Control Tasks to Test Probes of Implicit Meaning

Chandler Ault
Princeton University
cjault@princeton.edu

Will Drury
Princeton University
wdrury@princeton.edu

Abstract

It has already been shown that neural language models (NLMs) implicitly represent syntactic structure, lexical relations, and other linguistic properties; however, one key question is whether or not these models encapsulate meaning (relationships between objects, consequences of actions, etc.) We investigate previous work by (Li et al., 2021) which found that dynamic representation of meaning and implicit simulation of behavior contributed to accurate prediction of final entity states. Building off (Hewitt and Liang, 2019) we design two control experiments which validate prediction accuracy by examining the capacity for probes to memorize tasks independently of underlying language model representations. Ultimately, we find evidence for implicit meaning in NLMs which cannot be explained by over-expressive probes.

1 Introduction

Given the overwhelming complexity, flexibility, and nuances of human language, it is no surprise that natural language models (NLMs) are often considered black boxes. As further insight into these models yield more evidence for underlying linguistic properties which parallel human expectations, the next logical question is to what extent they actually understand the worlds they describe.

With no external signals grounding their representations it is understandable why some researchers claim NLMs will never fully understand human language, however this does not exclude the possibility of implicit representations derived from training. Although some research has found empirical evidence for NLMs encoding implicit meaning in contextual word and sentence embeddings, the results are predicated on the accurate retrieval of the underlying representations themselves. This project seeks to validate previous claims by analyzing the capacity for probes to extract the true underlying representations from an NLM. Our analysis is rooted in previous work which found that some probes learn to complete linguistic tasks on their own rather than extracting linguistic structure encoded in the model's representation.

2 Related Work

2.1 Implicit Representations of Meaning

Some research has looked at how meaning is implicitly represented in neural language models (Li et al., 2021; Summers-Stay et al., 2021). Such research attempts to extract implicit states of entities in the setting of a given text. Consider the following sequence of sentences:

- You enter a room.
- A key is on a table.
- You pick up the key.

For the above corpus, there are Objects $O = \{$`key, table, etc.`$\}$, properties $P = \{$`empty, open, etc`$\}$, and relations $R = \{$`on, in, etc.`$\}$. These components can be combined to form **propositions** (Φ). For example, the proposition $\phi =$ `on(key, table)` would be false after the sequence above. Although it is not explicitly stated, the key is no longer on the table. This is implicitly understood given prior actions. Language models have been probed to test if these implicit relationships are contained in the LM embeddings. The use of probing LMs has shown that certain implicit relationships are, in fact, encoded in LM embeddings (Li et al., 2021). Furthermore, models perform well at generating likely thoughts and feelings by entities in the text (Summers-Stay et al., 2021). However, propositions requiring deeper thought, multi-step inferences, or reasoning are not captured by the models (Summers-Stay et al., 2021).

It is important to ensure that it is the LM embedding the relationships rather than the probe learning to perform the task itself. To control for this, model ablations were introduced. These ranged from not using language models to remapping the localization of certain entities. These results pointed in the direction that LMs are actively encoding pertinent information. However, there is still room to explore whether the probe is extracting relationships present in the LM embedding or learning to predict states based on entities in the embedding.

2.2 Control Tasks

A key question surrounding the use of probes in linguistic tasks is whether they are extracting linguistic structures and meanings encoded in the representation of the input text, or whether they are learning the linguistic tasks themselves. Some have proposed **control tasks** to test how well a probe is extracting as opposed to learning the task (Hewitt and Liang, 2019).

Control tasks test a probe by randomly mapping outputs to certain words and testing how well a probe can predict the output. In theory, a well designed probe should perform poorly on the control task and better on the true task. This is because the LM should not encode anything about the control task so the probe will have nothing to extract. If the probe performs well on the control task, this indicates it is learning to perform the task rather than extracting what the LM encodes. The difference between the two performances is called specificity. Thus, high specificity indicates the probe is extracting information rather than performing the task.

A good control task design has structure (i.e. is a deterministic function of a word type) and is random. Furthermore, the input and output space of the control task should be the same as the target linguistic task.

3 Data

Our experiment leverages Microsoft's open-source engine TextWorld to generate a language modeling dataset. The platform was created to generate text games in which players strive for implicitly defined goals by interacting with fictional environments conveyed solely through text descriptions. Reinforcement learning (RL) agents would then be trained to generate actions for the game through which researchers could test language understanding, grounding, and sequential decision making skills.

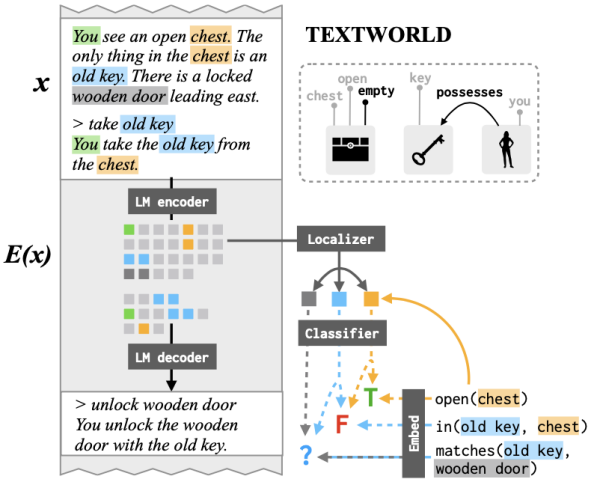

Figure 1: Overview of probe model. The LM is first trained to generate the next action and game response from prior context (left). Next, the LM encoder is frozen and the probe is trained to recover propositions about the current state (right) (Li et al., 2021).

Since our research is concerned with the description of the scene and the representation of objects, we used an NLM to generate both actions and game responses given an initial room and a list of objects accessible to player. Each action and game response consists of an English language description (x) and an **information state** (I) which encodes the players current knowledge of the game. More explicitly, for any sequence of actions and game responses $x_{1:n} = [x_1, ...x_n]$, the game maintains an information state I_i for each sequence $x_{1:i}$

Knowledge in each information state is stored by **propositions** (Φ) taking one of the following forms:

$$p(o) = \text{“the } o \text{ is } p\text{”}$$
$$r(o_1, o_2) = \text{“the } o_1 \text{ is } r \text{ } o_2\text{”}$$

Propositions can be either true (T), false (F), or unknown (?) in situations where the property of an object or the relationship between two objects has not been revealed. For example, following action i a player may find themselves in a room with a locked door and a key on a table. Until the player uses the key on the door it is impossible to know if will fit. In our model we would represent this as $\phi = \texttt{fit(key, door)}$ and $I_i(\phi) = \texttt{unknown}$. Thus each information state I_i fully encapsulates all possible properties Φ for a sequence $x_{1:i}$.

4 Model Architecture

4.1 Encoder:

Our encoder used the base T5 model without fine-tuning. T5 models are transformer models trained end-to-end taking text as both the input and the output of the model. These models have been trained on an extremely large corpus and have demonstrated potential in a wide variety of applications.

We selected the T5 architecture over a BART architecture because of prior performance advantages it has shown for this task (Li et al., 2021). We did not fine-tune the T5 model because the baseline performed well enough on the implicit meaning task to conduct an analysis of probe specificity without the need of additional training.

4.2 Probe:

To retrieve the underlying representations encoded by our T5 model we utilized a semantic probe to recover information states at various steps during the experiment. The probe consists of three parts (Figure 1):

1. A proposition embedder `embed`: $L \rightarrow \mathbb{R}^d$ (where L is the set of logical propositions and R^d is the d-dimensional encoding).

2. A localizer `loc` : $L \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ which finds, extracts, and compiles sentence fragments $x_{i:j}$ and the corresponding information states I_i which encode proposition ϕ. Previous work provides evidence that meaning in LMs is often distributed across multiple word tokens making this an imperative feature of our probe.

3. A classifier $\texttt{cls}_\Phi$: $\mathbb{R}^d \times \mathbb{R}^d = \{T, F, ?\}$ which takes a sequence of words $x_{i:nj}$ from `loc` to predict the truth value of a proposition ϕ.

When training our probes, we utilized the default linear probe suggested in the source code (Li et al., 2021).

5 Modifications

We measure the accuracy with which the probe is able to predict the labels of the control task as well as the original implicit meaning task. The difference between these two accuracies represents the selectivity of the probe and gives an indication of the quality of the probe.

Although some control tasks are introduced in the original implementation, the following control tasks will test if the probe is memorizing which word tokens are associated with certain propositions or if it is learning to extract implicit relationships from the LM.

5.1 Modification 1: Object Map

Our first modification proposes an additional control task which deterministically maps propositions to truth values. To do this, each object in the object set (O) is given a random truth-value mapping:

$$\texttt{Map}(o) = x \qquad o \in O,\ x \in_R \{\texttt{T}, \texttt{F}\}$$

The propositions are then mapped to truth values based on the first object in the proposition:

$$\texttt{Map}(\phi) = \texttt{Map}(o_1) \qquad \phi \in \Phi,\ o_1 \in O$$

For example, consider ϕ_1 = `open(chest)`, ϕ_2 = `closed(chest)`, and $\texttt{Map}(\texttt{chest}) = \texttt{T}$. Object mapping will assign both propositions the same information state (i.e. $I(\phi_1) = I(\phi_2) = \texttt{True}$). Now consider ϕ_3 = `beside(chest, table)`, ϕ_4 = `beside(table, chest)`, and $\texttt{Map}(\texttt{table}) = F$. Since `chest` appears first in ϕ_3 but not in ϕ_4, the two information states are not necessarily the same (i.e. $I(\phi_3) = \texttt{True}$, $I(\phi_4) = \texttt{False}$).

In order to learn the object assignment task on its own, we hypothesize a probe would have to memorize cases proportional to the number of objects in TextWorld. Since the number of objects is relatively small, we expect the probe accuracy to be close to that of the true implicit meaning task because the probe will be capable of memorizing the underlying mapping.

5.2 Modification 2: Context Map

Our second control task deterministically assigns values to propositions using the context and the proposition itself. If the context (`c`) is given by the text preceding proposition ϕ, then we create a mapping as follows:

$$\texttt{Map}(c, \phi) = x \qquad \phi \in \Phi,\ x \in_R \{\texttt{T}, \texttt{F}\}$$

This method ensures that not only are propositions mapped to different truth-state values, but the mapping also depends on the context of the proposition. This means that essentially every proposition that the probe seeks to classify will have a random

Experiment	Average EM	Specificity
Original	0.943	NA
Baseline	0.838	NA
Object Mapping	0.938	-0.100
Context Mapping	0.619	0.219

Table 1: Results from the remapping performed in modifications 1 and 2. Original experiment refers to results from (Li et al., 2021).

and deterministic mapping. Since training occurs over multiple text corpuses, the context associated with each proposition will vary from query to query. Thus, we expect the number of unique assignments to be roughly equivalent to the number of total queries. Consequently, we expect the probe accuracy to be lower than the baseline task since it will be difficult for the probe to memorize mappings for each unique context.

6 Results

Probes were evaluated using the same metric defined in the original paper, **Average Exact-Match (EM)**, which calculates the percentage of entities, entity pairs, and information states for which all propositions are correctly labeled (Li et al., 2021). We also include the specificity of the probe under our two modifications. A summary of the key findings can be found in Table 1.

Additionally, we tracked the Average EM on the validation set as we trained the probe. The probe quickly learned the mapping for the object control whereas the probe could not attain as high of average EM in the context control experiment. The results of this can be seen in Figure 2.

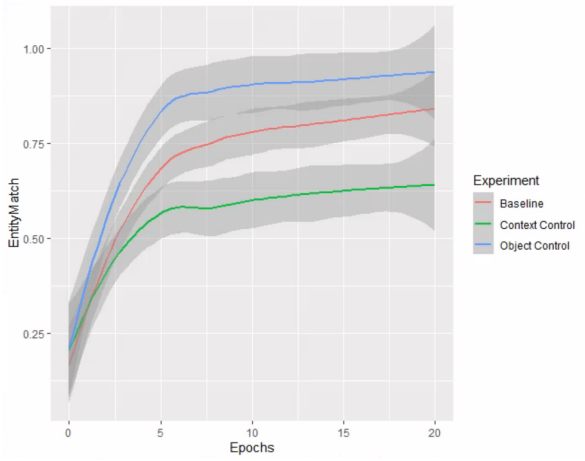

Figure 2: Average EM for all three experimental conditions

7 Analysis

Low Selectivity in Object Mapping: As we expected, the accuracy of the probe on the object mapping control task was very high, surpassing expectations and outperforming the baseline task. In conjunction with a negative specificity score, this indicates that the probe was effectively memorizing the small number of mapping tasks assigned to it.

High Selectivity in Context Mapping: Results in Table 1 support our hypothesis that probes are unable to memorize a large number of relationships on their own, indicated by the low average EM and high specificity of our context mapping control task.

From the results of the object mapping task we can see that probes are able to learn certain tasks as indicated in the control task paper (Hewitt and Liang, 2019). However, the context mapping control task indicates that it is not likely that the probe is learning the baseline task since the sheer number of parameters exceeds the capacity of the probe. This validates the claims laid out in the implicit meaning paper (Li et al., 2021).

8 Conclusions and Future Work

In this paper we documented our reproduction of Li, Nye, and Andreas's paper "Implicit Representations of Meaning in Neural Language Models" in addition to two modifications (Li et al., 2021). Our focus was to validate claims in the original paper by creating control tasks inspired by Hewitt and Liang's paper "Designing and Interpreting Probes with Control Tasks" (Hewitt and Liang, 2019). The first task modified training data for the probe by mapping propositions to fixed truth values using the first object in each proposition. Results from this experiment validated our intuition that probe are able to learn simple memorization tasks. The second task modified training data by assigning

propositions to truth values based on the context in which the propositions were located. Our results showed high selectivity for this experiment, suggesting the probe was unable to memorize a large number of relationships, thus validating the original papers conclusion that probes are able to successfully extract implicit representations of meaning from nerual language models.

Credits

This is a final project for COS484, Spring 2022 at Princeton University. Special thanks to professor Karthik Narasimhan and TA Carlos Jimenez for providing feedback on our project.

Project Files

https://github.com/Dreamweaver2k/state-probes-control-tasks

References

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. *CoRR*, abs/1909.03368.

Belinda Z. Li, Maxwell I. Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. *CoRR*, abs/2106.00737.

Douglas Summers-Stay, Claire Bonial, and Clare Voss. 2021. What can a generative language model answer about a passage? In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 73–81, Punta Cana, Dominican Republic. Association for Computational Linguistics.