Probing Language Models for Contextual Scale Understanding

by

Saaketh Vedantam

S.B, Computer Science and Engineering and Mathematics,
Massachusetts Institute of Technology (2023)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Saaketh Vedantam
Department of Electrical Engineering and Computer Science
May 12, 2023

Certified by: Yoon Kim
Assistant Professor
Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Probing Language Models for Contextual Scale Understanding

by

Saaketh Vedantam

Submitted to the Department of Electrical Engineering and Computer Science
on May 12, 2023, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Pretrained language models (LMs) have demonstrated a remarkable ability to emit linguistic and factual knowledge in certain fields. Additionally, they seem to encode relational information about different concepts in a knowledge base. However, since they are trained solely on textual corpora, it is unclear whether these models implicitly understand anything grounded about the real world. This work investigates the extent to which LMs learn the structure of the physical world. By probing the contextualized embeddings of sentences, we examine how well LMs predict the sizes of real-world objects. We further explore the effect of adjectival modifiers on object embeddings. We show that while larger models more accurately convey scalar information through their embeddings, they perform on par with smaller models in the task of contextual prediction. Fortunately, the models are capable of identifying a difference in scale when an adjectival modifier is introduced, implying that the relevant context is successfully incorporated into the object's embedding through the LM's attention mechanism.

Thesis Supervisor: Yoon Kim
Title: Assistant Professor

Acknowledgments

I picked up this project in fall 2022 as a fan of the NLP field. I am happy to say I've gained experience, knowledge, and a lot more faith in my abilities over the course of this year. However, my abilities alone did not get me here. I deeply appreciate everyone who supported my efforts and made my time here so enriching.

First and foremost, I would like to express my immeasurable thanks to my thesis advisor, Yoon Kim, whose valuable insights, persistent guidance, and continuous support have helped me throughout this entire process. His availability and breadth of knowledge on the topic helped facilitate the research process and let me focus on the most interesting parts of the project. I feel like he has observed every step of my NLP journey, from teaching the course I took and TA'd to offering countless insights into my research.

I would like to thank my past instructors, and MIT as a whole, for providing such a productive learning environment in my undergraduate and graduate studies. I am glad I met so many interesting, thoughtful, and passionate people who pushed me over the years. I am grateful to have made friends who have laughed with me, challenged me, and positively influenced me time and again.

Finally, I am indebted to my family, who raised me to have a strong thirst for knowledge and the willpower to overcome obstacles. They have always supported me in my academic endeavors, and I could not have done any of this without their love and trust.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Neural language models have come a long way since their inception in the early 2000s. They transitioned from using simple word features to train a feed-forward network to learning pretrained word embeddings that are almost universal. The introduction of large language models (LLMs), such as GPT-4, has revolutionized the field of natural language processing (NLP) and artificial intelligence (AI) as a whole. As LLMs are only a few years old, their potential applications are still limitless, and there is much to be explored regarding their capabilities and limitations.

One common limitation expressed in literature is the ungrounded training of LLMs. It's true that LLMs are incredibly powerful, but without relevant experience in the real world, it's unclear how much an agent can actually understand about language [3]. If we want intelligent agents to be able to communicate with us, we need them to understand the physical and social context of language before generating it. Thus, it makes sense to determine the extent of the knowledge contained within LLMs.

This has led to a class of LLM training tasks called *probing* tasks. Probes are simple supervised models trained to predict a property of language from a linguistic representation. The idea is that if a word representation contains information about some linguistic property, then it shouldn't require much additional processing and data to predict that property. Probes have achieved success in tasks identifying morphology and part-of-speech [2], syntactic and semantic information [15], and various

tasks ranging from superficial phenomena to subtle meaning [7].

In this work, we will focus on using LMs to predict scalar attributes. [27] calls this task *scalar probing*. We will extend this to *contextualized scalar probing*. This will assess the LM's ability to predict scalar quantities based not only on concepts (nouns), but also on potential adjectival modifiers. In the next few sections, we will provide a concise summary of fundamental concepts related to this study and end with a formal exposition of the problem that we intend to address.

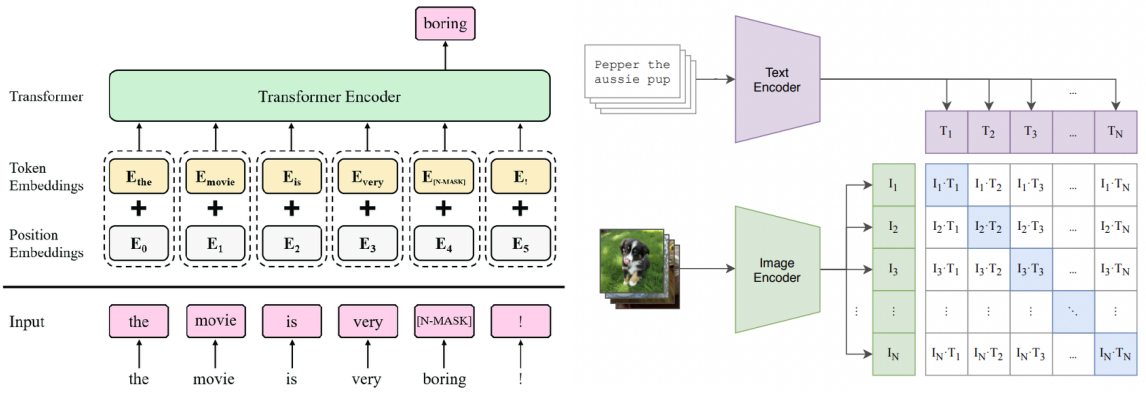
1.1 Large Language Models

LLMs are advanced neural networks that are trained on massive amounts of text data using unsupervised learning techniques. The training process involves feeding the model a large corpus of text, such as books, articles, and web pages, and optimizing its parameters to predict the likelihood of observing a given sequence of words. The primary architecture used for LLMs is the Transformer model [26], whose main building block is the self-attention layer. The attention mechanism successfully allows the model to learn useful contextual representations of text that capture long-term dependencies and result in greatly improved performance of various NLP benchmarks. There's evidence that the various attention heads learn different syntactic and semantic connections between words [6].

LLMs have achieved remarkable results on many NLP tasks, demonstrating the potential for these models to be used in various applications. However, the computational cost of training and deploying LLMs is still a major challenge, and there are ethical considerations surrounding the use of these models, including bias and fairness issues, that need to be carefully addressed. In this paper, we will focus on the capabilities of LLMs, specifically in the scope of perceptual understanding.

1.1.1 Text-Only Training

Many of the most successful LLMs have been trained purely on textual data. As shown in Figure 1-1a, these models employ a masked language modeling objective.



(a) Text-only masked language modeling. (b) Image-aware contrastive pre-training.

Figure 1-1: Approaches to learning contextual text representations.

Here, part of the input is hidden, and the model must learn the next word or sentence from the context its given. Through this procedure, the model learns rich representations of the words, informed by the context surrounding them. This text-only nature of LLM training has both advantages and limitations.

One advantage of text-only training is that it allows the models to learn from vast amounts of diverse and unstructured data. This enables the models to learn the complexity of natural language. Furthermore, the models can be easily adapted to new languages and domains by simply adding more data to the training corpus.

However, they may not be able to capture some aspects of human knowledge and experience that are no present in the text itself. This can lead to biases and limitations in the models' understanding of the world, as they may lack certain cultural or social context that is crucial for understanding human language. Additionally, the models may struggle with tasks that require more than just text understanding, such as image or video captioning, where visual cues and context play an important role.

So far, for such a simple training objective, LLMs have proven to be capable of several unexpected tasks. These include code generation and documentation [5], few-shot planning [22], and long-form conversation (ChatGPT). We ask if this can be extended beyond textual tasks to the grounded world, without introducing specific real-world data. If so, this would provide yet another insight into the importance of language as a tool for communication.

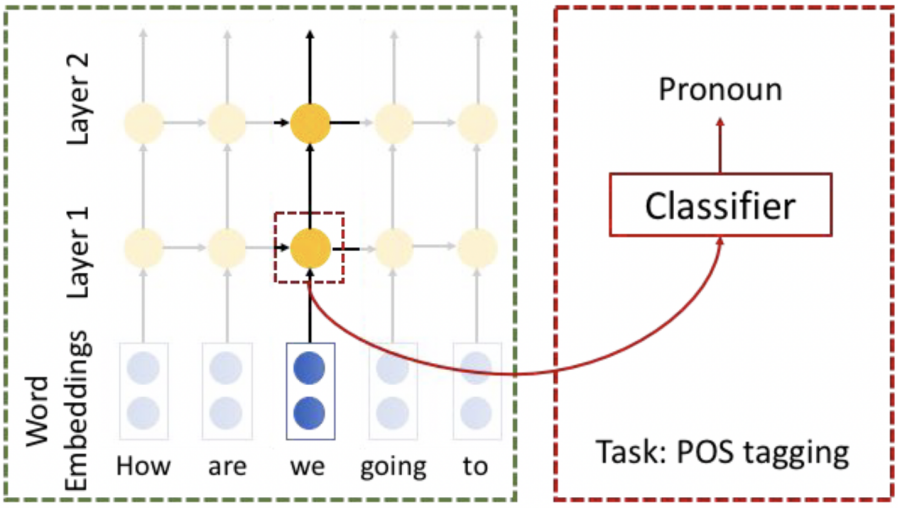


Figure 1-2: An example of probing for POS tagging. The word embedding is extracted and run through a classifier to solve some downstream task.

1.1.2 Image-Aware Training

One way to ground language models is to introduce images into the training procedure, thereby learning multimodal representations. These vision-language pre-training (VLP) methods have achieved success in such tasks as image captioning, visual question-answering, and image-text retrieval [11]. Two such cutting-edge models are CLIP [16] and GPT-4 [13]. Both models jointly learn aligned representations of images and text (Figure 1-1b). It's possible that because of this, the models are better at perceptual understanding tasks than the text-only models. We will explore this later on.

1.2 Probing

Probing is a technique used to understand the internal workings of language models. It involves training a simple linear classifier on top of the hidden representations of a language model, with the goal of predicting a linguistic property or feature of the input text (Figure 1-2). By probing the hidden representations in this way, we can gain insights into what aspects of language the model is capable of capturing, and how it processes and represents linguistic information. Probing has been used to

investigate a wide range of linguistic phenomena, including syntax, semantics, and pragmatics. For example, probes have been used to study whether language models can understand identify temporal properties of expressions [24].

One advantage of probing is that it allows us to gain insights into the strengths and weaknesses of language models without requiring large amounts of annotated data or complex task-specific architectures. Probing can be applied to any pre-trained language model, regardless of its architecture or objective, and can be used to compare different models or configurations in a systematic and controlled way. In a way, the output of probes can be used to interpret the black-box structure of these deep models.

However, we also have to be wary of the limitations of probing. For example, the linear classifiers used in probing may not capture the full complexity of the linguistic properties being studied, and may be prone to overfitting. If the probe is too complex, then it might be able to learn the relation regardless of the input embedding, and the result will not tell us anything useful about the underlying language model. Still, it's a valuable technique for investigating the abilities of LLMs, and we will employ them in our scalar prediction task.

1.3 Thesis Outline

In chapter 2, we will continue discussing works that are relevant to this research problem. Chapter 3 describes the problem statement in more detail, as well as the methods and models that will be used to explore it. Chapter 4 talks about the training and evaluation dataset used in our studies. Chapter 5 goes into the experiments conducted, metrics collects, and results. Chapter 6 includes an analysis of these results. Finally, chapter 7 concludes with the broader implications of this work.

Chapter 2

Related Work

2.1 Transformers

The Transformer is a type of neural network architecture that has revolutionized NLP in recent years. It has become a dominant paradigm in language understanding, outperforming previous state-of-the-art models on a wide range of tasks [26].

At a high level, transformers consist of an encoder and a decoder, each of which is composed of multiple layers of self-attention and feedforward neural networks. Self-attention allows the model to selectively focus on different parts of the input sequence, while the feedforward networks provide non-linear transformations of the input. The main draw of this approach is that the attention layers help develop contextualized representations for each word in a phrase, and everything is parallelizable, unlike previous iterations of RNNs.

Some popular Transformer-based models include BERT [8], the GPT series [17], and the T5 model [19]. These are all examples of pre-trained models, where the actual training occurs in a self-supervised fashion. By pre-training on a data-rich environment, the goal is to learn general-purpose knowledge that can then be transferred to downstream tasks. One important, and rather surprising, insight was that LLMs trained on Transformers didn't need to be finetuned on large amounts of task-specific data in order to complete a task. Instead, if they were big enough, they could learn through few-shot prompting [4]. This opened up the possibilities for exploring the

scope of LLMs beyond linguistic tasks, to more general intelligence. Here, we explore the role of language in communicating perceptual features.

2.2 Scalar Probing

Previous works have studied mappings from LMs to sizes, colors, and physical directions. Through these studies, there seems to be evidence that text-only models encode interesting relational information about the concepts they model.

[27] introduces scalar probing as a prerequisite task for common sense reasoning. Scalar probing involves predicting some numeric attribute of an object, such as its mass or length. A probe is a very simple model, for example a linear regression model, that maps an LM representation to the scalar target value. This turns the supervised task of predicting the probed value into a technique for interpreting the richness of the underlying embedding. According to [12], linear probes achieve a high selectivity compared to non-linear ones, making them a better choice for our experiments. [27] found success in probing their model for such scalar information.

Additionally, this paper designed NumBERT, a model which processes numbers in scientific notation. This improved how the model processed magnitudes, which turned out to be important in the scalar probing task.

2.3 Perceptual Alignment

A closely related problem is that of aligning text-only representations with their grounded counterparts. [1] found a correspondence between the BERT embeddings of color terms and CIELAB, a color space where distance relates to color similarity. Further, [14] showed that when prompted with certain perceptual information (e.g. northeast and its direction on a compass), GPT-3 could predict the corresponding perceptual property of a related word.

These experiments provide evidence that LMs represent relations between textual concepts in a structure similar to what a grounded model learns.

Chapter 3

Methodology

At a high level, this project will consist of two steps: training a probe model and evaluating it. We will use a LM to generate representations of common nouns and train a probe to predict some physical attribute of the noun, such as the mass. We can validate these probes to see how well their scale attribute prediction actually generalizes. This will include data collection and preprocessing, model generation and training, and evaluation on handcrafted tasks.

Extending previous works that probed nouns for their attributes [27], we explore how adjectival modifier embeddings affect these predicted quantities. For example, we test if a language model understands that 'small cat' or 'baby cat' will have a smaller mass than 'cat'. These adjectives serve as contextual modifiers that we as humans understand as adjustments to our prior of the object's features, but it would be very interesting to see how the language model treats them.

3.1 Language Models

The first step involves training a probe model on top of language model representations. We test different language models to assess how their size and training task affects the representations learned. The overall collection of language models and their properties are listed in Table 3.1.

Model	Transformer Arch.	Training	# Params
BERT-Base	Encoder	Text	110M
BERT-Large	Encoder	Text	340M
CLIP	Encoder	Text and Images	63M
GPT-2 Medium	Decoder	Text	345M
GPT-2 Large	Decoder	Test	774M

Table 3.1: Language models used to generate representations.

3.1.1 BERT

The BERT (Bidirectional Encoder Representations from Transformers) model [8] is a state-of-the-art language model. The model is trained using a masked language modeling approach, whereby the model is presented with a sequence of words or sentences with some of the words randomly masked out, and it is required to predict the masked words based on the surrounding context. The model is also trained using a next sentence prediction task, whereby the model is presented with pairs of sentences and it is required to predict whether the second sentence follows logically from the first. For the first task, the input needs context from before and after the masked word, hence the bidirectional nature of BERT. Because this is a simple classification task, BERT only uses a Transformer encoder. For the second task, an additional `[CLS]` token is inserted at the front of a sentence as an attempt to represent the sentence embedding. This token is often used for other downstream tasks involving sentence classification. However, recent works have indicated that different word-level concatenations do a better job of document classification [23].

BERT comes in various sizes, with the number of parameters ranging from 110 million to over 340 million. The smaller models are faster and more efficient but may not perform as well as the larger models on complex NLP tasks. The largest model, known as BERT Large, has 345 million parameters and requires a significant amount of computational resources and memory to train and use. However, it has achieved state-of-the-art performance on many NLP benchmarks.

We will use the uncased BERT implementations, meaning they are trained on lower-case English text. The two main models are `bert-base` and `bert-large`.

`bert-base` has a hidden size of 768, 12 attention heads, and 12 layers. `bert-large` has a hidden size of 1024, 16 attention heads, and 24 layers. Because everything about these models, except their size, is the same, we expect `bert-large` to perform better because its representations are more expressive.

3.1.2 CLIP

The CLIP (Contrastive Language-Image Pre-Training) model [16] is a state-of-the-art AI model developed by OpenAI. It is a zero-shot model that can efficiently learn visual concepts from natural language supervision. The model is trained on a massive dataset of 400 million pairs of images and text.

CLIP has two sub-models, called encoders, including a text encoder and an image encoder. The text encoder is responsible for embedding text into a high-dimensional vector space, while the image encoder is responsible for converting an image into a latent vector representation.

One of the key strengths of CLIP comes from its use of a contrastive objective function, which enables it to learn representations that are highly discriminative. This objective function requires that the model learns to distinguish between pairs of (image, text) samples that are related versus those that are unrelated. It does this by aligning the embeddings of text that is a caption of an image and decreasing the cosine similarity between those that aren't matched.

CLIP's ability to connect text and images allows it to perform a wide range of tasks, including image classification, semantic segmentation, and object detection. Unlike other image classification models that are trained solely on images, CLIP is trained on both images and text, making it highly versatile and capable of understanding the context and relationships between different objects in an image.

We only use the text encoder of CLIP. This has a hidden size of 512, 8 attention heads, and 12 layers.

3.1.3 GPT-2

GPT-2 [18] is yet another pre-trained language model, but it is different from BERT in several key ways, including architecture, training methods, and intended use cases. GPT-2 is based on a Transformer decoder architecture, instead of BERT's encoder base. For this reason, its attention is only unidirectional. GPT-2 is also trained only using the next-word prediction task. The difference means that GPT-2 is specifically suited for generating text, while BERT is more suited for understanding text.

One of the key strengths of GPT-2 is its ability to generate highly coherent and human-like text, based on the context and prompt provided. The model is trained to predict the most likely next word in a given context, but it can also generate entire paragraphs of text based on a starting prompt or topic. However, this might also mean that it is less suited for probing tasks, because it only learns to generate fluent language, not necessarily understand it. GPT-2 is less domain-specific than BERT, meaning that BERT is more suited for tasks like sentiment analysis and text classification.

GPT-2 comes in various sizes, ranging from 124 million to over 1.5 billion parameters, with larger models having greater capacity and better performance on complex NLP tasks. The largest model, known as GPT-2 XL, has 1.5 billion parameters and can generate highly coherent and human-like text that is difficult to distinguish from text written by humans. Therefore, the GPT-2 model sizes are quite a bit larger than those of the BERT models.

We will use `gpt2-medium` and `gpt2-large`. `gpt2-medium` has a hidden size of 1024, 16 attention heads, and 24 layers. `gpt2-large` has a hidden size of 1280, 20 attention heads, and 36 layers. The medium model has roughly the same architecture size as `bert-large`, so it would be interesting to see how the encoder and decoder compare against each other for the task of scalar probing.

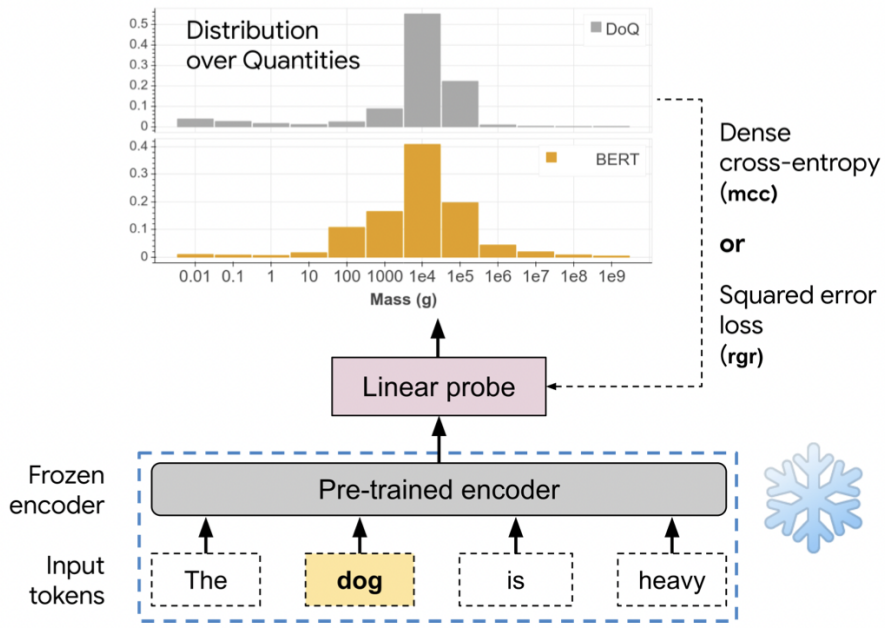


Figure 3-1: [27, Figure 1]: The probing architecture. The weights for the pre-trained language model are frozen and used to generate word embeddings. This passes through the probe and generates an output distribution.

3.2 Probe Model

With the embeddings from the LMs above, we then train a linear probe model. The representations are given by the first embedding layer in the Transformer.

3.2.1 Tasks

The target is a distribution over the size of an object. To this end, we specify two tasks: multi-class classification (**mcc**) and linear regression (**rgr**). These are taken from [27].

mcc splits the output distribution into 111 logarithmically-spaced buckets, from 10^{-2} to 10^{9}. [25] shows that a similar approach works well for modeling image pixel values. Because the different buckets are treated as separate classes, the relationship between adjacent buckets is ignored, but it allows us to label the full empirical distribution in a non-parametric manner.

rgr is trained to predict the log of the median value of the distribution. This

point estimate might be easier to compute, but it might not be enough to understand the full scale properties of an object.

3.2.2 Training

We use the training procedure from [27]. Specifically, for **mcc**, we use a linear classifier with a softmax activation function and regularization strength of 0.01. For **rgr**, we use ridge regression with a regularization strength of 1. The evaluation will be described in chapter 5.

3.3 Adding Context

Once the probe model is trained and evaluated, we will apply it to the novel task of contextual scalar probing. In this task, the object will be modified by some scalar adjective, like 'short' or 'large'. We then run the object's embedding through the probe (in evaluation mode) to get the predicted scalar attribute. This will be compared against the base size predicted by the probe. If the attention mechanism is successfully able to incorporate information from the modifier into the object's contextual embedding, then we would expect to see a change in the right direction.

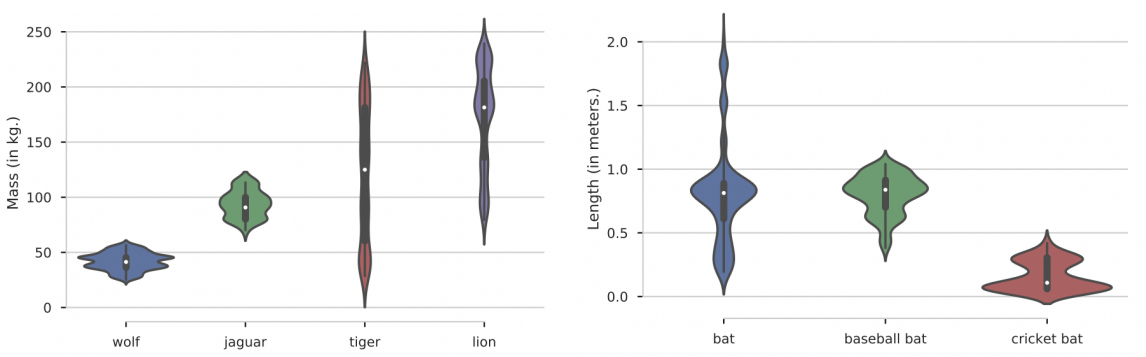
Chapter 4

Datasets

4.1 Distributions over Quantities

To obtain our ground truth information, we will use the Distributions over Quantities (DoQ) dataset [9]. DoQ provides the empirical data on several scalar attributes of objects. It contains data on over 300K nouns across 10 scalar attributes, including mass, speed, length, and price. Since the true size of an object is inherently uncertain, DoQ offers a distribution over the quantity for each noun. This is represented by a set of counts, which can then be bucketed into a histogram or modeled as some distribution. Figure 4-1 shows some examples of generated mass distributions.

In this work, we will primarily look at the MASS and LENGTH attributes of objects. Size is a salient perceptual feature of any object that isn't accounted for in text-only models, so it would be interesting if LMs could infer the size simply based on their understanding of linguistic structures. As a disclaimer, DoQ is scraped from web data and itself is noisy. However, it is a very large resource, containing over 100 million nouns, and it is a good starting point for collecting information on absolute quantities.



(a) Example mass distributions.

(b) Example length distributions.

Figure 4-1: Some objects and their corresponding scalar attribute distributions according to DoQ. Note that in (b), words can have multiple senses according to context and this is accounted for by having different values.

4.2 Train Dataset

The training step occurs when we want to build the probe model off the LM representations.

4.2.1 Preprocessing

The raw DoQ dataset is very large, containing almost 7GB of data. We first filter by the relevant dimensions to get a dataset of MASS and LENGTH. As mentioned above, these data points are noisy estimated collected heuristically from an unsupervised scraping scheme. To reduce variance, we only use objects with more than 100 values collected. To create the distributions, we first take the $\log_{10}$ of all values in the dataset.

For **mcc**, we model the distribution as a categorical distribution over 111 values, ranging from -2 to 9. We chose this range because it captures the scale of every object in the dataset. We chose 111 because there are 12 integer buckets in this range. Empirically, 12 buckets might be too coarse, as one bucket would represent an entire order of magnitude (e.g. 100-1000 kg). Therefore, many objects' distributions would end up looking very spiky. Splitting this 10 times, from -2.0 to 9.0 in 0.1 increments, helps mitigate that effect by distribution the object's scalar attributes more finely.

We then take the normalized counts in each bucket as the true distribution for the scalar value of that object.

For **rgr**, we calculate the median of the empirical log-values. We choose the median because it's a more robust measure of the center of the distribution than the mean. Additionally, some distributions are multimodal, due to word sense ambiguity. We do not explore the effect of different modalities in this work, but using the median helps pick a value that is more represented in the data.

After preprocessing, the MASS subset contains 74,102 objects and the LENGTH subset contains 240,068 objects. During training, we employ 10-fold cross validation and average the results. We then evaluate on our novel task of contextual scale prediction, using the full dataset.

Chapter 5

Experiments

5.1 Probe Training

In this section, we detail the experiments done to train the probe model. This includes how we determined the inputs, outputs, and evaluation metrics.

5.1.1 Input Representations

As our transformer LMs are contextual text encoders operating on full sentences, we generate artificial sentences with the following templates. Here, `X` is a stand-in for some noun.

- MASS: How heavy is the `X`?
- LENGTH: How big is the `X`?

Other works have used the `[CLS]` token embedding for BERT or the final token/`[EOS]` embedding for GPT-2. However, we want to examine the contextual capabilities of the Transformer's attention mechanism, so we average the embeddings of each token in the object's name. As mentioned in [27], LENGTH measurements in DoQ can refer to length, width, or height, so we use the all-encompassing adjective 'big' here for our representation template. We could've also gone more literal; e.g. "What is the mass of X?" or "What is the length of X?". These variations don't seem

to affect the performance of the probe, so we focus on the templates listed above. Additionally, we experimented with the more neutral prompt "This is the `X`." Again, the results were somewhat similar to those of our original prompts.

5.1.2 Evaluation Metrics

As mentioned in section 3.2, we have two training tasks: **mcc** and **rgr**. We calculate three performance metrics depending on which probe we trained. For **mcc**, we calculate accuracy and EMD. For **rgr**, we calculate MSE.

Accuracy This measures the similarity of the predicted values to the ground-truth mode. For **mcc**, this means calculating the highest scoring class from the predicted distribution and comparing it to the highest scoring class from the empirical distribution. We choose the mode instead of the median because many distributions predicted by the probe model have a spike. In the ground truth distributions, the median and mode are often in the same, or very close, buckets, because most objects have somewhat symmetric scalar distributions and are unimodal.

As mentioned before, the buckets are quite fine-grained, so exactly matching to a bucket would yield a very low accuracy. Instead of matching to the exact ground truth bucket defined above, we match to any of the 5 nearest buckets (2 to the right and 2 to the left).

Mean Squared Error (MSE) Let p and $\hat{p}$ be the ground truth and predicted scalar values, respectively. Then the squared error here is $(p - \hat{p})^2$, and the MSE across the validation dataset is the average of these errors across all objects. To get the ground truth estimate, we use the median of the distribution.

Earth Mover's Distance (EMD) This is also known as the Wasserstein distance [21]. Let p and $\hat{p}$ be the ground truth and predicted distributions, respectively. Then the EMD is defined as

$$D(p, \hat{p}) = \inf_{\pi} \mathbb{E}_{(x,y)\sim\pi}[d(x, y)],$$

where π is any joint distribution whose marginals are p and $\hat{p}$ and d is a distance metric over the probability space. For example, since we're working with positive real

numbers as the scalar magnitudes, we can use absolute difference: $d(x, y) = |x - y|$. For this special case of distance function, the EMD is also equal to

$$D(p, \hat{p}) = \int_{-\infty}^{\infty} |P(x) - \hat{P}(x)| \, \mathrm{d}x,$$

where P and $\hat{P}$ are the cumulative density functions of p and $\hat{p}$, respectively [20].

EMD is a special case of optimal transport. Intuitively, it accounts for how far density would need to move from one distribution to the other to equalize the distributions. For example, two distributions with far apart modes would have a larger EMD than two with modes next to each other. This is potentially useful to calculate the error in the **mcc** probes, since we want to penalize wildly incorrect predictions more than small shifts.

5.1.3 Baseline

For this problem, we do not have a previous baseline to compare to. Therefore, we construct a version of the majority baseline, or ZeroR, used in classification tasks. We aggregate the buckets of every object in our dataset, and return the empirical distribution over the dataset. This **aggregate** baseline is evaluated for every object in the dataset and the metrics are calculated. We then compare our fully trained models to this baseline to get a sense of how much better than average they are.

5.1.4 Results

The results for the scalar probe training experiments are shown in Table 5.1.

5.2 Contextual Evaluation

In this section, we detail the experiments conducted to evaluate the contextual probing abilities of our models.

		Accuracy (mcc)	MSE (rgr)	EMD (mcc)
Lengths	Aggregate	0.25	0.070	0.079
	CLIP	0.35	0.091	0.050
	BERT-Base	0.34	0.101	0.053
	BERT-L	0.38	0.086	0.052
	GPT2-M	0.36	0.090	0.056
	GPT2-L	**0.41**	0.083	**0.046**
Masses	Aggregate	0.12	0.075	0.090
	CLIP	0.32	0.085	0.061
	BERT-Base	0.28	0.087	0.071
	BERT-L	0.34	0.085	0.057
	GPT2-M	0.31	0.086	0.057
	GPT2-L	**0.38**	0.084	**0.052**

Table 5.1: Results for the scalar probe experiments. The largest model consistently outperforms the others.

5.2.1 Input Representations

To evaluate our probes, we again need to generate input representations. The inputs to our language models are artificial sentences according to the following templates:

- MASS: How heavy is the *large/small* `X`?
- LENGTH: How big is the *long/short* `X`?

We do not change the overall structure of the sentence, so as to minimize confounding. However, we do add a relevant adjective in front of the noun `X`. Again, we only pass the average embedding of the tokens `X` as the input to the language model. The goal is to test if the information from the adjectives gets transferred to the noun through the LMs' attention mechanism.

5.2.2 Evaluation Metrics

In all of our evaluations, we compare the output given by `[adj] X` with that of `X`. Therefore, we're only comparing the model's predictions, not the ground truth distribution.

% change To evaluate the effect of our adjectival modifiers, we calculate the percentage change in the predicted sizes. If we let x^* be the modified prediction and x be the original prediction, the formula is given as follows:

$$\Delta(x^*, x) = 100 \cdot \frac{x^* - x}{x} \approx 100(\ln x^* - \ln x) = 100 \ln 10 \cdot (\log_{10} x^* - \log_{10} x),$$

where we use the approximation because we predicted log scale in our probes. For **rgr**, we compare the predicted values. For **mcc**, we compare the medians of the predicted distributions. The median is calculated by finding the median bucket and interpolating to estimate the 50th percentile.

One-Way EMD Similar to our evaluation of the probe model, we use EMD to compare the original distribution to the modified distribution. This is only for the **mcc** probe. Since we want to measure the direction of change as well as the magnitude of change, we slightly modify the metric:

$$D(p^*, p) = \int_{-\infty}^{\infty} P(x) - P^*(x)\,\mathrm{d}x,$$

where p^*, p are the modified and original distributions, respectively, and P^*, P are their respective cumulative distributions. By removing the absolute value, we measure the extent to which p^* is to the right of p.

5.2.3 Results

The percent change results are shown in Table 5.2. As we do not have a baseline for this task, we report standard errors for all the tasks to show a significant deviation from 0. These are calculated by aggregating the percent change results across all objects in the dataset.

The results using the EMD metric are shown in Table A.1.

		% change			
		large	small	long	short
mcc	CLIP	$+52.4 \pm 12.7$	-23.5 ± 5.9	$\mathbf{+57.4 \pm 13.1}$	$\mathbf{-25.8 \pm 7.6}$
	BERT-Base	$+51.4 \pm 10.8$	-22.0 ± 6.3	$+56.4 \pm 10.1$	-24.4 ± 6.9
	BERT-L	$+53.2 \pm 9.6$	$\mathbf{-25.8 \pm 6.8}$	$+55.8 \pm 10.8$	-24.6 ± 7.0
	GPT2-M	$+50.6 \pm 11.4$	-23.7 ± 5.8	$+56.0 \pm 11.3$	-23.7 ± 6.5
	GPT2-L	$\mathbf{+54.8 \pm 10.9}$	-24.6 ± 6.6	$+57.0 \pm 12.1$	-23.8 ± 7.4
rgr	CLIP	$\mathbf{+30.6 \pm 8.3}$	$\mathbf{-19.7 \pm 5.4}$	$\mathbf{+40.0 \pm 11.6}$	$\mathbf{-17.3 \pm 4.9}$
	BERT-Base	$+26.2 \pm 9.2$	-13.0 ± 5.5	$+36.8 \pm 10.3$	-14.4 ± 4.6
	BERT-L	$+25.8 \pm 8.7$	-13.5 ± 4.8	$+37.2 \pm 9.9$	-14.3 ± 5.0
	GPT2-M	$+25.4 \pm 9.3$	-13.4 ± 6.2	$+37.2 \pm 12.2$	-13.9 ± 5.9
	GPT2-L	$+26.8 \pm 8.9$	-13.8 ± 6.0	$+37.9 \pm 11.1$	-14.5 ± 5.4

Table 5.2: Metrics for the contextual probing experiments. Results are averaged over all objects in the dataset.

Chapter 6

Discussion

In this section, we analyze the results of the experiments in chapter 5.

6.1 Probe Models

First, we offer an interpretation of the values in Table 5.1. Better results are indicated by smaller MSE and EMD, with the minimum at 0. MSE is upper bounded by 1, while EMD could theoretically go as high as the range of the scalar attribute. Accuracy is better at higher values, and could theoretically go up to 1.

The data show an interesting relationship between the size of the models, their training details, and their performance. Within the BERT and GPT-2 subclasses, the larger model consistently outperforms the smaller one. This implies that model expressivity is important in generating representations that can be used for scale understanding. One interesting observation is that GPT2-M performs slightly worse that BERT-L. The difference is small, meaning it could be due to noise. However, it is consistent across evaluation metrics. Since these models have the same number of parameters, this could imply that the training style of GPT-2 makes it less suitable for scale understanding tasks than BERT. One reason for this could be that BERT is trained for many linguistic understanding and inference tasks, while GPT-2 is primarily a text generation model. Nevertheless, GPT2-L outperforms all, indicating that model size is still the biggest gauge for representation richness.

When put into context with its size, CLIP's performace is surprising. It is the smallest text encoder model represented, but it performs on par with GPT2-M, and in between BERT-Base and BERT-L. This indicates that its image aware training has introduced some scale understanding into its text embeddings, without needing the raw expressive power of the more state-of-the-art models.

Finally, we see that probes trained on the **mcc** objective consistently outperform probes trained on the **rgr** objective. In fact, the **rgr** probes do not always seem able to beat our aggregate baseline. This may mean that having information about the full distribution is more useful than solely regressing to the median. Even with these less promising results, we still evaluate the **rgr** probes on our contextual task.

6.2 Contextual Results

Table 5.2 displays various notable results. In general, the probes are able to produce significant differences for scalar attributes in the right direction. `large` and `long` percent increases, and `small` and `short` show percent decreases. The magnitude of the decrease for the diminutive adjectives is smaller than the magnitude of increase for the augmentative adjectives. One possible reason for this is that the decrease is bounded by 100%, and often much smaller due to the minimum size of objects. On the other hand, large objects don't usually have a tight upper bound on their size. Rather, they can be made larger until their existence becomes infeasible. The results for MASS adjectives are similar to those of LENGTH adjectives, implying that either the underlying models do not grasp the difference between dimensions or they are similar enough scalar attributes that their descriptors are also similar.

For the **mcc** probe, the results for the augmentative adjectives lie around the 50-57% mark and the results for the diminutive adjectives lie around the 22-26% mark. This occurs for all language models, and it doesn't seem like there's any special choice of LM that consistently works better here. One reason for this might be due to our choice of buckets. Our logarithmic scale, which is 10^{12} wide, is split into 120 buckets, meaning each bucket is a factor of $10^{0.1} \approx 1.259$ apart. Similarly, a distance of two

buckets corresponds to a factor of $10^{0.2} \approx 1.585$. Since we take the median bucket and interpolate for **mcc**, a large part of the difference is given by the difference in buckets. Therefore, it might be the case that `large` and `long` shift the distribution on average by 2 buckets, while `small` and `short` only shift it by 1. For the diminutive case, we also check that $10^{-0.1} \approx 0.794$, which corresponds to a 20.6% decrease.

For the **rgr** probe, the results are much different. The magnitude of change is smaller than that of the **mcc** probe, but the relative change between the adjectives is around the same. Additionally, we see a clear winner in our choice of LM. Although the standard errors are high, CLIP on average outperforms the other models for all adjective tasks. This provides evidence that its image aware training lends it a significant advantage when comparing relative sizes. One could hypothesize that this occurs because the images trained alongside `large X` show bigger objects than the ones for `X`, and CLIP can capture that insight. As for the rest of the LMs, it doesn't look like model capacity influences their performance in the contextual scalar probing task. All four text-only models perform on par with each other, implying that their relative scale understanding is unaffected by their size.

Still, it's very impressive that they can transfer the signals in the adjectives to the object embeddings in a meaningful way. One reason for this might be that they were trained on text data with numerical information about the size of objects, with and without extra context. Since there's evidence that language models can infer relative physical knowledge about objects [10], it makes sense that they can leverage this alongside grounded absolute knowledge to generalize to new descriptions.

As an example, Figure 6-1 shows the distinction when "large dog" is fed in vs. just "dog". It's quite hard to see in the image, but the median line is actually two buckets to the right in the `large` case. This reflects a roughly $10^{0.2} - 1 \approx 58\%$ increase in predicted mass. With median interpolation, the actual percent increase is around 48%. However, this is still significantly greater than 0, showing that some scalar signal is being passed from the adjective to the noun.

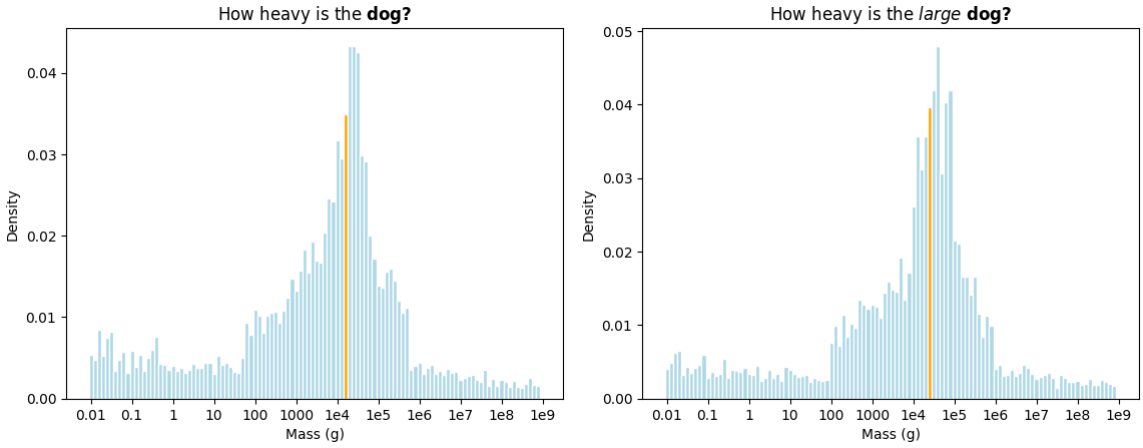


Figure 6-1: An example of the **mcc** MASS probe being run on the word 'dog', with and without the context `large`. The orange line represents the median bucket.

Chapter 7

Conclusion

We have extended upon the work of [27] to examine the perceptual understanding capabilities of LLMs given contextualized sentences. We trained scalar probe models to predict objects' masses and lengths from their symbolic representations in language models. These probe results show some scalar understanding, with larger, more expressive, models better able to capture perceptual signals. Once we've aligned the LM embeddings using a linear model, we saw whether adding a relevant adjectival modifier significantly changes the embedding in the right direction. These novel contextual experiments showed that while LMs can transfer relative scalar information, our current process of adding model complexity doesn't improve it any further.

As language models gain the ability to generate fluent text, respond to questions with relevant answers, and learn patterns through few-shot learning, we continue to search for more indicators of artificial general intelligence. One example of the is commonsense reasoning (CSR). CSR is the ability of a machine or an agent to understand and reason about everyday situations that are commonly understood by humans but are not explicitly stated in text or speech. CSR involves understanding concepts, relationships, and expectations that are shared among humans and are often not explicitly mentioned in language.

CSR is important because language is inherently ambiguous and context dependent. Humans rely heavily on their commonsense knowledge to understand the intended meaning of a sentence, even when it is not explicitly stated. However, ma-

chines lack this kind of knowledge, and therefore, they often struggle to comprehend the meaning of text and to generate relevant text that's not exactly related to its input.

In this work, we focused on a subtask in CSR, namely numerical reasoning. In an agent is to be deployed in the real world, it not only needs to understand relative scale information, such as "The lion is bigger than the house cat" or "The car is faster than the bicycle," but also absolute information about these measures. Nowadays, LLMs are being connected to the Internet and continually updated with the latest information, so they can retrieve absolute sizes. However, they still need the underlying common sense to be able to process the numerical data and apply it in context.

7.1 Future Work

Our work is limited in the scope of perceptual measures we use and the extent we test them. We only look at masses and lengths, while there are other perceptual features, like color, which could be interesting to examine. Future work could train more probe models, or define a generalizable probe model that also takes in the dimension of interest as input.

Additionally, we only use two simple adjectives in our experiments. This provides a proof of concept for how linguistic embeddings are affected. However, we could also vary the strength of these adjectives, for example by adding `"humongous"` or `"tiny"`. There are also a lot more subtle adjectives that could be used to change our understanding of an object's size. For example, for an animal, we know `"baby X"` is much smaller than `"X"`. Objects can be described in different classes, such as dog breeds. `"German shepherd"` is larger than `"chihuahua"`. Future works could more broadly study the effect of different classes.

Appendix A

Tables

	One-way EMD			
	`large`	`small`	`long`	`short`
CLIP	$+.19 \pm .06$	$-.12 \pm .05$	$+.23 \pm .08$	$-.09 \pm .04$
BERT-Base	$+.21 \pm .05$	$-.10 \pm .04$	$+.22 \pm .07$	$-.10 \pm .05$
BERT-L	$+.19 \pm .07$	$-.11 \pm .05$	$+.22 \pm .06$	$-.11 \pm .05$
GPT2-M	$+.18 \pm .06$	$-.11 \pm .04$	$+.19 \pm .06$	$-.10 \pm .04$
GPT2-L	$+.20 \pm .06$	$-.12 \pm .03$	$+.21 \pm .07$	$-.12 \pm .04$

Table A.1: Average EMD for the **mcc** probe across all objects in the dataset.

Bibliography

[1] Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. Can language models encode perceptual structure without grounding? a case study in color, 2021.

[2] 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.

[3] Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. Experience grounds language, 2020.

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

[5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

[6] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at? an analysis of bert's attention, 2019.

[7] 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.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[9] Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. How large are lions? inducing distributions over quantitative attributes, 2019.

[10] Maxwell Forbes and Yejin Choi. Verb physics: Relative physical knowledge of actions and objects, 2017.

[11] Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, and Jianfeng Gao. Vision-language pre-training: Basics, recent advances, and future trends, 2022.

[12] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks, 2019.

[13] OpenAI. Gpt-4 technical report, 2023.

[14] Roma Patel and Ellie Pavlick. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*, 2022.

[15] Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[17] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[18] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[19] 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, 2020.

[20] Aaditya Ramdas, Nicolas Garcia, and Marco Cuturi. On wasserstein two sample testing and related families of nonparametric tests, 2015.

[21] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66, 1998.

[22] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models, 2023.

[23] Hirotaka Tanaka, Hiroyuki Shinnou, Rui Cao, Jing Bai, and Wen Ma. Document classification by word embeddings of bert. In Le-Minh Nguyen, Xuan-Hieu Phan, Kôiti Hasida, and Satoshi Tojo, editors, *Computational Linguistics*, pages 145–154, Singapore, 2020. Springer Singapore.

[24] Shivin Thukral, Kunal Kukreja, and Christian Kavouras. Probing language models for understanding of temporal expressions, 2021.

[25] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks, 2016.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[27] Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. Do language embeddings capture scales?, 2020.