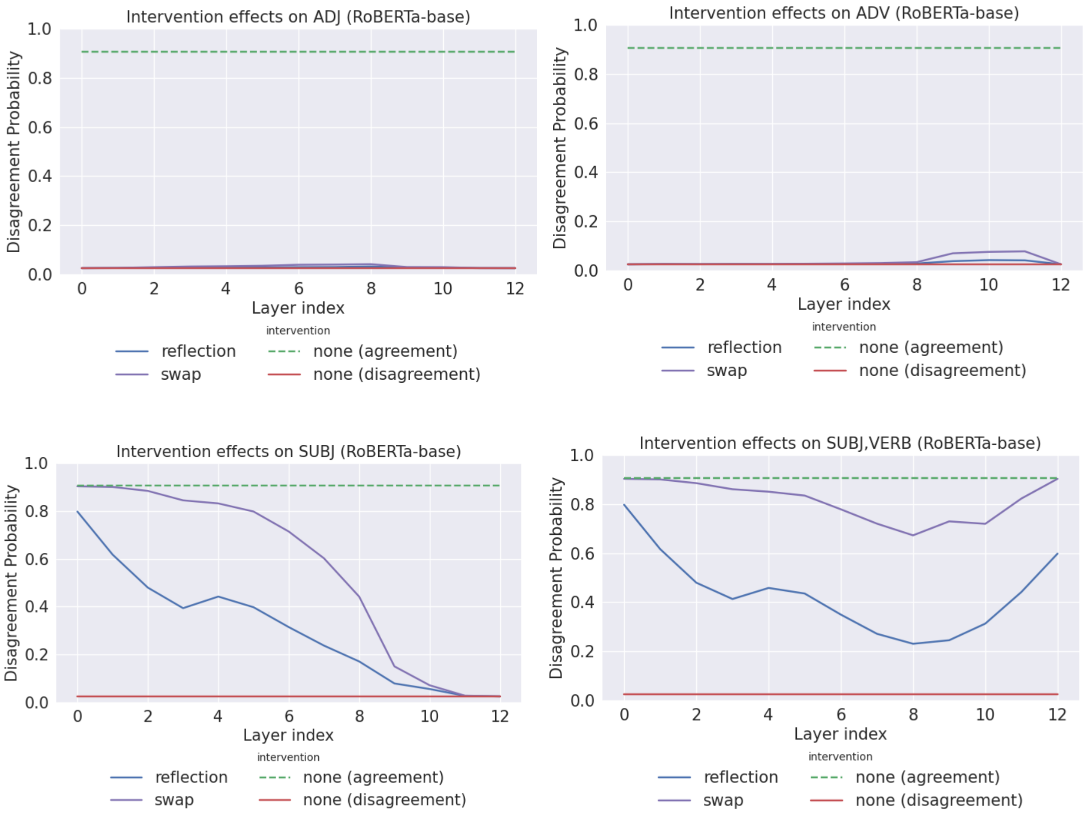


Figure 4.9: Results of the reflection and swap interventions performed on the BLiMP dataset with intervening adjective and adverb (roberta-base).

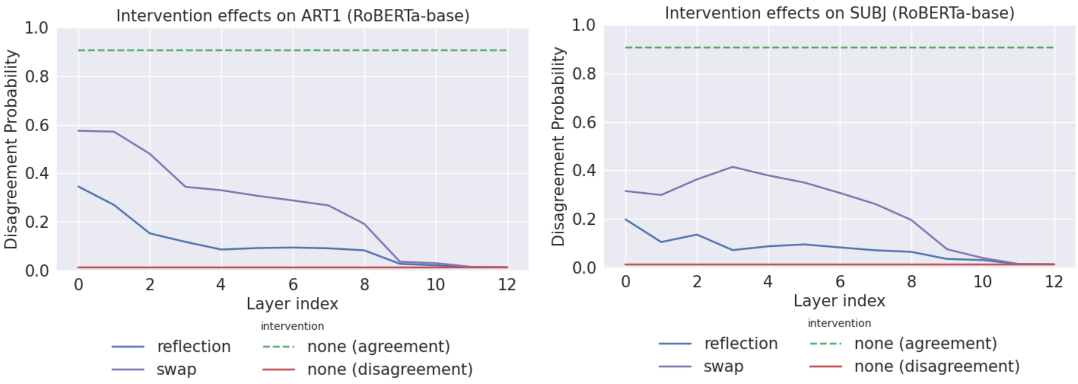


Figure 4.10: Results of the reflection and swap interventions performed on the BLiMP dataset with article agreement (roberta-base).

see that the model does not rely at all on the adjective when determining the form of the output verb; it relies only slightly on the adverb. Thus, intervening on these has no effect despite their proximity to the subject.

In contrast, intervening on the subject is still effective, and intervening on subject and verb yields even greater effects. It is notable, however, that the effect sizes of both the subject intervention and the subject-verb intervention are somewhat lower than in the prior dataset, and in prior experiments. This is most visible in the middle layers of the model, right where ADV interventions have a small but non-zero effect. We hypothesize that this occurs because the model is paying attention to the adverb instead of the subject or verb, indicating a minor error in how it processes subject-verb agreement. While in this case, the effects are small, repeated errors such as this could add up, creating mistakes in large and complex sentences. This phenomenon is a promising subject for future work.

3. The results of the third experiment, which changed the subject article from the invariable “the” to the number dependent “this / these / that / those”, are in Figure 4.10. The most notable finding is that performing the reflection and swap interventions on ART1 is drastically more effective than in the prior cases, where ART1 did not exhibit any subject agreement. The swap intervention, even in layer 1, makes disagreement more likely than not, indicating that the LLM is in fact relying on the plurality information in the article. This is in stark contrast to past experiments, where what little effects article intervention had on disagreement were found in mid layers. Here, intervening on ART1 causes moderate mid-layer effects with the swap intervention, but only mild effects for reflection.

   Meanwhile, the effect of intervening on the subject alone is significantly decreased, compared to the last experiment. While previously, the earliest layers were the most effective targets for intervention, they are now less effective than early-mid layers. Notably, this is the inverse of the trends in ART1. This suggests that the LLM is relying on ART1 in earlier layers, and on SUBJ in early-mid layers. Thus, there seems to be a trade-off effect, as seen less strongly with ADV in the prior dataset. Since the LLM is relying more heavily on ART1 in early layers, it relies less on the subject; however, the effect of intervening on ART1 and SUBJ together remains constant.

In summary, our three experiments yielded positive results, confirming three predictions made about how these interventions would generalize. In the first experiment, we saw that the chosen probing intervention techniques do generalize to a new dataset. In the second experiment, we added an adjective before, and adverb after the noun, and found that the model relied little on the information encoded in the, adjective and adverb, but continued to rely on the subject and verb. This suggested that LLMs are not relying on the subject and verb's encoded plurality information solely due to their proximity. In the third experiment, we introduced another instance of agreement in each sentence, and found that LLMs do in fact use the plurality information encoded in it. All of these together provide evidence for our main hypothesis: that LLMs encode subject plurality information in all words, but use only the information encoded in those words that exhibit agreement.

5. Discussion

Having conducted numerous different experiments, and presented their results, we now in this section synthesize and contextualize these results. In particular, we discuss two distinct questions. First, what have these experiments taught us about probing? And second, what have these experiments taught us about subject-verb agreement and plurality in LLMs?

5.1 Probing

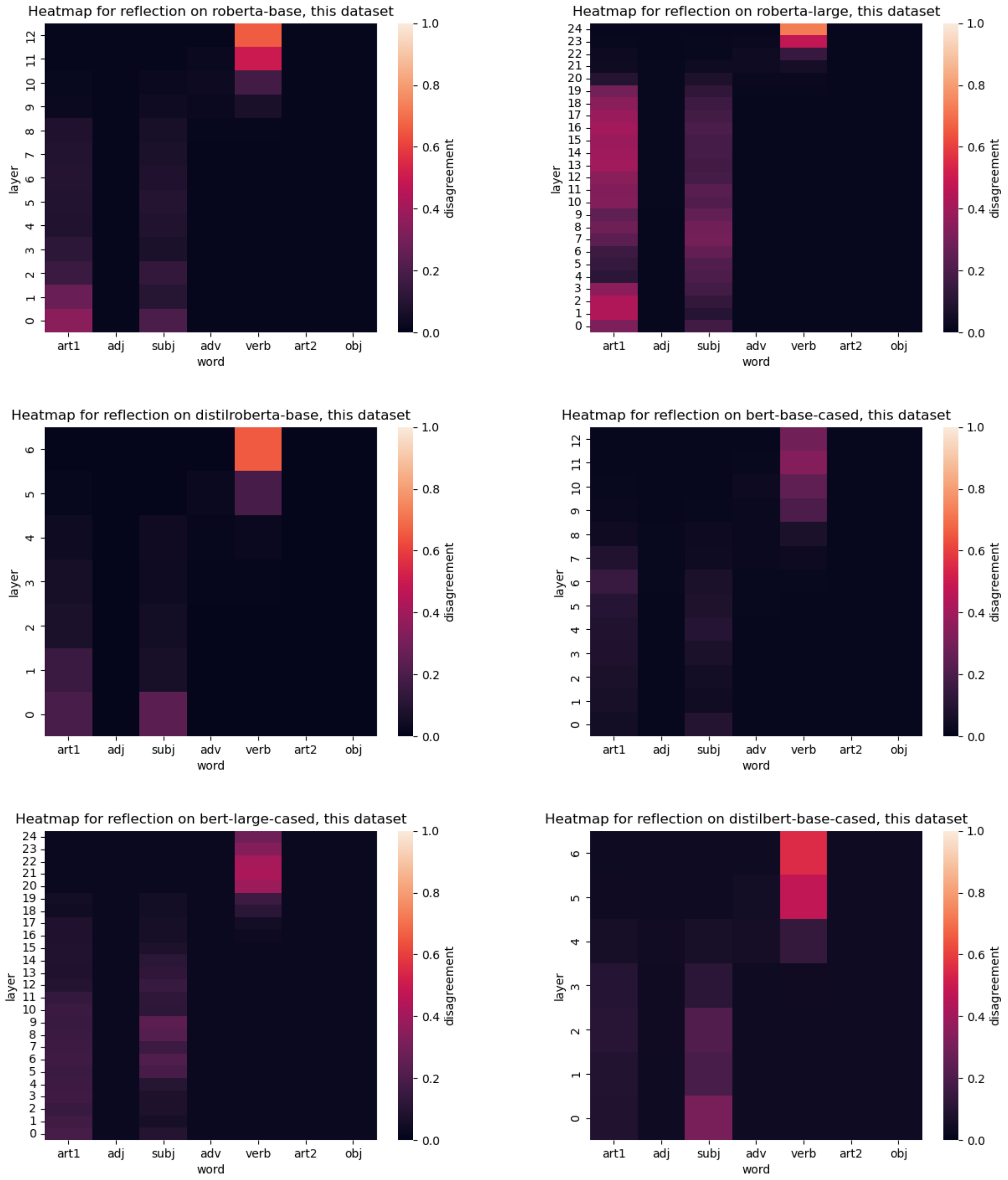
With respect to probing, our experiments have focused on one main question: to what degree do probes capture encoded subject plurality information in a way that reflects LLMs' usage thereof? Results here are mixed: in Section 4.4, probes found subject plurality information in all words of the sentence, but in Sections 4.5 and 4.6, we found that only information in certain words is actually used. This suggests a somewhat negative result—certainly it is the case that probes capture functionally irrelevant information encoded in LLMs' representations. This is consistent with earlier work [Elazar et al., 2021] that suggests the same thing: probes do not consistently capture functionally relevant information.

However, it is not the case that all of the information captured by probes is irrelevant. When targeting words that encode subject plurality information used by LLMs, the reflection probing intervention does work, although its effects are less than that of the swap intervention upper bound. Moreover, compared to prior work [Ravfogel et al., 2021] that uses a high-impact intervention (α-reflection where $\alpha = 4$) for small effects, we are able to elicit large effects with a minimal intervention ($\alpha = 1$).

Still, this statement must be tempered by the uneven effects of the various types of probing interventions. While the reflection intervention is effective, projection and α-reflection interventions (specifically for $\alpha < 1$), are not. This indicates that while probes learn to extract some functionally relevant information from word representations whose information is used, the probes are not perfect. That is, the decision boundary learned by the probes and used in geometric probing interventions is not the one used by LLMs, if such a boundary exists.

Another aspect that must be considered is the effectiveness of these probing methods across models. While in the preceding sections, we report results primarily for roberta-base, reflection interventions should be effective across various models if we are to draw the conclusion that they capture subject plurality information in a functionally relevant way.

In Figure 5.1, we present an inter-model comparison of our most complex experiment, where sentences have the form "ART1 ADJ SUBJ ADV VERB ART2 OBJ nowadays." and the first article agrees with the subject. For each model we present a heatmap, where the x axis is the word in the sentence, and the y axis is a model layer. The value at x, y is the disagreement caused by performing a reflection intervention on the representation of the word x at layer y. These values are averages taken over all examples in the dataset. Models may have distinct numbers of layers, but heatmaps are scaled to be the same length, and their color ranges correspond to the same range of disagreement (0-1.0). Note



Heatmap for reflection on roberta-base, this dataset

Heatmap for reflection on roberta-large, this dataset

Heatmap for reflection on distilroberta-base, this dataset

Heatmap for reflection on bert-base-cased, this dataset

Heatmap for reflection on bert-large-cased, this dataset

Heatmap for reflection on distilbert-base-cased, this dataset

Figure 5.1: Effects of geometric probing interventions on various models. The value at $word, layer$ is the disagreement caused by using a probing intervention on the given word, at the given layer

that the base agreement for these models is around 0.8, so although models do not reach 1.0 disagreement, this is expected. Finally, note also that these diagrams only display the effects of interventions on one word, not multiple at a time as in prior experiments.

In all models, we see some patterns emerge. There are three words on which interventions have an effect: the first article, the subject, and the verb. In all models, the verb's intervention effect is the highest of all of the words', and occurs only in the last few layers of the model (the exact number of layers scales with model depth). In contrast, the subject and its article yield effects only when intervened on in the remaining (beginning-mid) layers of the model.

However, differences quickly begin to emerge as well. The exact magnitude of intervention effects (reported as averaged over the entire dataset) depends heavily on the model. In the most mild case, the verb, we see that intervention effects are greater in RoBERTa-based models than in BERT-based models, especially bert-base-cased. But effects are even greater in the other two words. While in some models (roberta-base and roberta-large), the magnitude of ART1 interventions is moderate, and stronger than that of SUBJ interventions, in others (bert-base-cased and distilbert-base-cased) it is much weaker. Broadly, there seems to be a split, in which RoBERTa-based models show higher effects of ART1 interventions, and BERT-based models show higher effects of SUBJ interventions. Individual models still display individual variance, though: roberta-large shows above-average effects in all words, while effects in bert-base-cased are very muted.

It is unclear why these trends appear. The differences between RoBERTa and BERT models could be reasonably attributed to the difference in the models' training regimes. However, explaining this difference via their training regimes is challenging: the two models differ in training objective and training data / time, which have no obvious tie to the observed differences. The most likely culprit is tokenization: BERT uses WordPiece tokenization [Wu et al., 2016], while RoBERTa uses byte-pair encoding [Sennrich et al., 2016]. Because subjects were not constrained to be one word in the intervention trials, it is possible that some subjects were tokenized differently in BERT and RoBERTa models, leading to different probes and different performance when intervening on the subject.

We confirm this difference in tokenizations by examining the test set of the dataset used to create the heatmaps. In this sample, 440 out of 1000 examples had distinct subject tokenizations between BERT and RoBERTa models. This is largely due to the extra tokenization on the RoBERTa models' part: while BERT represents 937 example subjects as single tokens, RoBERT does so for only 523. This limits the degree to which we can directly compare models; we cannot untangle differences due to tokenization from those that are due to other factors. However, we emphasize that in spite of these tokenization differences, the big-picture conclusions regarding probing are broadly similar across models.

Finally, one area in which differences notably do not appear is in distilled models, as opposed to their non-distilled counterparts. We tested distilroberta-base and distilbert-base-cased with the hypothesis that, because they were distilled, they might have less extraneous information in their representations, such as unused subject plurality information. However, this was not the case; just like other models, subject plurality information was extractable from all words at many different layers. Furthermore, the patterns in the use of this information do not differ notably from non-distilled counterparts. The BERT vs. RoBERTa trends are much stronger, and better explain behavior in each distilled model.

To conclude, we successfully use the reflection probing intervention to induce disagreement in a variety of models. However, other probing interventions (projection, α-reflection) are less effective. Moreover, the effectiveness of reflection varies widely across models; in some it is notably lower. The effects of these probing interventions are encouraging; however, considering the rigidity and simplicity of the datasets we use, and the varied effects of probing across models, more research is needed to determine the robustness of these techniques across problem settings.

5.2 Subject-verb agreement and plurality

We have also learned a great deal about subject-verb agreement and how LLMs determine the plurality of the subject noun, and thus the conjugation of the agreeing present tense verb. Through the swap intervention, we have determined that, when only the subject noun and verb demonstrate agreement, the former contributes primarily in early-mid model layers, while the latter contributes more in later layers. However, when the subject's article agrees with the subject, it also contributes in earlier layers, leaving the subject to middle layers of the model. Moreover, because we used the swap intervention as part of our interventions, we can also be sure that this analysis holds independently of how well probes reflect actual model internals; the swap intervention is probe-free.

How well do our findings about subject-verb agreement processing generalize across models? As in the prior section, we consider this in Figure 5.2. Effects are much stronger in this section than in the prior, generally speaking. However, similar results emerge. Generally, RoBERTa-based models place greater emphasis on the article than the subject, while the reverse is true for BERT-based models. Since the swap intervention does not use probes at all, this suggests that this phenomenon may be due to model behavior, rather than a simple artifact of probing. After all BERT and RoBERTa models should exhibit some different behavior, as they are known to have different performance.

In this light, we can perhaps explain the different behavior of BERT-based models as modeling failures. For example, RoBERTa-based models appropriately assign high importance to both the article and subject. In contrast, bert-base-cased and distilbert-base-cased do not react strongly to interventions on the article. The only BERT model that succeeds is the largest among them. These may point to weaknesses in BERT fixed by RoBERTa's training procedure.

Turning now to what existing literature has to say on this topic, we find that the layer-wise analysis of subject-verb agreement processing is, to the best of our knowledge, novel. Many analyses focus on one or the other of these topics, but few on both. For example, most subject-verb agreement studies are behavioral, finding that BERT is generally good at subject-verb agreement [Goldberg, 2019], but that BERT produces errors when tested with distractors [Pandia and Ettinger, 2021], or in contexts with repeated embedded phrases [Chaves and Richter, 2021]. These studies focus more on BERT's behavioral performance, without focusing on how it arrives at its predictions.

Other studies look at the layer-wise pipeline of language processing in BERT,

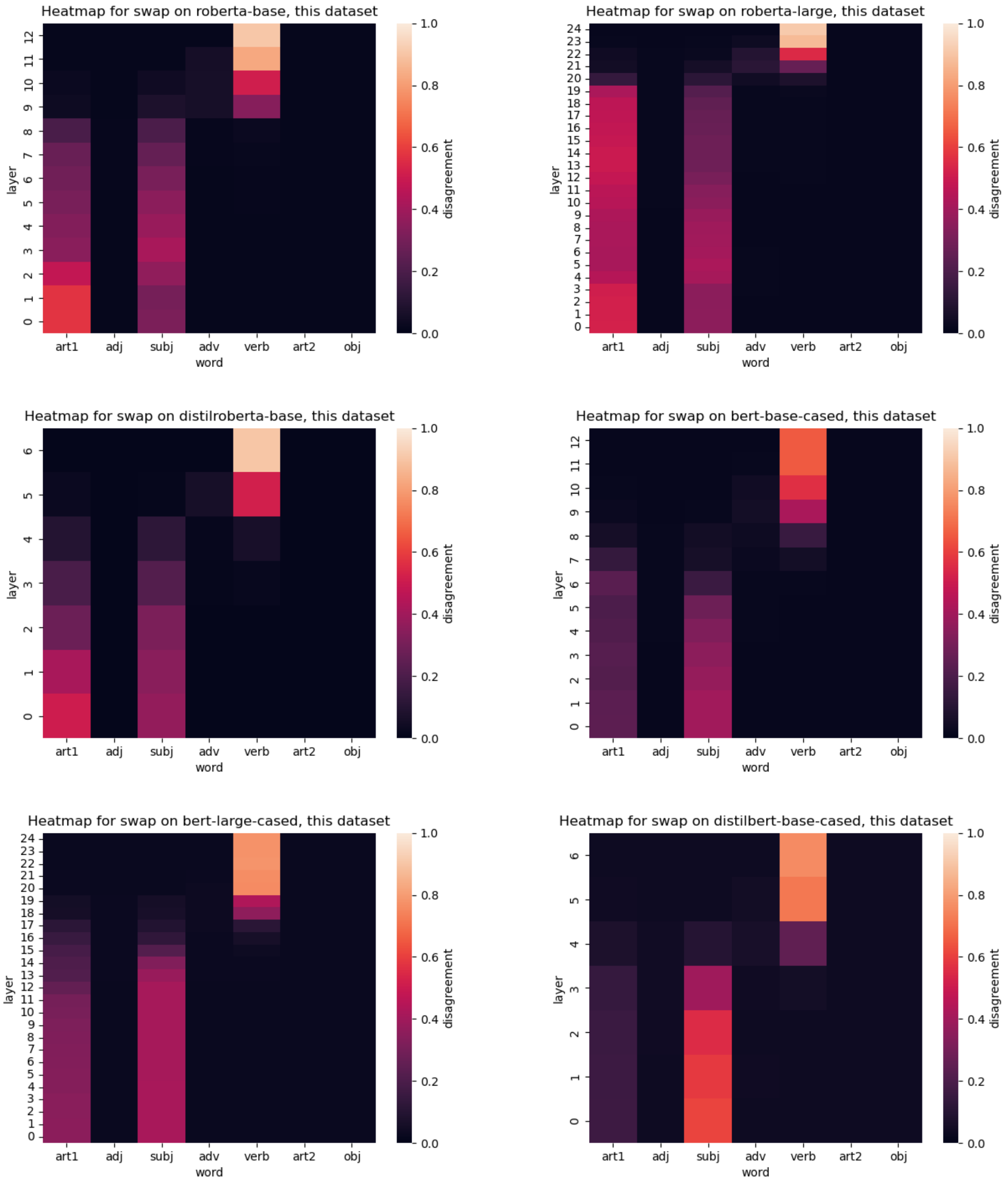


Figure 5.2: Effects of swap interventions on various models. The value at *word, layer* is the disagreement caused by using a swap intervention on the given word, at the given layer

but again comparison is difficult, because these do not take a functional approach. [Tenney et al., 2019a] find that BERT rediscovers the classical NLP pipeline, with syntax on the bottom and semantics at the top, but they find this via probing. In some sense, our findings disagree with this: we show that an effective intervention exists at all points in the pipeline, from start to finish. In contrast, a classical NLP pipeline account would put subject-verb agreement at the beginning of the pipeline only. However, it is unclear that such a direct comparison is possible: the existence of a pipeline from a probing point of view does not preclude different results from a causal intervention point of view. In the end, these causal interventions yield a distinct perspective on LLMs and their representations, best suited for answering questions about how models actually perform language processing.

In conclusion, models generally seem to place the most weight on the subject, verb, and other words agreeing with the subject, when performing subject verb agreement. This trend is stronger for RoBERTa-based models, while smaller BERT-based models pay less attention to the article of the subject, even when it agrees with the subject. These results are novel, but do agree with behavioral studies that find BERT is able to perform subject-verb agreement well. Other studies look at layer-wise pipeline of BERT, but their results do not clearly correspond to our finding, perhaps because of our functional approach.

Conclusion

In this thesis, we have used causal interventions in LLMs to shed light on how they process subject-verb agreement with respect to plurality, in English. Specifically, we expanded an existing question by demonstrating that LLMs encode the plurality of subject nouns not only in the noun itself, but in all tokens of the sentence. Moreover, they do so at all layers of the model. However, we show using causal interventions that while LLMs encode subject-noun plurality in all tokens of the sentence, they only use information that is encoded in tokens that agree in number with the subject noun. Thus, we conclude LLMs are behaving in a linguistically-justified way, using the subject-noun plurality information of only tokens whose surface form reflects somehow the subject noun's plurality. Finally, we arrive at these conclusions using methodologies that clearly distinguish between what encoded information probes learn to extract, and what encoded information LLMs use as they process language input.

This analysis furthers the development of a new approach to model interpretability, that brings together model internals and model behavior. While other analyses have shown that model internals capture subject plurality information [Klafka and Ettinger, 2020], or that models perform well on subject-verb agreement [Goldberg, 2019], ours is able to show not only what models predict on a subject-verb agreement task, but what these models rely on in order to generate their predictions. As the debate regarding LLMs and their language abilities rages on, this type of model analysis becomes increasingly important. By uncovering the connections between model internals and behavior and determining what underlying mechanisms control LLM output, we can work towards discovering if LLMs lack understanding, simply repeating things they have seen in training [Bender and Koller, 2020], or possess real generalization abilities [Bowman, 2021]. As language model abilities improve, the need to interpret NLP models continues to grow—and causal methods provide a promising starting point for doing so.

Bibliography

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Geoff Bacon and Terry Regier. Does bert agree? evaluating knowledge of structure dependence through agreement relations, 2019. URL https://arxiv.org/abs/1908.09892.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98/COLING '98, page 86–90, USA, 1998. Association for Computational Linguistics. doi: 10.3115/980845.980860. URL https://doi.org/10.3115/980845.980860.

Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA, October 2010. Association for Computational Linguistics. URL https://aclanthology.org/D10-1115.

Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219, 04 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00422. URL https://doi.org/10.1162/coli_a_00422.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1080. URL https://aclanthology.org/P17-1080.

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://aclanthology.org/2020.acl-main.463.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, mar 2003. ISSN 1532-4435.

Arianna Bisazza and Clara Tump. The lazy encoder: A fine-grained analysis of the role of morphology in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2871–2876, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1313. URL `https://aclanthology.org/D18-1313`.

Samuel R. Bowman. The dangers of underclaiming: Reasons for caution when reporting how nlp systems fail, 2021. URL `https://arxiv.org/abs/2110.08300`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Rui P. Chaves and Stephanie N. Richter. Look at that! BERT can be easily distracted from paying attention to morphosyntax. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 28–38, Online, February 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.scil-1.3`.

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`.

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 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://aclanthology.org/W19-4828.

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 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198.

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 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10.

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://aclanthology.org/2020.tacl-1.3.

Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. Assessing composition in sentence vector representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1790–1801, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1152.

William Falcon and The PyTorch Lightning Team. PyTorch Lightning, 3 2019. URL https://github.com/PyTorchLightning/pytorch-lightning.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1184. URL https://aclanthology.org/N15-1184.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

Alona Fyshe, Leila Wehbe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. A compositional and interpretable semantic space. In *Proceedings* *of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 32–41, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1004. URL `https://aclanthology.org/N15-1004`.

Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL `https://aclanthology.org/2020.blackboxnlp-1.16`.

Atticus Geiger, Hanson Lu, Thomas F Icard, and Christopher Potts. Causal abstractions of neural networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=RmuXDtjDhG`.

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`.

Yoav Goldberg. Assessing bert's syntactic abilities, 2019. URL `https://arxiv.org/abs/1901.05287`.

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://aclanthology.org/N18-2017`.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL `https://aclanthology.org/D19-1275`.

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://aclanthology.org/N19-1419`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. Do attention heads in bert track syntactic dependencies?, 2019. URL https://arxiv.org/abs/1911.12246.

Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016. doi: 10.1162/tacl_a_00101. URL https://aclanthology.org/Q16-1023.

Josef Klafka and Allyson Ettinger. Spying on your neighbors: Fine-grained probing of contextual embeddings for information about surrounding words. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4801–4811, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.434. URL https://aclanthology.org/2020.acl-main.434.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

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*, 104:211–240, 1997.

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 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-demo.21. URL https://aclanthology.org/2021.emnlp-demo.21.

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://aclanthology.org/W19-4825`.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1112. URL `https://aclanthology.org/N19-1112`.

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, 2019b. URL `https://arxiv.org/abs/1907.11692`.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. Probing across time: What does RoBERTa know and when? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 820–842, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.71. URL `https://aclanthology.org/2021.findings-emnlp.71`.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.645. URL `https://aclanthology.org/2020.acl-main.645`.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6297–6308, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

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 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL `https://aclanthology.org/P19-1334`.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *arXiv preprint arXiv:2202.05262*, 2022.

William Merrill, Ashish Sabharwal, and Noah A. Smith. Saturated transformers are constant-depth threshold circuits, 2021. URL `https://arxiv.org/abs/2106.16213`.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Honza Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf`.

Lalchand Pandia and Allyson Ettinger. Sorting through the noise: Testing robustness of information processing in pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1583–1596, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.119. URL `https://aclanthology.org/2021.emnlp-main.119`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `https://dl.acm.org/doi/10.5555/3454287.3455008`.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL `https://aclanthology.org/D14-1162`.

J.A. Perez-Ortiz and M.L. Forcada. Part-of-speech tagging with recurrent neural networks. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*, volume 3, pages 1588–1592 vol.3, 2001. doi: 10.1109/IJCNN.2001.938396.

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`.

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 2020. 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://aclanthology.org/S18-2023.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

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.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL https://aclanthology.org/2020.acl-main.647.

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.15. URL https://aclanthology.org/2021.conll-1.15.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.295. URL https://aclanthology.org/2021.eacl-main.295.

Jeff Reback, Wes McKinney, jbrockmendel, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, gfyoung, Sinhrks, Adam Klein, Matthew Roeschke, and et al. pandas-dev/pandas: Pandas 1.0.3. Mar 2020. doi: 10.5281/zenodo.3715232.

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.

Rudolf Rosa and David Mareček. Inducing syntactic trees from bert representations, 2019. URL https://arxiv.org/abs/1906.11511.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

D. E. Rumelhart and J. L. McClelland. *On Learning the Past Tenses of English Verbs*, page 216–271. MIT Press, Cambridge, MA, USA, 1986. ISBN 0262132184.

David E. Rumelhart and James L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. 1987.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

Hinrich Schütze. Word space. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992. URL `https://proceedings.neurips.cc/paper/1992/file/d86ea612dec96096c5e0fcc8dd42ab6d-Paper.pdf`.

Priyanka Sen and Amir Saffari. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.190. URL `https://aclanthology.org/2020.emnlp-main.190`.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL `https://aclanthology.org/P16-1162`.

Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1159. URL `https://aclanthology.org/D16-1159`.

Jakub Sido, Ondřej Pražák, Pavel Přibáň, Jan Pašek, Michal Seják, and Miloslav Konopík. Czert – Czech BERT-like model for language representation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1326–1338, Held Online, September 2021. INCOMA Ltd. URL `https://aclanthology.org/2021.ranlp-1.149`.

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 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.230. URL `https://aclanthology.org/2021.emnlp-main.230`.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/D12-1110.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

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 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=SJzSgnRcKX.

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. Von 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.

Oriol Vinyals and Quoc Le. A neural conversational model. *ICML Deep Learning Workshop, 2015*, 06 2015.

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.

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://aclanthology.org/W18-5446.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

Shuohang Wang and Jing Jiang. Learning natural language inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1170. URL https://aclanthology.org/N16-1170.

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 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1286. URL https://aclanthology.org/D19-1286.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York, January 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.scil-1.47.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 932–948, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.72. URL https://aclanthology.org/2021.emnlp-main.72.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022. URL https://arxiv.org/abs/2206.07682.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. *ArXiv*, abs/1909.10430, 2019.

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, October 2020. 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. URL `https://arxiv.org/abs/1609.08144`.

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://aclanthology.org/2020.acl-main.383`.

Mo Yu and Mark Dredze. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2089. URL `https://aclanthology.org/P14-2089`.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL `https://aclanthology.org/P19-1139`.

Yukun Zhu, Ryan Kiros, Richard S. 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. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, 2015.

List of Figures

List of Tables

List of Abbreviations

In (rough) order of appearance:

- **LLM**: Large Language Model
- **BERT**: Bidirectional Encoder Representations from Transformers (a popular LLM from Devlin et al. [2019])
- **NLP**: Natural Language Processing
- **FFN**: Feedforward Neural Network
- **RNN**: Recurrent Neural Network
- **SotA**: State of the Art
- **NLU**: Natural Language Understanding
- **NLI**: Natural Language Inference
- **POS**: Part Of Speech
- **MLM**: Masked Language Model

A. Appendix

A.1 Probe Training Details

Probes are implemented as PyTorch linear layers with a sigmoid activation function. Each probe took in inputs with size equal to the hidden size of the transformer that generated the representations taken as input. Each probe had an output dimension of 1. Once passed through the sigmoid function, outputs less than 0.5 corresponded to the singular class, and those greater than 0.5 corresponded to the plural class. Probes were trained using binary cross-entropy loss, and trained for 40 epochs using batch gradient descent (batch size = 16) and the Adam optimizer (learning rate = 0.001).