STANDARDS FOR BELIEF REPRESENTATIONS IN LLMS

DANIEL A. HERRMANN, BENJAMIN A. LEVINSTEIN

Abstract. As large language models (LLMs) continue to demonstrate remarkable abilities across various domains, computer scientists are developing methods to understand their cognitive processes, particularly concerning how (and if) LLMs internally represent their beliefs about the world. However, this field currently lacks a unified theoretical foundation to underpin the study of belief in LLMs. This article begins filling this gap by proposing adequacy conditions for a representation in an LLM to count as belief-like. We argue that, while the project of belief measurement in LLMs shares striking features with belief measurement as carried out in decision theory and formal epistemology, it also differs in ways that should change how we measure belief. Thus, drawing from insights in philosophy and contemporary practices of machine learning, we establish four criteria that balance theoretical considerations with practical constraints. Our proposed criteria include **accuracy**, **coherence**, **uniformity**, and **use**, which together help lay the groundwork for a comprehensive understanding of belief representation in LLMs. We draw on empirical work showing the limitations of using various criteria in isolation to identify belief representations.

> "The truth may be out there, but the lies are inside your head."
>
> — Terry Pratchett, *Hogfather*

1. Introduction

Large language models (LLMs) have been doing remarkable things: they can write code, summarize text, role play as different characters, and even play games of strategy like chess at a reasonable level. In light of these recent achievements, there has been a push to understand how they are able to accomplish these feats and how their cognition (if they have it) works. In particular, computer scientists have been developing methods that aim to read things like *belief* and *world models* off of both the internal activations and the behavior of the LLM (Li et al. (2023); Olsson et al. (2022); Bubeck et al. (2023)).

This work is valuable and exciting, but it is currently in a pre-paradigmatic state; individual groups are deploying engineering-style solutions in order to solve particular problems but without a shared understanding of the overall goal and theoretical basis of such an endeavor. The field currently lacks a philosophically rigorous and practice-informed *conceptual foundation* of belief representation in LLMs.

In this article we begin filling this gap. To do so, we propose conditions of adequacy for an LLM to have a belief-like representation. Our conditions are motivated by insights from decision theory and formal epistemology, as well as by the details of actual machine learning models and practices. They build upon our previous work, and aim to address some of the shortcomings we identified in contemporary belief measurement techniques in LLMs (Levinstein and Herrmann 2024). A central upshot of our proposal is its ability to guide the development of future belief measurement techniques.

2. The Basics of LLMs

Contemporary LLMs are based on the transformer architecture first described by Vaswani et al. (2017). Here we'll focus on decoder only (autoregressive) models like ChatGPT.

The basic idea is as follows. Some prompt like `New Orleans is in` is fed to the model. Each token (in essence, word, subword, or punctuation mark) gets converted into an initial vector called an embedding. Each embedding gets "massaged" as it passes through a long chain of computations called layers. Finally, the model outputs a probability distribution over what the next token will be. In this case, `Louisiana` should get high probability if the LLM is good, and `banana` and `aardvark` should get low probability.[1]

At each of these layers, two things happen. The first is that the model moves information around from earlier tokens to later ones through the mechanism of attention. The embedding for `Orleans` in the above prompt receives information from the embeddings for `New` and for `Orleans`, while the embedding for `in` receives information from the embeddings from `New`, `Orleans`, `is`, and `in`. The second is that the embedding for any given token passes through a vanilla neural network (i.e., a multi-layer perceptron, though usually with a single hidden layer).

Ultimately, the prediction for the next token is based solely on the embedding at the last layer for the last token. So, one way or another, the model must move all information relevant to prediction of the next token to this embedding. In other words, as illustrated in fig. 1, the computational graph for transformers is directed and acyclic, with information flowing from earlier tokens to later tokens and from earlier layers to later layers.

To generate new text, we select a token the model assigns relatively high probability to, append it to the prompt, and then feed the new prompt to the LLM.

The way LLMs get so impressively good at generating text is via training. Training comes in two phases. In the first phase, called *pre-training*, we take a passage (e.g., a Wikipedia article) and give the model an initial

[1] In actual LLMs, tokens are typically smaller than full words, but we use words as convenient illustrations of the core ideas.

segment of that passage. The model predicts what comes next. We slightly adjust the model's parameters so that it would assign a higher probability to the actual next token were it to be fed the same initial segment of the passage again. After being trained on the high-quality portion of the internet multiple times, large models are able to achieve impressive fluency.

The second phase is called fine-tuning. Fine-tuning comes in many forms, but the most popular models like ChatGPT that retail users interact with are refined using Reinforcement Learning from Human Feedback (RLHF) or some variant, such as Constitutional AI.[2] In essence, these methods make models be more conversational and get better at telling users what they want to hear.

Without going into too much technical detail, RLHF works by having the model generate multiple answers to the same prompt, having users rate the responses, and then gradually training the model to output responses rated higher. (At some point, this training is usually assisted by a second AI model that learns to predict what users will like.)

3. Beliefs in LLMs

Now that we have a handle on the basics of LLMs, let's consider why we might want to attribute *beliefs* to them. As we described in §2, LLMs are trained first to minimize predictive loss (the pre-training stage), and then to output text that is in some sense desirable (the fine-tuning and RLHF stage). Furthermore, as described in §1, LLMs are very successful at achieving these goals. We want to understand *how* they are so successful.

A standard explanation is that they are successful partially because they represent certain features of the world, and they use these representations to help decide what text to output (Li et al. (2023); Olsson et al. (2022); Bubeck et al. (2023), Marks and Tegmark (2023)). For example, there is strong evidence that LLMs have both spatial and colour representations (Abdou et al. (2021), Patel and Pavlick (2021)).

Given how useful it is to track truth in many contexts, it should be a live hypothesis that LLMs represent whether or not certain sentences are true, just as they represent direction and colour. Indeed, if it is the case that LLMs are representing the truth at least some of the time, and this representation helps guide their outputs, then having a way to measure what they believe might help us make predictions about the model. Moreover, if our measurements and understanding of the LLM's representation are good enough, this might allow us to intervene usefully on the model by manipulating the representation.

Some have expressed skepticism that LLMs have anything resembling beliefs, *even in principle* (Bender and Koller (2020); Bender et al. (2021); Shanahan (2022)). In particular, these arguments tend to rely on the claim

[2] For RLHF see Christiano et al. (2017); Stiennon et al. (2020); and Ouyang et al. (2022); for Constitutional AI see Bai et al. (2022) and Kundu et al. (2023).