Fact Recall, Heuristics or Pure Guesswork? Precise Interpretations of Language Models for Fact Completion

Anonymous ACL submission

Abstract

Recent work in mechanistic interpretability of language models (LMs) has established that fact completion is mediated by localized computations. However, these findings rely on the assumption that the same computations occur for all predictions, as long as the model is accurate, and aggregate results for these. Meanwhile, a parallel body of work has shown that accurate fact completions can result from various inference processes, including predictions based on superficial properties of the query or even pure guesswork. In this paper, we present a taxonomy of relevant prediction mechanisms and observe that a well-known dataset for interpreting the inference process of LMs for fact completion misses important distinctions in this taxonomy. With this in mind, we propose a model-specific recipe for constructing precise testing data, which we call PREPMECH. We use this data to investigate the sensitivity of a popular interpretability method, causal tracing (CT), to different prediction mechanisms. We find that while CT produces different results for different mechanisms, aggregations are only representative of the mechanism that corresponds to the strongest signal. In summary, we contribute tools for a more granular study of fact completion in language models and analyses that provide a more nuanced understanding of the underlying mechanisms.

1 Introduction

Improving our understanding of how language models process and respond to factual queries can inform a safer and more efficient use of these systems. One field that aims to examine and explain model behavior is mechanistic interpretability (Elhage et al., 2021; Geiger et al., 2021). Recent work by Meng et al. (2022); Geva et al. (2023); Haviv et al. (2023) has focused on the inference process of LMs for fact completion for simple (subject, relation, object) fact tuples, illustrated in Figure 1. This body of work hypothesizes that LMs follow a distinct process when producing accurate fact completions, namely that LMs recall information stored in middle range MLP layers.

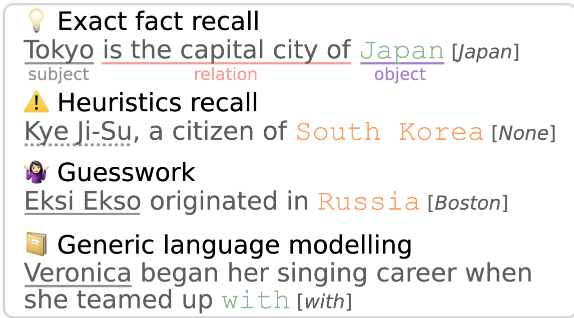


Figure 1: Prediction mechanisms and fact completion examples. Words in `code font` indicate model predictions for the missing object and words in [brackets] indicate the gold label. Subjects are underlined and dashed underlines signify synthetic subjects.

Meanwhile, research into model performance on factual benchmarks has shed light on different factors affecting a prediction. Work by Poerner et al. (2020); Cao et al. (2021); Ladhak et al. (2023) found that accurate LM predictions in fact completion situations may stem from shallow heuristics, such as lexical overlap, person name bias or prompt bias. Work on fact editing (De Cao et al., 2021) as well as probing for factual knowledge (Elazar et al., 2021), has illustrated issues with consistency (i.e. a model switching its prediction when the prompt is rephrased), while other knowledge probing investigations (Kandpal et al., 2023) have demonstrated that models struggle more with facts rarely seen during training, suggesting a correlation between training data frequency and memorization.

By assuming that accurate predictions correspond to one distinct process, previous interpretations of LMs disregard fine-grained factors that influence LM predictions. In this work we provide an approach for exploring these nuances and analyze how they may affect the model and interpretations of it. Our contributions can be summarized as follows:

- We present a detailed taxonomy of different types of inference processes, referred to as *prediction mechanisms*, related to factual queries (see Figure 1) and explore these for a dataset previously used to study fact completion, showing the need for a more precise dataset.
- We propose a method for creating model-specific datasets that contain examples of each separate mechanism in our taxonomy. We create and release the datasets PREPMECH for GPT-2 XL and Llama 2 7B, respectively.
- Using PREPMECH, we evaluate the sensitivity of a popular interpretability method – causal tracing (CT) – for detecting and measuring different prediction mechanisms. We observe how this method yields distinctive results for each prediction mechanism in isolation, while results based on aggregations over multiple prediction mechanisms are imprecise and dominated by the characteristics of only one mechanism.[1]

Mechanism	Fact compl	Confi-dent	No heuristics
Generic LM	✗	-	-
Guesswork	✓	✗	-
Heuristics recall	✓	✓	✗
Exact fact recall	✓	✓	✓

Table 1: Our four identified prediction mechanisms and their corresponding three criteria. A '-' denotes that the mechanism does not differentiate between ✓ and ✗ cases. Generic LM refers to generic language modeling, and fact compl to fact completion.

2 Prediction mechanisms

Mechanistic interpretability aims to explain model behavior by investigating the underlying computations (Conmy et al., 2023). Results are typically validated on datasets with examples that can be assumed to trigger the computation under consideration. Therefore, ensuring a close match between the dataset and the targeted phenomenon is crucial. Such a close match may not hold for previous studies of LMs for fact completion, which distinguish between queries that *do* recall factual associations and those that do not based on the models' accuracy when responding to these queries (Meng et al., 2022; Geva et al., 2023). Some authors even go so far as to define the model "knowing a fact" as its ability to elicit the correct answer through a prompt (Petroni et al., 2019). This perspective yields a very coarse categorization of model behavior and does not align well with previous studies showing that accurate predictions may result from different prediction mechanisms with varying levels of reliability, such as predictions based on surface-level artifacts in the query (Poerner et al., 2020; Cao et al., 2021; Ladhak et al., 2023). Therefore, in this paper, we aim to introduce a precise and comprehensive conceptual framework of different LM inference processes for fact completion. We refer to them as *prediction mechanisms*.

We define three fine-grained criteria important for a precise evaluation of model prediction mechanisms in fact completion. By exploring the factors affecting accuracy rather than working with accuracy directly, we can disentangle the underlying phenomena. Specifically, our criteria are (1) whether the prediction actually represents fact completion rather than generic language modeling (Section 2.1); (2) whether the prediction is confident and robust to insignificant signals in the prompt (Section 2.2); and (3) whether the prediction is based on the exact factual information expressed in the query or on heuristics triggered by surface-level artifacts (Section 2.3). Based on relevant combinations of these criteria, we define four prediction mechanisms, as indicated in Table 1 and discussed in the sections below. We argue that these mechanisms should be studied in separation since they rely on disparate signals with varying degrees of soundness and correctness for fact completion situations.

We conclude the section with a description of how we implement the criteria in practice and investigate them for a dataset previously used for the study of fact completion – the known samples from CounterFact (Section 2.4). These are the 1,209 examples from the data for which GPT-2 XL produces a correct completion for the prompt.

2.1 Generic language modeling

The first criterion we consider is *fact completion* – whether a prompt and the corresponding prediction exemplify the setting of a model completing a fact. A precise study of model behavior in fact-

[1] All of our code and data will be open-sourced once the anonymity period is over.

intensive situations relies on only studying queries that necessitate the processing of a fact. One way to ensure this is to work with queries corresponding to fact completion, exemplified in Figure 1.

Based on the fact completion criterion, we define one of our four prediction mechanisms – the *generic language modeling* mechanism – important for baseline comparisons. This mechanism is assumed to take place for generic model predictions, illustrated in Figure 1, and to be different from mechanisms taking place for factual completion situations (Haviv et al., 2023).

2.2 Random guesswork

The second criterion is *confident prediction* – whether the prediction is robust across insignificant perturbations to the query. Since LMs cannot abstain from answering, we may end up in situations when a LM makes the correct prediction by chance while it has a near-uniform output distribution. Stored model knowledge should correspond to confident and robust predictions for prompts that request the stored knowledge.

Based on the prediction confidence criterion we define our second prediction mechanism – *random guesswork* – corresponding to unconfident model predictions in fact completion situations. These predictions can be accurate or inaccurate.

2.3 Heuristics and exact fact recall

The final criterion is *no dependence on heuristics* – indicating the prediction is based on the exact factual information expressed in the prompt (subject and relation) rather than only on partial signals. Poerner et al. (2020) and Cao et al. (2021) found that accurate fact completion may stem from surface level artifacts, such as lexical overlap, person name bias or prompt bias. As can be seen from Figure 1, for example, where the synthetically generated person name "Kye Ji-Su" is predicted to be a citizen of "South Korea" probably due to the structure of the name (name bias). Such predictions indicate an over-reliance on unintended correlations in the training dataset based on surface forms of names or prompts, and are therefore unreliable (Cao et al., 2021; McCoy et al., 2019). Recalling information that is disputable and overgeneralizing (that is, capturing some statistical pattern that is only partially correct) is not equivalent to recalling the exact fact requested by a prompt.

Based on this final criterion, we separate *exact fact recall* from *heuristics recall*. Both mechanisms denote when the LM makes use of stored information for its prediction, i.e. performs a *recall*. The difference lies in what type of information is recalled and what the recall is based on. Heuristics recall occurs for predictions based on learned overgeneralized heuristics triggered by surface level artifacts. Exact fact recall corresponds to situations for which a LM has memorized the full fact tuple expressed by the prompt and fetches this from memory for the prediction. We assume the prediction mechanisms for these two instances to be different due to their fundamental differences in the information used. Furthermore, since predictions based on heuristics are far less reliable compared to predictions based on exact fact recall, it is important that we analyze them separately.

2.4 Detecting prediction mechanisms

Here, we outline our choice of detection methods for the criteria described above. We also use these methods to inspect a dataset frequently used for the interpretation of LMs performing fact completion, namely, the 1,209 known samples from CounterFact for which GPT-2 XL is accurate (Meng et al., 2022; Geva et al., 2023).

Fact completion To ensure we study fact completion, we follow previous work (Petroni et al., 2019; Meng et al., 2022; Geva et al., 2023) and limit ourselves to simple queries that express an incomplete fact tuple subject–relation, with the intent to let the LM generate the object as the next token. Each of our samples thus consists of a query and the corresponding model output. The authors of CounterFact Meng et al. (2022) let the model generate freely until it produces an entity, but this may distort the original meaning of the template, e.g., by adding negation (Appendix L.4). Therefore, we only retain (*query*, *prediction*) samples for which the next token corresponds to an entity or concept that can be considered relevant for fact completion. This excludes tokens such as "the", "a" and "with". The known CounterFact examples also fulfill the criterion on fact completion.

Confident prediction There is a wide variety of methods proposed for estimating model confidence. Research on model calibration (Jiang et al., 2021; Vasudevan et al., 2019) has shown that token probability does not align with performance and as such cannot be used as a good approximation of confidence. Some research has suggested, however, that other internal model states may encode information related to model confidence (Burns et al., 2023). However, different extraction methods have varying success and are model as well as dataset dependent (Yoshikawa and Okazaki, 2023). Additionally, most of this work is from the field of model calibration, and uses accuracy as the single measure of performance.

In this paper, we opt for a definition of confidence grounded in desirable model behavior. We proxy model confidence by consistency in the face of semantically equivalent queries (Elazar et al., 2021; Portillo Wightman et al., 2023) and use paraphrases from the ParaRel dataset (Elazar et al., 2021). More specifically, we classify a prediction as confident if it occurs among the top 3 predictions for at least 5 paraphrased queries. A prediction that only appears for one of the rephrased queries is deemed unconfident. We cannot estimate confidence for the known CounterFact samples since the dataset only provides one prompt per fact.

No Heuristics To detect surface-level signals indicating the potential use of heuristics, we use filters based on prompting the LM under investigation, as proposed by Poerner et al. (2020); Cao et al. (2021). We also complement this approach with memorization estimations based on work by Mallen et al. (2023) and Kandpal et al. (2023).

For the surface-level filters, we make use of person name bias and lexical overlap filters by Poerner et al. (2020). Person name bias can only be detected for relations where the subject is a person name and the object is a location. We also build a prompt bias filter based on the findings by Cao et al. (2021). Lexical overlap is detected if there is a string match between subject and object. Prompt and person name bias are detected by querying the model with the partial fact – i.e. expressing only the relation with a generic subject, or querying for a typical location associated with the name without specifying how that location is related to the subject. The templates used for prompting can be found in Appendix G. These filters only reveal the possibility of heuristics recall taking place.

For exact fact recall, we also complement our detection method with LM knowledge estimations. Previous work in this field indicates that queries asking for fact tuples rarely found in the LM training data are less likely to be known by the model (Mallen et al., 2023; Kandpal et al., 2023). If we know that the LM does not know the fact requested by a prompt but still makes a confident prediction, we can assume that it corresponds to some form of heuristics recall. Similarly, if we know with high certainty that the LM ought to know a particular fact, we have higher reason to believe that the correct prediction for a query asking for this fact corresponds to exact fact recall. Following Mallen et al. (2023), we use fact popularity to approximate frequency in training data. Similarly to their approach we measure fact popularity by Wikipedia page views and collect the average monthly Wikipedia page views for year 2019 for each query subject and object using the Pageview API.[2] Mallen et al. (2023) found queries with popularity scores below 1000 unlikely to have been memorized, unless surface-level artifacts were present. We label a prediction as corresponding to a known fact only if it is accurate and the fact corresponds to an average page view above 1000.

For heuristics recall, we ensure no exact fact memorization is taking place by using synthetic data. If a model has a highly confident prediction, we can assume it has identified some heuristics to guide that decision. We combine this with prompt-based bias detection and only select synthetically generated facts that have been detected by the bias filter. This provides assurances that the model could not have performed exact fact recall and we have evidence that specific types of surface level signals were present instead, making it most likely that those were used to provide a prediction.

Analysis of known CounterFact samples We check for predictions based on shallow heuristics for the known CounterFact samples. We find 335 samples that may correspond to prompt bias, 155 to name bias and 20 to both name and prompt bias. There are a total of 205 samples corresponding to person names for which we can check for name bias, meaning that we detect name bias in 175 of a total 205 samples. No lexical overlap between sample subject and object is found. Some examples marked for bias can be found in Appendix L.1.

Using fact popularity, we also evaluate the known CounterFact samples through the lens of LM knowledge estimation. Appendix L.2 lists the popularity scores distribution for the dataset. We find approximately 365 known CounterFact samples with popularity scores below 1000. These are unlikely to have been memorized by the model and

[2] https://wikitech.wikimedia.org/wiki/Analytics/AQS/Pageviews

Mechanism	GPT-2 XL #samples (#fact tuples)	Llama 2 7B #samples (#fact tuples)
Exact fact	1322 (191)	5481 (580)
Heuristics	8352 (1868)	8414 (1960)
Guesswork	3282 (3181)	2917 (2846)
Generic LM	1000 (-)	1000 (-)

Table 2: Statistics for our PREPMECH dataset for each LM considered in our study.

are therefore unlikely to correspond to exact fact recall. Moreover, we find that around 50% of these samples (172 samples) have been detected by our heuristics filters (Appendix L.2), indicating that the remaining samples may also contain surface level signals not detected by our filters. This supports our claim that popularity metadata can serve as a complement for separating exact fact recall samples from heuristics recall samples.

Apart from the analysis described above, we also scrutinize the known CounterFact samples with respect to the total effect of perturbing the subject (Appendix L.3) and negated queries (Appendix L.4). Our results indicate an additional set of potentially problematic samples that may hinder a precise study of prediction mechanisms.

3 PREPMECH: a dataset for precise studies of prediction mechanisms

To facilitate precise interpretations of prediction mechanisms, we develop the dataset PREPMECH (PRecise Examples of MECHanisms) with samples that separately trigger each mechanism identified in Section 2. The dataset and our subsequent analysis is focused on the English language. This section describes our sampling methods for queries corresponding to exact fact recall (Section 3.1), heuristics recall (Section 3.2), random guesswork (Section 3.3) and generic language modeling (Section 3.4). PREPMECH is model-specific since it indicates samples learned by a model and predictions based on model biases, which differ between LMs. We develop a dataset for GPT-2 XL (Radford et al., 2019) and Llama 2 7B (Touvron et al., 2023), respectively. General statistics for PREPMECH can be found in Table 2. Appendix K includes examples corresponding to each prediction mechanism.

3.1 Exact fact recall samples

To get queries for which the LM performs exact fact recall, we first build a dataset based on LAMA and ParaRel query templates (Petroni et al., 2019; Elazar et al., 2021). We then extract exact fact recall predictions from this dataset based on the criteria and methods described in Section 2.4. Extracted exact fact recall predictions are 1) not labelled as corresponding to any bias, 2) correct, 3) corresponding to a fact known by the LM and 4) confident. It is not a problem if this excludes samples corresponding to exact fact recall, as we prioritize precision for these samples. A more detailed description of our sampling process for the exact fact recall samples can be found in Appendix D.

The composition of the relations that make up the exact fact recall samples is further analyzed in Appendix H. We note that the majority of the samples in the dataset are based on the relations P740 *location of formation* and P1376 *capital of*.

3.2 Heuristics recall samples

To provide a testing ground for comparing results to a baseline case where we can be certain the model is performing recall of heuristics, we use synthetic tuples in place of LAMA tuples. Since the fact tuples represented by the samples are synthetic, they cannot have been memorized by the model (Liu et al., 2023; Basmov et al., 2024). Confident predictions for these samples should therefore correspond to heuristics recall.

To obtain the relevant data and labels, we first build a dataset based on synthetic fact tuples and ParaRel query templates. We then apply our criteria as described in Section 2.4. Confident predictions for which a single type of bias is identified form our heuristics recall samples. A more detailed description of our sampling process and deeper analysis of the heuristics recall samples can be found in Appendix E and Appendix I, respectively.

3.3 Random guesswork samples

To collect samples corresponding to random guesswork, we start from the same source data as Section 3.1 and filter for samples that are 1) unconfident, 2) found in the gold label set (correspond to a fact completion situation) and 3) not corresponding to a fact known by the LM.

3.4 Generic language modeling

To get samples corresponding to generic language modeling we use Wikipedia[3], following an approach similar to that of Haviv et al. (2023), and collect sentences that start with the subject of the article. The extraction is done by sampling subject-first examples in order to mirror the fact completion setting, while exploring the role of the subject in a natural, but not fact completion setting (see Appendix F for details).

4 Sensitivity of causal tracing to different prediction mechanisms

To illustrate the importance of precise interpretations of LMs, we investigate the sensitivity of a popular mechanistic interpretability approach – causal tracing (CT) – to different prediction mechanisms and their aggregations.

CT is a mechanistic interpretability method that has been highly influential and provided interpretations of LMs (Stolfo et al., 2023; Monea et al., 2023). The method works by first recording intermediate model representations during normal generation (clean run). Then noise is added to the query subject embeddings to obtain corrupted intermediate model representations (noised run). By restoring corrupted representations at different token-layer positions it is possible to infer what parts of the network are important for assigning a high probability to the predicted token with respect to the subject (patched run). The measured signal is referred to as *indirect effect* and defined as

$$\mathrm{IE}_{h_i^{(l)}}(o) = P_{h_i^{(l)},\,\mathrm{patched}}(o) - P_{\mathrm{noised}}(o) \quad (1)$$

where $P_{h_i^{(l)},\,\mathrm{patched}}(o)$ is the probability for token o after patching state $h_i^{(l)}$ at layer l for the input token at position i and $P_{\mathrm{noised}}(o)$ is the probability of o for the noised run. To reason about the general process of generating a prediction, results for important states are averaged over several samples to get the average indirect effect (AIE).

Our sensitivity analysis of CT is centered around two questions: (1) Are aggregation plots of CT results representative of the whole sample? and (2) Do the CT results and corresponding conclusions change with the underlying prediction mechanism? To answer these questions, we concretize the conclusions reached by previous CT studies in Section 4.1, perform an aggregation analysis described in Section 4.2, and present results in Section 4.3.

4.1 Conclusions from previous CT studies

Based on aggregations of CT results for accurate fact completions, Meng et al. (2022) conclude that MLP modules at mid model layers at the last subject token have a decisive role for fact completion. In this work, we refer to this conclusion as the *decisive role conclusion*. Based on this conclusion, Meng et al. (2022) reason that MLP module computations at middle layers have an essential role when recalling a fact and that their results reveal the location of MLP key–value mappings capable of recalling facts about a subject. As this conclusion is a central part of original CT studies, we focus our investigations on whether results for different prediction mechanisms lead to it. If results for all mechanisms lead to the same conclusion, it would indicate that CT is not sensitive to different prediction mechanisms. CT results leading to the decisive role conclusion are defined as results for which MLP states at (last subject token, mid-layers) are decisive, i.e. yield an AIE with a lower confidence bound higher than the AIE upper confidence bound for any other (token, layer) state.

4.2 Aggregation analysis

Meng et al. (2022) averaged CT results over 1000 samples in order to reason about the general pattern of recall of factual associations. Since these results are dependent on the absolute values of the probability of the traced (predicted) token, we hypothesize that the result could be driven by a few high-probability samples and not representative of the low-probability[4] strata of the data. To test this, we take inspiration from work by Hase et al. (2023) and compare the IE results to their normalized counterpart. We define the *normalized indirect effect* as

$$\mathrm{NIE}_{h_i^{(l)}}(o) = \frac{P_{h_i^{(l)},\,\mathrm{patched}}(o) - P_{\mathrm{noised}}(o)}{|P_{\mathrm{clean}}(o) - P_{\mathrm{noised}}(o)|} \quad (2)$$

where $P_{\mathrm{clean}}(o) - P_{\mathrm{noised}}(o)$ is the total effect (TE) defined as the difference between the clean and the noised runs. The normalized IE measures the percentage of recoverable probability that was recovered by patching state $h_i^{(l)}$.

[3]We use `20220301.en` from HuggingFace at https://huggingface.co/datasets/wikipedia

[4]With *probability*, we here refer to the probability corresponding to the clean run prediction.

For some samples, predominantly low-probability predictions, the division by the TE may result in unnatural $\mathrm{NIE}_{h_i^{(l)}}(o)$ values above 1 or below -1. The state patching should not be able to restore more than the clean run probability and we therefore cap the $\mathrm{NIE}_{h_i^{(l)}}(o)$ to a range of $[-1, 1]$. With this approach, each sample is valued on the same scale. Plots for homogeneous datasets should therefore yield normalized CT results that are similar to their non-normalized counterparts.

4.3 Results

Figure 2 shows average indirect effects of different states in GPT-2 XL for 1000 samples composed of 400 exact fact recall, 400 heuristics recall and 200 guesswork samples from PrepMech.[5] This figure also indicates the results for samples corresponding to each prediction mechanism in isolation, allowing us to study the effect of aggregation. The corresponding results for Llama 2 7B can be found in Appendix M.1. We use the same hyperparameters as Meng et al. (2022) for our CT analysis. Our aforementioned questions are answered, as follows.

Are aggregated CT results representative of each studied sample? The results for the non-normalized plot in Figure 2a are dominated by the exact fact recall samples with larger non-normalized indirect effects. The exact fact recall samples clearly lead to the decisive role conclusion and the same holds for the non-normalized results, even though subsets of the included data (heuristics recall and guesswork samples) do not lead to the same conclusion with as high certainty.

For the normalized results in Figure 2b we find that equal weights for all evaluated samples yield a different pattern compared to the non-normalized results, with a weaker peak for the last subject token. Moreover, we find that normalization yields the same pattern when applied to samples of isolated mechanisms (e.g. Figure 2c and Figure 2f). We conclude that aggregations of CT results across multiple prediction mechanisms are not representative of each studied sample. Also, comparisons between non-normalized and normalized results may reveal nonhomogeneous datasets with respect to prediction mechanism. The results for Llama 2 7B in Figure 3 support the same conclusions.

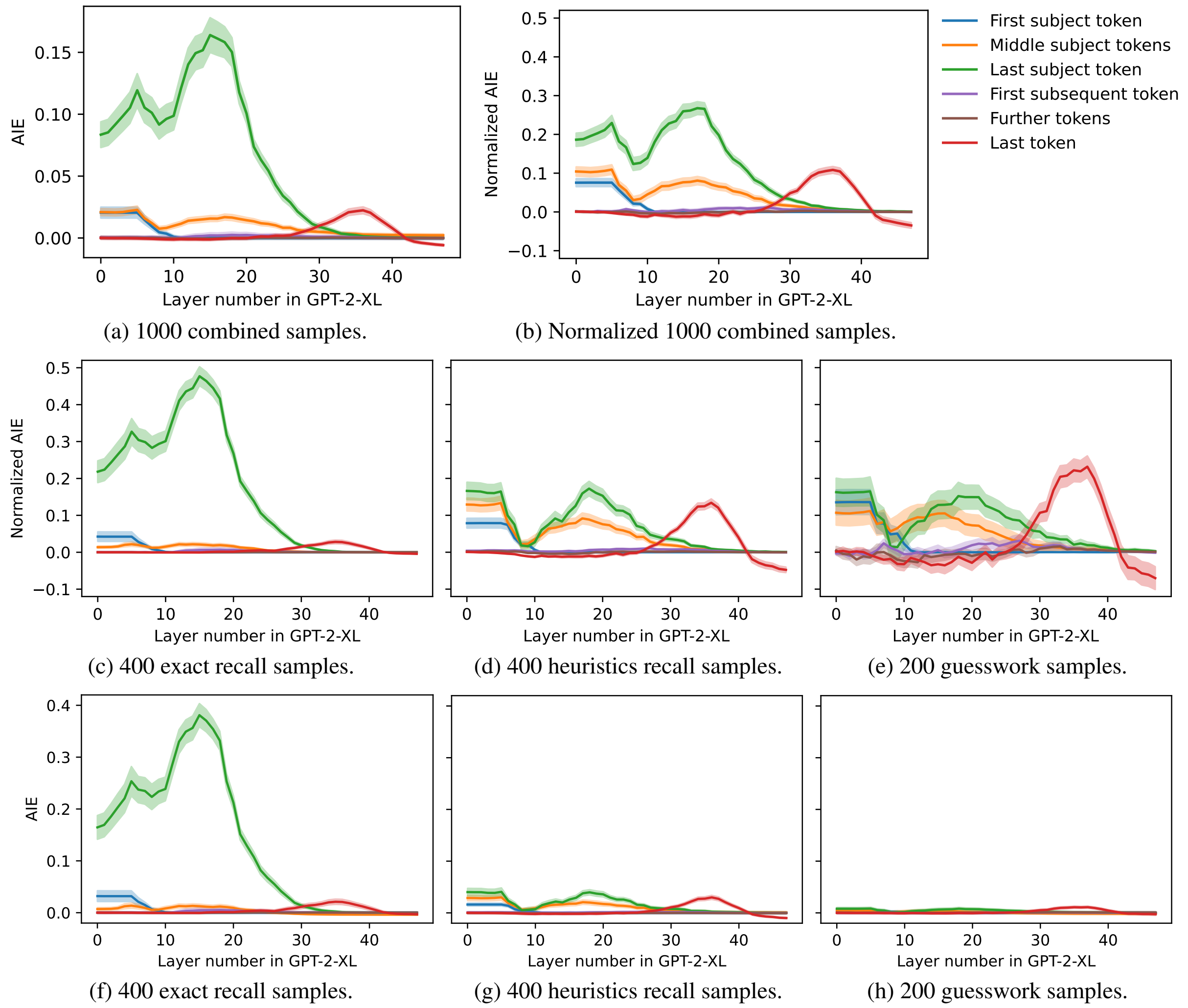
[5] Appendix M.3 includes normalized CT plots for each prediction mechanism for GPT-2 XL and Llama 2 7B. Results for the subsets are found to be representative of the larger sets.

Do the CT results and corresponding conclusions change with the underlying prediction mechanisms? For samples corresponding to each prediction mechanism in isolation, we find distinct differences between the normalized CT results for each mechanism. For the exact fact recall samples, the significance of the last subject token state at early to middle layers is profound compared to all other (token, layer) states. Evidently, the LM relies heavily on information from MLP mid-layers for the exact fact recall prediction mechanism. For the heuristics recall samples, the importance of the last subject token state is downplayed and the importance of the last token state is increased as well as the importance of all subject tokens in early layers. The heuristics recall results still lead to the decisive role conclusion, but with a small margin. For the guesswork samples, the last token state is decisive and the results do not lead to the decisive role conclusion. The Llama 2 7B results in Figure 3 show similar trends: while both the heuristics recall and guesswork samples lead to the decisive role conclusion, they do so with a smaller margin compared to the exact fact recall samples.

Additional analysis. We already noted that CT is sensitive to prediction probabilities. This also holds when the underlying prediction mechanism is kept constant. Appendix M.2 includes the same plot as in Figure 2 but for samples corresponding to the lowest model probabilities for each prediction mechanism in PrepMech. For these samples, the last token state is assigned a higher importance with all prediction mechanisms. Normalized CT results for generic language modeling in Appendix M.3 do not indicate a decisive role for any MLP state corresponding to the last subject token.

Appendix M.4 presents normalized CT results for the heuristics recall samples partitioned by prompt bias and person name bias. The prompt bias results suggest a higher importance of the last token state, compared to the last subject token state, when compared to the person name bias results.

We conclude that CT is sensitive to different prediction mechanisms, and therefore CT results yield different interpretations depending on the selection of samples. We find normalization of results to be a feasible approach to indicate sample nonhomogeneity. Furthermore, we find alignment with previous work regarding the importance of middle MLP layers of the last subject token in our exact fact recall sub-sample. However, our results



(a) 1000 combined samples.
(b) Normalized 1000 combined samples.
(c) 400 exact recall samples.
(d) 400 heuristics recall samples.
(e) 200 guesswork samples.
(f) 400 exact recall samples.
(g) 400 heuristics recall samples.
(h) 200 guesswork samples.

Figure 2: CT results with GPT-2 XL for 1000 samples from PrepMech of which 400 samples correspond to exact fact recall, 400 to heuristics recall and 200 to guesswork. Shaded regions indicate 95% confidence intervals.

suggest there might be other processes at play for heuristics recall and guesswork that warrant further investigation. Finally, CT results are sensitive to prediction probabilities, even when the prediction mechanism is held constant. This potentially indicates room for improvement with respect to our metrics for prediction confidence.

5 Conclusion

Based on a set of basic criteria, we identify four prediction mechanisms that are fundamentally different and of differing reliability. These are *exact fact recall*, *heuristics recall*, *guesswork* and *generic language modeling*. We show that previous interpretability work for fact completion situations treat many of these mechanisms as equivalent by using accuracy as the sole criterion for differentiating between prediction mechanisms. Our analysis of a dataset frequently used by previous interpretability work – known examples from CounterFact – reveals samples that may trigger heuristics recall as opposed to exact fact recall and other problematic phenomena. To facilitate precise interpretations of prediction mechanisms, we present a method for creating a model-specific dataset PrepMech with samples that separately trigger each of our identified prediction mechanisms. We produce a version of this datasets for each of GPT-2 XL and Llama 2 7B, and use it to test the prediction mechanism sensitivity of an influential interpretability method, causal tracing (CT). We find that different prediction mechanisms yield distinct CT results if studied in isolation. Consequently, CT results are not representative of the dataset as a whole if it contains examples of different prediction mechanisms. Our results highlight the importance of studying different prediction mechanisms in isolation and provide a method for doing this.

Limitations

Our results are limited to auto-regressive models and subject-first template queries. Using the methods described in this paper, PREPMECH datasets can be constructed for other types of LMs, such as encoder-based models, while we leave this for future work.

Moreover, the heuristics filters used for our dataset creation can only reveal the *possibility* of shallow heuristics being used by the LM. We also observe some suspicious samples that go undetected by the filters, indicating that the filters are leaky. Furthermore, we find signs of name based heuristics for non-person subjects for which we have no applicable filters. The detection of these cases would rely on more advanced detection methods and is left for future work. By complementing our dataset creation with knowledge estimations and sampling of synthetic fact tuples, we should avoid most filter failures, while we cannot completely rule out the possibility of there being some problematic samples in PREPMECH.

Even though we partition the PREPMECH samples based on whether the prediction is confident, we find that our results are sensitive to whether we investigate predictions with high or low probabilities from each partition. This indicates room for improvement for our method of detecting confident predictions, for which we already have noted a lack of comprehensive studies of model confidence metrics.

Lastly, we note that multiple interpretability methods would need to be applied to validate the exact underlying computation used by our LMs for the different mechanisms in our taxonomy. When applying only CT, we cannot with certainty distinguish between effects of different prediction mechanisms being used by the LM, as opposed to effects of data-sensitive quality issues of the CT method.

Ethics Statement

Interpretability methods for fact completion situations are not directly associated with any ethical concerns. Neither is the LAMA dataset or synthetic fact tuples used in this work.

References

Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2024. Llms' reading comprehension is affected by parametric knowledge and struggles with hypothetical statements. *arXiv preprint arXiv:2404.06283*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1:1.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 17643–17668. Curran Associates, Inc.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 248–264, Dubrovnik, Croatia. Association for Computational Linguistics.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Faisal Ladhak, Esin Durmus, Mirac Suzgun, Tianyi Zhang, Dan Jurafsky, Kathleen McKeown, and Tatsunori Hashimoto. 2023. When do pre-training biases propagate to downstream tasks? a case study in text summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3206–3219, Dubrovnik, Croatia. Association for Computational Linguistics.

Genglin Liu, Xingyao Wang, Lifan Yuan, Yangyi Chen, and Hao Peng. 2023. Prudent silence or foolish babble? examining large language models' responses to the unknown. *arXiv preprint arXiv:2311.09731*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. 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. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kıcıman, Hamid Palangi, Barun Patra, and Robert West. 2023. A glitch in the matrix? locating and detecting language model grounding with fakepedia. *arXiv preprint arXiv:2312.02073*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.

Gwenyth Portillo Wightman, Alexandra Delucia, and Mark Dredze. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 326–362, Toronto, Canada. Association for Computational Linguistics.

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

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052, Singapore. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vishal Thanvantri Vasudevan, Abhinav Sethy, and Alireza Roshan Ghias. 2019. Towards better confidence estimation for neural models. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7335–7339.

Hiyori Yoshikawa and Naoaki Okazaki. 2023. Selective-LAMA: Selective prediction for confidence-aware evaluation of language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2017–2028, Dubrovnik, Croatia. Association for Computational Linguistics.

A Computational resources

Experiments in this work are done on T4, A40 and A100 NVIDIA GPUs. Models used are GPT-2 XL, which has 1.5B parameters and Llama 2 7B which has 7B parameters.

B Selection process of LAMA relations

The LAMA relations included in our PrepMech dataset have been selected based on the following criteria:

1. We only include relations that have multiple templates for which 1) the object comes last in order to fit the autoregressive setting and 2) the subject comes first in order to simplify the causal reasoning of intervening on the subject;

2. We exclude relations with a lot of overlap between the subject and object and relations for which the answers are highly imbalanced toward only a few alternatives.

This corresponds to the relations P19 *place of birth*, P20 *place of death*, P27 *country of citizenship*, P101 *field of work*, P495 *country of origin*, P740 *location of formation* and P1376 *capital of*.

C ParaRel templates

We use the templates as described in Tables 3 and 4 for the creation of PrepMech queries.

D Creation process for exact fact recall samples

To get queries for which the LM performs exact fact recall, we follow an iterative process as described below:

1. Take all fact tuples from LAMA[6] corresponding to the relations P19 *place of birth*, P20 *place of death*, P27 *country of citizenship*, P101 *field of work*, P495 *country of origin*, P740 *location of formation* and P1376 *capital of*. Our relations selection process is further described in Appendix B.

2. Create paraphrased queries for the fact tuples using the ParaRel templates described in Appendix C (Elazar et al., 2021).

3. Collect LM predictions for the queries. Keep all top 3 tokens and store the corresponding softmaxed logits as metadata. We now have a dataset with query and prediction pairs, plus some additional metadata.

4. Collect estimations of LM knowledge for the prompts following the approach described in Section 2.4.

5. Collect estimations of whether each given prediction is based on surface level artifacts in the query following the approach described in Section 2.4.

6. Label predictions corresponding to trivial tokens and add as metadata to our dataset.

7. Categorize the predictions into "correct" or "incorrect" using the LAMA gold labels. For Llama 2 7B we say that the prediction is correct if it has more than 3 characters and fully matches the start of the gold label. This was necessary since the tokenizer for this model is more prone to split the gold labels into small tokens.

8. Add confidence metadata following the approach described in Section 2.4. Biased predictions are separated from predictions without any potential bias before we count the number of consistent predictions. A biased prediction that is consistent with an unbiased prediction does not count for the unbiased prediction and vice versa.

9. Extract samples that should correspond to exact fact recall from the dataset above. Exact fact recall samples should correspond to predictions that are 1) not labelled as corresponding to any bias, 2) correct, 3) corresponding to a fact known by the LM and 4) confident. It is not a problem if this excludes samples corresponding to exact fact recall, as we are only interested in precision and not recall for these samples.

E Creation process for heuristics recall samples

The heuristics recall samples are constructed to align with the format of the exact fact recall samples. Therefore, we create this partition based on the same relations as used in Appendix D. To obtain the relevant data and labels, we perform the following steps:

[6] https://github.com/facebookresearch/LAMA

1. Identify subject type distributions for the selected relations.
2. Generate subjects of the required types using https://www.fantasynamegenerators.com.
For relations P19, P20, P27 and P101 the only allowed subject type is person, so the generated subjects are human names. For P1376 the subject type is city, and the generated data is city names. Relations P495 and P740 have a variety of allowed subject types. For these, we produce a distribution over the original LAMA data and match that as closely as possible with the available name generators.
3. Perform de-duplication and check against Wikidata that no subject corresponds to a real entity. The Wikidata check is performed on a label level, since the generated names are pure strings. This limits our ability to check for a subject's existence, as we can only find exact matches.
4. Generate prompts corresponding to each relation by applying the ParaRel templates described in Appendix C to the synthetic subjects.
5. Collect LM predictions by extracting the top 3 tokens.
6. Identify non-trivial answers. This is carried out by querying the Wikidata database and suffers from the same limitations as discussed above – we are limited to exact matching strings. This can result in additional challenges due to e.g. tokenization truncating the full entity.
7. Filter on confidence. We only keep predictions marked as confident and apply the same definition of confidence as described in Section 2.4.
8. Add metadata on: prompt bias, name bias, subject-object string overlap. The distribution of these flags is presented in Appendix J. The samples for which a single type of bias is identified form our heuristics recall samples.

F Creation process for generic language modeling samples

Data is sampled from Wikipedia extraction `20220301.en` from HuggingFace at https://huggingface.co/datasets/wikipedia. We randomly select an entry from the data. For each, we select a single sentence from the page that begins with any part of the title name (i.e. it could be the surname, if the subject is a person). If the sentence is longer than 10 words, we cap it. We do not select sentences if they are: 1) shorter than 5 words, 2) with more than 3 capitalized words (likely to be section headings), and 3) whose natural continuation begins with a capital or number (indicating this could be an entity and thus potentially fact completion). We repeat this until we have 1000 datapoints (for 1000 unique entries in the data). For CT experiments, we trace the next token, freely predicted by the model.

G Detection filters for heuristics

Our detection of heuristics is based on model predictions for prompts expressing only a part of the requested fact. For person name bias, we query with the following prompts: "[X] is a common name in the following city:" and "[X] is a common name in the following country:". Where "[X]" is replaced with the subject name to check for bias. If any of the top 10 token predictions for these queries matches the model prediction for the full fact query, we mark that (*query*, *prediction*) pair as corresponding to person name bias. We can detect person name bias for relations P19, P20, P27, used in PrepMech and additionally for P103 and P1412, present in CounterFact.

For the detection of prompt bias we use the original prompt templates as defined by ParaRel and replace the subject placeholder with generic substitutions. We use the substitutions described in Table 5 for each relation. We also remedy basic capitalization and grammar errors that might surface from this automated prompt creation. An example of a prompt for detecting prompt bias for "Tokyo is the capital city of [Y]" is "It is the captial city of [Y]". If the top prediction for the former query is found among the top 10 token predictions for the latter query, the former query and corresponding prediction is marked as based on prompt bias.

H Analysis of the exact fact recall samples in PrepMech

The composition of the relations that make up the exact fact recall queries in PrepMech is shown in Table 6.

I Analysis of the heuristics recall samples in PREPMECH

Our final heuristics recall set, described in Section 3.2, contains 1,771 examples where no bias was identified. This can be counter intuitive, as we do not expect the model to be able to make confident prediction when it has no bias to guide it. We therefore perform a deeper analysis of these samples.

These include 6 instances that identify the location of formation (P740) of "Oasis of Prejudice" as "London" (not identified as prompt bias, since the prompt bias check produces mostly years, indicating time to be the more natural interpretation of the queries). Two examples from P101 (field of work) show the model potentially ignoring part of the query, by connecting "Nina Schopenhauer" with "philosophy" and "Roch Chagnon" with "anthropology" (in total 9 rephrased samples). Another 23 examples of relation P495 show association of 5 fictional entities with Japan (3 of these contain the word "Berserk" – a possible conflating pattern with the manga of the same name). Further 790 examples come from relations P19 (born in) and P27 (citizen of). Some of these could be examples of a stronger association overwriting the expressed tuple (e.g. "Adolphe Trudeau" born in "Quebec"), others may point to weaknesses of our name bias detection method. Finally, the most represented relation is P1376 with 938 examples. This relation does not lend itself to our subject name bias filter, however, we suspect a linguistic correlation between city names and countries may exist and those surface level signals can potentially explain some of the predictions.

This analysis confirms our concerns related to the coverage of the implemented heuristics recall filters. Evidently, there are some heuristics that go undetected by our filters. This highlights the strength of our method based on sampling synthetic data for the heuristics recall detection and filtering for popularity for the exact fact recall detection.

J Bias and predicate distribution for synthetic data

Table 7 shows the distribution of bias in the synthetic data. Most samples have name bias detected. Table 8 shows the relation distribution of samples that have at least one confident non-trivial prediction. The most represented predicate is P27 *citizen-of*. This is inline with the name bias prevalence that we see.

K Examples from PREPMECH

Here, we include a few examples to illustrate the content of PREPMECH for different prediction mechanisms. See Tables 9 to 12.

Relation	Template
P19	[X] was born in [Y]
	[X] is originally from [Y]
	[X] was originally from [Y]
	[X] originated from [Y]
	[X] originates from [Y]
P20	[X] died in [Y]
	[X] died at [Y]
	[X] passed away in [Y]
	[X] passed away at [Y]
	[X] expired at [Y]
	[X] lost their life at [Y]
	[X]'s life ended in [Y]
	[X] succumbed at [Y]
P27	[X] is a citizen of [Y]
	[X], a citizen of [Y]
	[X], who is a citizen of [Y]
	[X] holds a citizenship of [Y]
	[X] has a citizenship of [Y]
	[X], who holds a citizenship of [Y]
	[X], who has a citizenship of [Y]
P101	[X] works in the field of [Y]
	[X] specializes in [Y]
	The expertise of [X] is [Y]
	The domain of activity of [X] is [Y]
	The domain of work of [X] is [Y]
	[X]'s area of work is [Y]
	[X]'s domain of work is [Y]
	[X]'s domain of activity is [Y]
	[X]'s expertise is [Y]
	[X] works in the area of [Y]
P495	[X] was created in [Y]
	[X], that was created in [Y]
	[X], created in [Y]
	[X], that originated in [Y]
	[X] originated in [Y]
	[X] formed in [Y]
	[X] was formed in [Y]
	[X], that was formed in [Y]
	[X] was formulated in [Y]
	[X], formulated in [Y]
	[X], that was formulated in [Y]
	[X] was from [Y]
	[X], from [Y]
	[X], that was developed in [Y]
	[X] was developed in [Y]
	[X], developed in [Y]

Table 3: ParaRel templates used for the relations P19-P495 in our dataset creation.

Relation	Template
P740	[X] was founded in [Y]
	[X], founded in [Y]
	[X] that was founded in [Y]
	[X], that was started in [Y]
	[X] started in [Y]
	[X] was started in [Y]
	[X], that was created in [Y]
	[X], created in [Y]
	[X] was created in [Y]
	[X], that originated in [Y]
	[X] originated in [Y]
	[X] formed in [Y]
	[X] was formed in [Y]
	[X], that was formed in [Y]
P1376	[X] is the capital of [Y]
	[X] is the capital city of [Y]
	[X], the capital of [Y]
	[X], the capital city of [Y]
	[X], that is the capital of [Y]
	[X], that is the capital city of [Y]

Table 4: ParaRel templates used for the relations P740 and P1376 in our dataset creation.

Relation	Subject substitutions
P19	[He, She]
P20	[He, She]
P27	[He, She]
P101	[He, She]
P495	[It]
P740	[It, The organisation]
P1376	[It, The city]

Table 5: Subject substitutions used for constructing prompts to detect prompt bias.

Relation	#unique tuples
P19	0
P20	0
P27	77
P101	18
P495	406
P740	95
P1376	726

Table 6: The number of unique tuples corresponding to each relation of the exact fact recall samples in PREP-MECH.

prompt bias	string match	name bias	#samples
FALSE	FALSE	FALSE	1771
		TRUE	7066
	TRUE	FALSE	34
		TRUE	8
TRUE	FALSE	FALSE	1252
		TRUE	4775
	TRUE	FALSE	6
		TRUE	7

Table 7: Distribution of detected bias in confident non-trivial predictions in the synthetic data of the PREPMECH dataset.

Relation	# samples
P101	9
P1376	1754
P19	2674
P20	5
P27	10436
P495	33
P740	8

Table 8: Distribution of relations in the synthetic data of the PREPMECH dataset that have a confident non-trivial prediction.

Model	Query	Prediction	Subject popularity	Gold label
GPT-2 XL	Thomas Ong is a citizen of	Singapore	1418	Singapore
	Shibuya-kei, that was created in	Japan	5933	Japan
	Palermo is the capital of	Sicily	34273	Sicily
Llama 2 7B	Disco Biscuits was created in	Philadelphia	3719	Philadelphia
	Don Broco, that was started in	Bed	6984	Bedford
	Nikephoros III Botaneiates passed away in	Constantin	1859	Constantinople

Table 9: (*query*, *prediction*) exact fact recall samples from PREPMECH for GPT-2 XL and Llama 2 7B.

Model	Query	Prediction	Rank	Gold label
GPT-2 XL	Sonar Kollektiv originated in	Russia	1	Berlin
	Haydn Bendall is originally from	England	2	Essex
	Joseph Clay was originally from	Ohio	2	Philadelphia
Llama 2 7B	Jean Trembley originated from	France	2	Geneva
	Dansez pentru tine, that originated in	France	2	Romania
	Milton Wright is originally from	Chicago	2	Georgia

Table 10: (*query*, *prediction*) random guesswork samples from PREPMECH for GPT-2 XL and Llama 2 7B.

Model	Query	Prediction	Bias
GPT-2 XL	Hirashima Hideyoshi, who has a citizenship of	Japan	name
	Balo Windhair has a citizenship of	Canada	prompt
	Olre Hellspirit was originally from	Hell	lexical
Llama 2 7B	Ha Songmin, who has a citizenship of	South (Korea)	name
	Wanda Hagel holds a citizenship of	Canada	prompt
	Limanaga, the capital city of	Lim	lexical

Table 11: (*query*, *prediction*) heuristics recall samples from PREPMECH for GPT-2 XL and Llama 2 7B.

Model	Query	Prediction	Gold label
GPT-2 XL	Dexmedetomidine is notable for its ability to provide sedation	and	without
	Solomon also defended the network's choice of games to	air	broadcast
	Walker added an immense amount of material to the	book	collections
Llama 2 7B	Dexmedetomidine is notable for its ability to provide sedation	and	without
	Solomon also defended the network's choice of games to	air	broadcast
	Walker added an immense amount of material to the	original	collections

Table 12: (*query*, *prediction*) generic language samples from PREPMECH for GPT-2 XL and Llama 2 7B.

L Prediction mechanisms represented by CounterFact

Here, we include additional information related to the study of prediction mechanisms used by GPT-2 XL when evaluated on known CounterFact samples.

L.1 Surface level artifacts

Examples of predictions marked for bias can be found in Table 13.

L.2 LM knowledge

The popularity score distribution for the known CounterFact samples can be found in Table 14.

It is highly unlikely that fact tuples corresponding to subjects with popularity scores below 100 have been stored by the LM. 17 of these 61 samples correspond to either prompt or person name bias. Closer inspection of the 44 samples not marked for bias reveal 4 potential issues with the case sensitivity of the Wikipedia pageview API for the subjects "macOS", "iPhone 3GS", "iTunes" and "iPhone" that lead to incorrect popularity score estimations.

Another 12 samples correspond to queries about the continent of which a subject is a part of for subjects that contain the word "Glacier", where the correct answer is "Antarctica". Our name bias filter cannot detect these cases as it is limited to person names. We observe additional samples among the 61 low popularity samples with similar issues, where the subject might have a very french sounding name like for the query "Galerie des Machines, in the heart of [Paris]".

Samples with popularity scores between (100, 1000] are also less likely to have been memorized. For this subset, 155 samples have been marked for prompt or person name bias. For the remaining 149 samples we again find potential issues with name bias that have gone undetected, such as "Si la vie est cadeau is written in [French]".

L.3 Total effects

We measure the total effect of perturbing the subject on the probability of the output prediction. This provides an alternative way of checking for signs of lack of exact fact recall. The method was introduced by Meng et al. (2022) and used to find model states important for the model prediction. By adding noise to the word embeddings corresponding to the subject of the query, the subject is perturbed. The idea is that the perturbation of the query makes the model incapable of performing the necessary recall of factual associations that resulted in the original prediction, thus lowering the model probability for the original prediction. We hypothesize that samples for which the added perturbation does not sufficiently lower the corresponding prediction probability are less likely to correspond to exact fact recall.

Method The total effect is measured as $\mathrm{TE}(o) = P_{\text{clean}}(o) - P_{\text{noised}}(o)$, where $P_{\text{clean}}(o)$ denotes the probability of emitting token o for a clean run and $P_{\text{noised}}(o)$ denotes the probability of emitting token o when the subject has been perturbed. For all our investigations, o is given by the prediction corresponding to the query stored in the dataset. We note that negative total effects imply that the perturbation of the subject increased the probability of the original prediction and that low positive effects potentially indicate that perturbing the subject had a small effect on the model prediction.

Similarly to Meng et al. (2022) we perturb the subject embeddings with noise $\epsilon \sim \mathrm{N}(0, \nu)$ where ν is set to be 3 times larger than the empirical standard deviation of all embeddings corresponding to the subjects of the dataset. We measure total effects for the known CounterFact samples as the average total effect of 10 runs with perturbed subjects.

TE results For the 1209 known CounterFact samples we find 22 samples with negative total effects, i.e. perturbing the subject increased the prediction probability, of which 18 potentially correspond to prompt bias and 2 to name bias. Inspection of the samples marked for prompt bias reveal prompt patterns such as "In [X], the language spoken is a mixture of" where the corresponding prediction is "English" or "German". Another pattern we detect is "[X] is affiliated with the religion of" for which the prediction always is "Islam". We hypothesize that some prompts reveal the correct prediction even when the subject is occluded, resulting in negative TE values.

Deeper study of TE results A deeper study of the TE values reveal an additional 37 samples for which the perturbation of the query subject decreased the original probability by less than 40%. For some of these samples we identify queries that potentially reveal the correct prediction even when the subject is perturbed. Two identified samples are "[X] professionally plays the sport of ice [hockey]" or "[X]'s expertise is in the field of quan-

Query	Prediction	Bias type
MacApp, a product created by	Apple	Prompt
Giuseppe Angeli, who has a citizenship of	Italy	Person name
The original language of La Fontaine's Fables is a mixture of	French	Prompt

Table 13: Examples of queries and predictions from the known CounterFact dataset that potentially correspond to bias. The predictions and analysis has been performed for GPT-2 XL.

Popularity score	# of samples
$(0, 100]$	61
$(100, 1000]$	304
$(1000, 10000]$	379
$(10000, 1176235]$	437

Table 14: The popularity scores for the known CounterFact samples. The maximum popularity score measured was 1,176,235.

tum [physics]". Prompt bias was detected for all of these queries. We measure a spearman correlation of -0.41 between normalized TE (Equation (3)) and the binary prompt bias metric over all known CounterFact samples. It is clear that the effect of perturbing the subject is smaller when the prediction is likely based on prompt bias, versus when it is not.

$$\mathrm{TE}_{\mathrm{norm}}(o) = \frac{P_{\mathrm{clean}}(o) - P_{\mathrm{noised}}(o)}{P_{\mathrm{clean}}(o)} \quad (3)$$

L.4 Negated queries

We identify a total of 8 samples in the dataset that contain the word "not" in the query. Two examples are "The language used by Louis Bonaparte is not the language of the [French]" or "The expertise of medical association is not in the field of [medicine]". These samples are problematic as they are marked as correct since they contain the correct label, while they express the opposite of the fact represented by the data sample. This problem is a consequence of the sampling technique used by Meng et al. (2022) in letting the LM generate a fluent continuation to a given query before making the prediction for the missing object. For the majority of the known CounterFact samples this leads to more fluent queries for which the LM might work better, but for some samples it results in reversed or revealing prompts.

M Additional results from the CT sensitivity analysis

This section contains additional results from the analysis in Section 4.

M.1 Llama 2 7B results

The results in Figure 3 correspond to the results in Figure 2 but here for Llama 2 7B instead of GPT-2 XL. We find that the Llama results essentially support the same conclusions as the results for GPT-2 XL.

M.2 Low-probability split

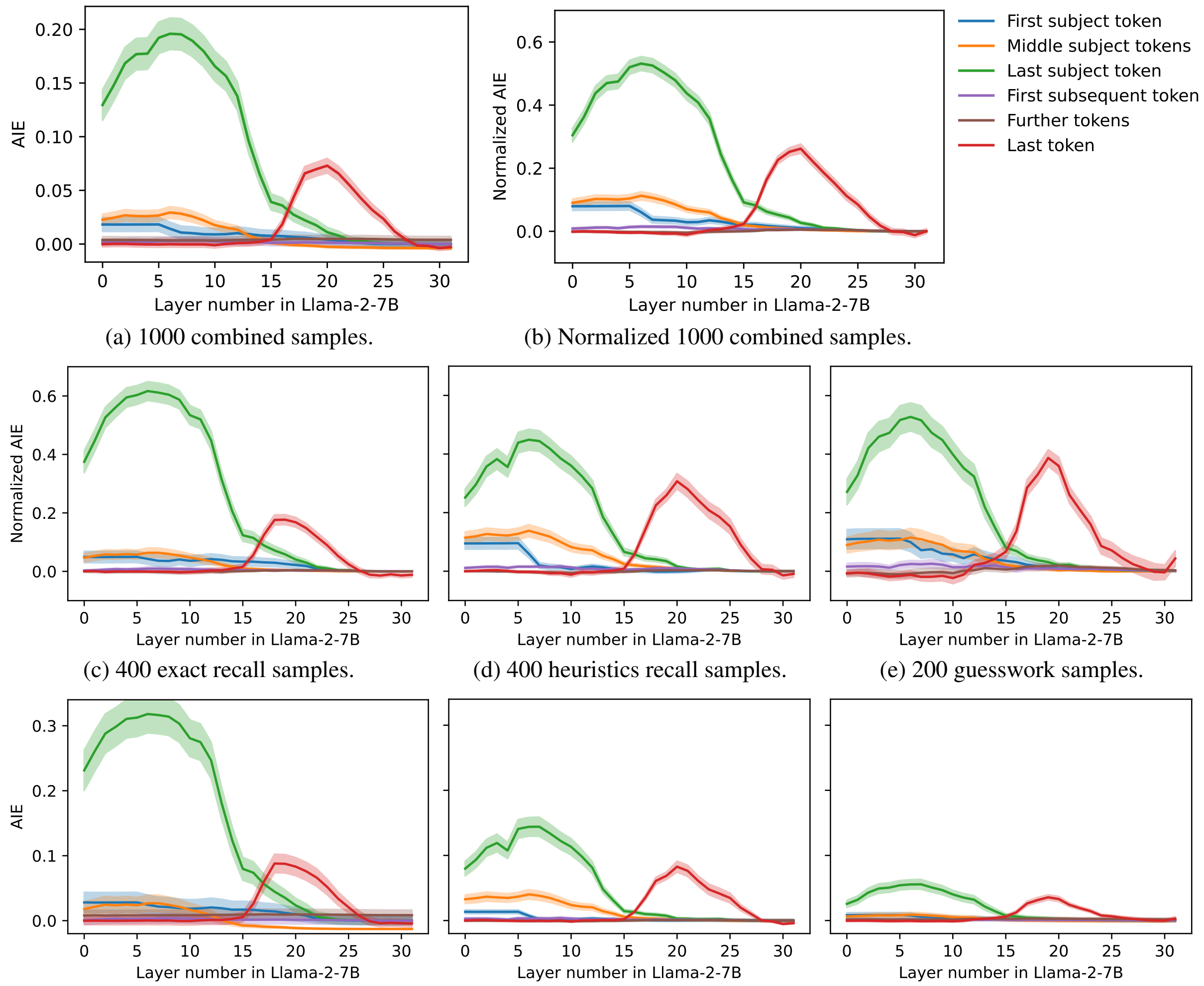
The results in Figures 2 and 3 correspond to a sample of top-ranked prediction probabilities. The results in Figures 4 and 5 correspond to a sample of bottom-ranked prediction probabilities. We observe qualitative differences between the two figure pairs, where bottom-ranked probability set corresponds to larger effects for the last token state.

M.3 Per prediction mechanism

CT results for 1000 samples from PrepMech designed to exemplify each of our identified prediction mechanisms can be found in Figures 6 and 7. We conclude that the subsets used for Figures 2 and 3 are representative of these larger sets. Moreover, we observe that the results for the generic language modelling mechanism in Figure 7 do not indicate a decisive role for the last subject token MLP state at middle layers.

M.4 Deeper study of heuristics recall

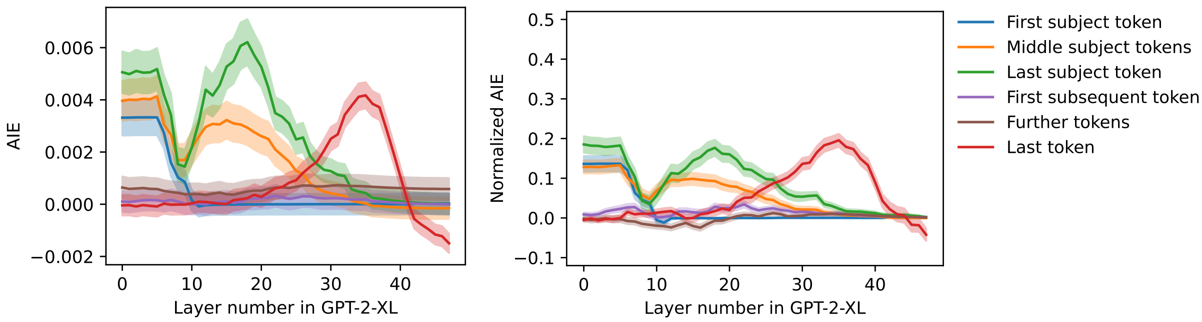
We analyze the CT results of each of the main heuristics recall categories, prompt bias and person name bias, in separation for GPT-2 XL and Llama 2 7B. The corresponding results can be found in Figure 8. These results suggest a higher importance of the last token state, compared to the last subject token state, for the prompt bias subset compared to the person name bias subset. Potentially, it makes sense that prompt biased predictions that should be



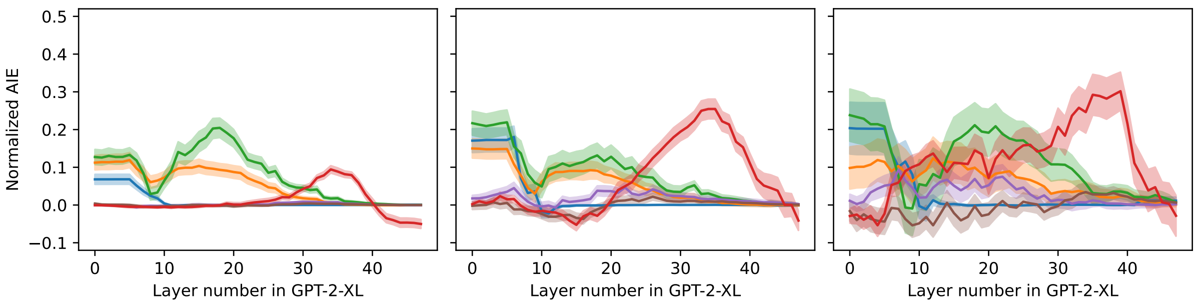
(a) 1000 combined samples.
(b) Normalized 1000 combined samples.
(c) 400 exact recall samples.
(d) 400 heuristics recall samples.
(e) 200 guesswork samples.
(f) 400 exact recall samples.
(g) 400 heuristics recall samples.
(h) 200 guesswork samples.

Figure 3: CT results on 1000 samples from PrepMech of which 400 samples correspond to exact fact recall, 400 to heuristics recall and 200 to guesswork. These are the results for Llama 2 7B.

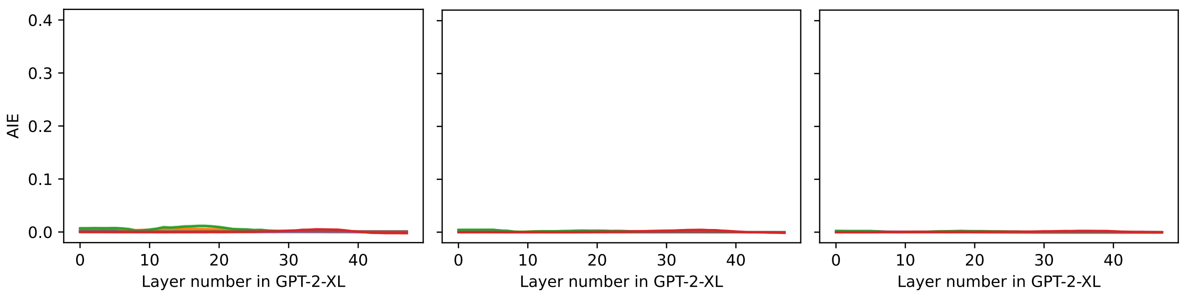
less sensitive to subject information attribute less importance to states corresponding to the subject.



(a) 1000 combined samples. (b) Normalized 1000 combined samples.

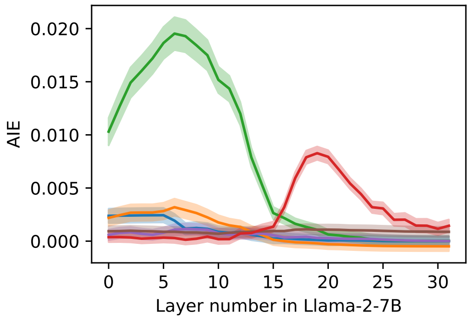


(c) 400 exact recall samples. (d) 400 heuristics recall samples. (e) 200 guesswork samples.

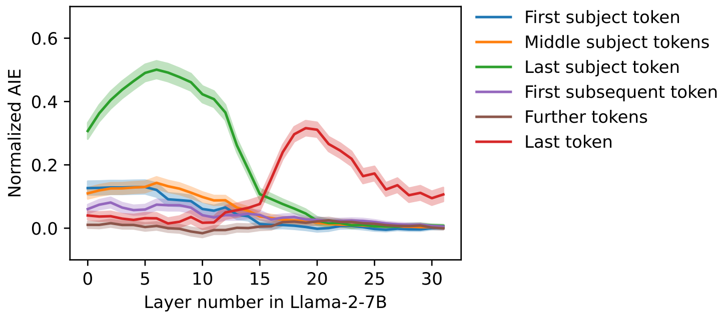


(f) 400 exact recall samples. (g) 400 heuristics recall samples. (h) 200 guesswork samples.

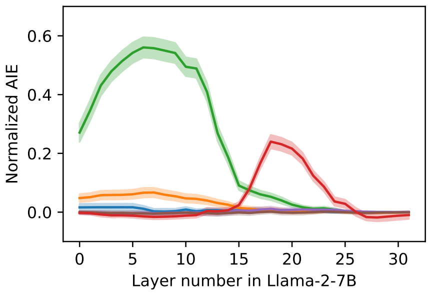
Figure 4: CT results on 1000 low-probability samples from PrepMech of which 400 samples correspond to exact fact recall, 400 to heuristics recall and 200 to guesswork. These are the results for GPT-2 XL.



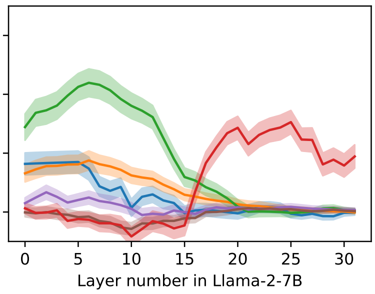
(a) 1000 combined samples.



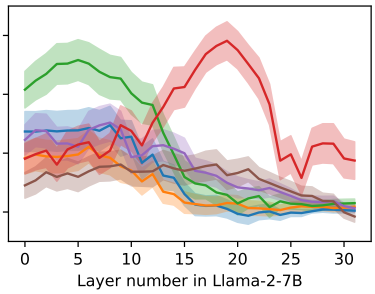
(b) Normalized 1000 combined samples.



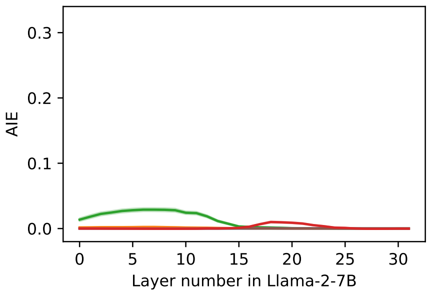
(c) 400 exact recall samples.



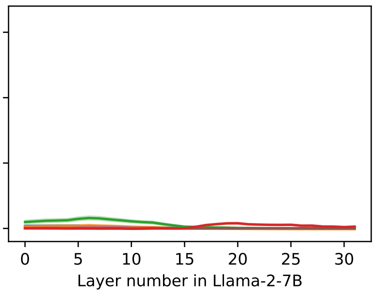
(d) 400 heuristics recall samples.



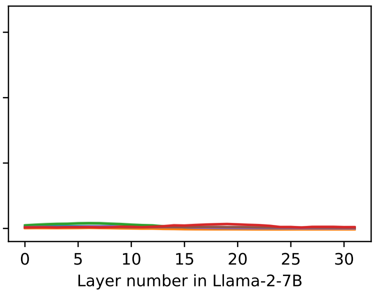
(e) 200 guesswork samples.



(f) 400 exact recall samples.

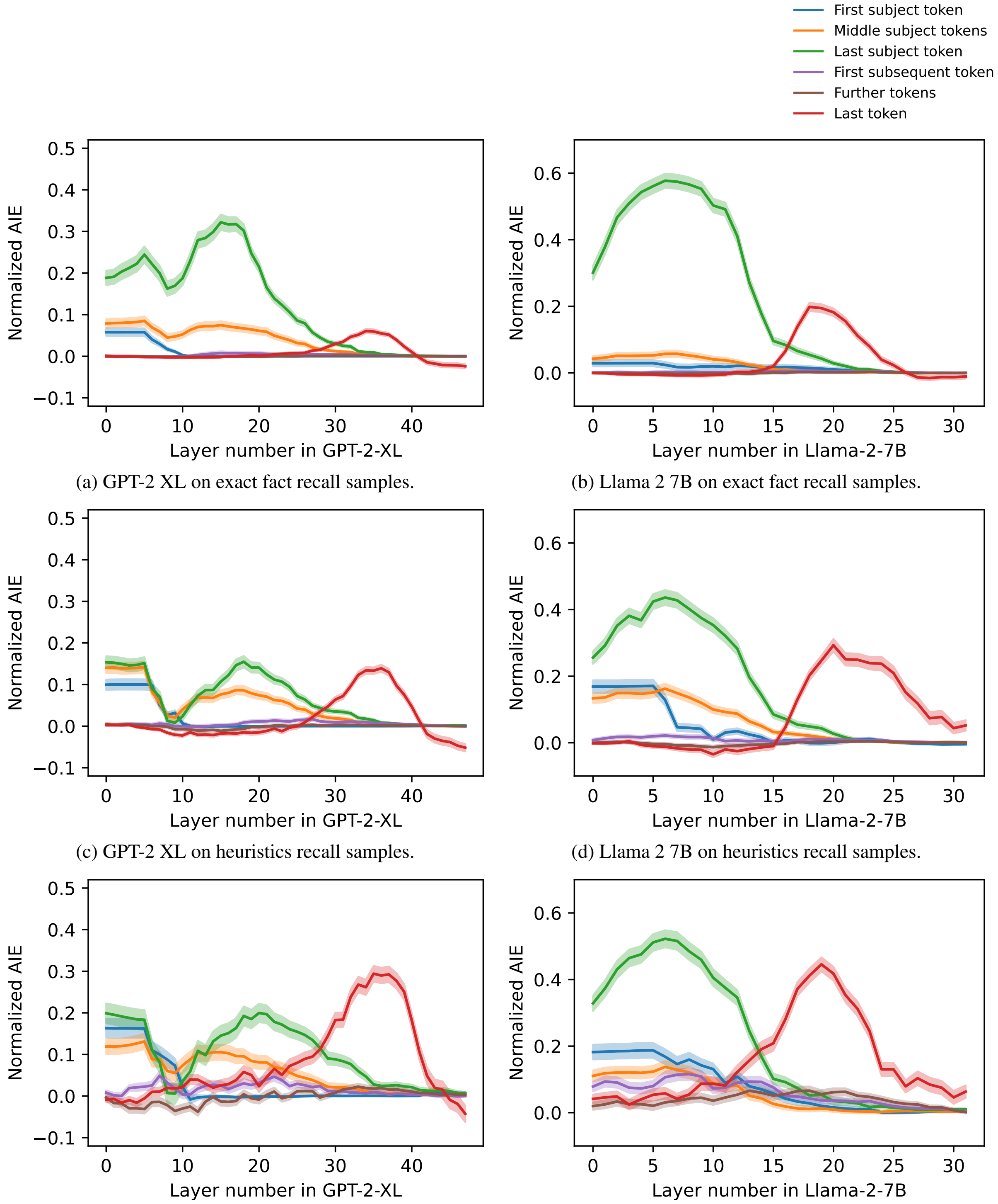


(g) 400 heuristics recall samples.



(h) 200 guesswork samples.

Figure 5: CT results on 1000 low-probability samples from PrepMech of which 400 samples correspond to exact fact recall, 400 to heuristics recall and 200 to guesswork. These are the results for Llama 2 7B.



(a) GPT-2 XL on exact fact recall samples.

(b) Llama 2 7B on exact fact recall samples.

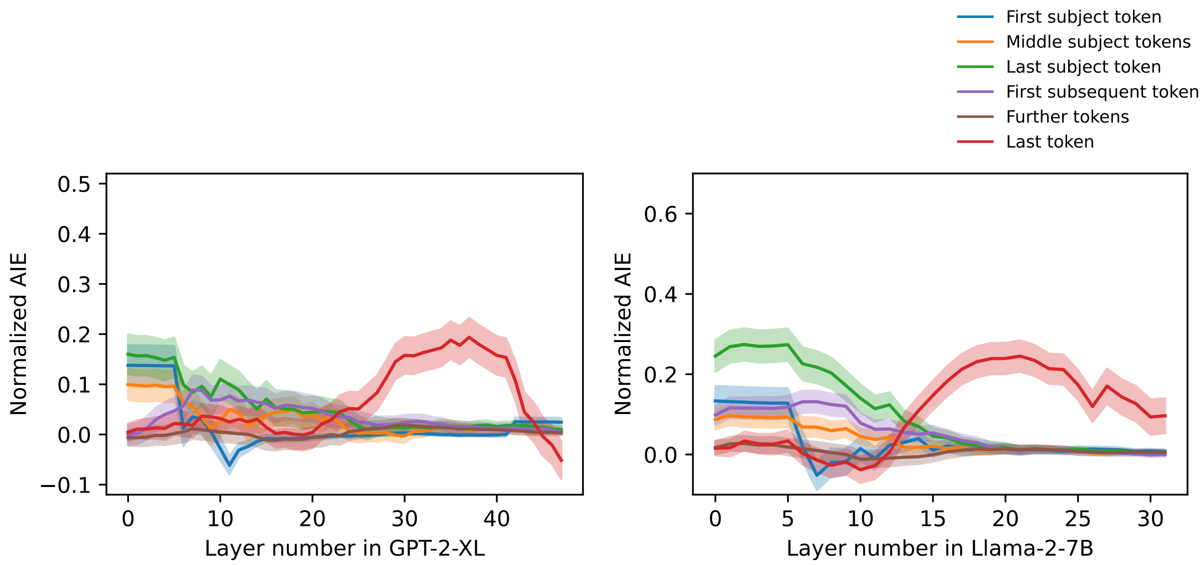
(c) GPT-2 XL on heuristics recall samples.

(d) Llama 2 7B on heuristics recall samples.

(e) GPT-2 XL on guesswork samples.

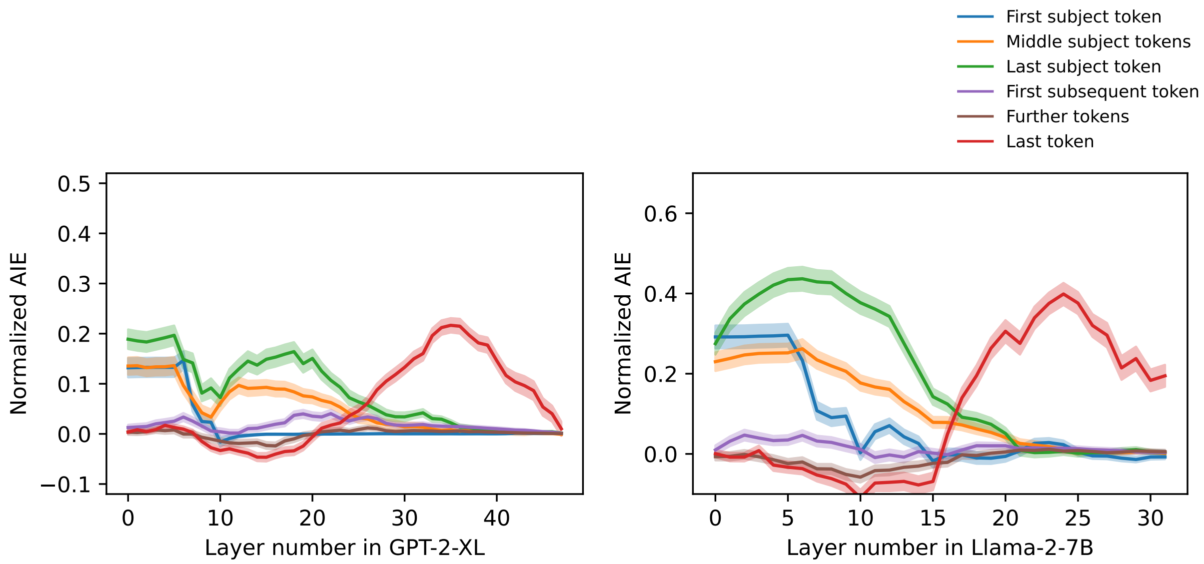
(f) Llama 2 7B on guesswork samples.

Figure 6: Normalized CT results for 1000 samples from PrepMech designed to exemplify each of the prediction mechanisms exact fact recall, heuristics recall and guesswork. Results are reported for both GPT-2 XL and Llama 2 7B.

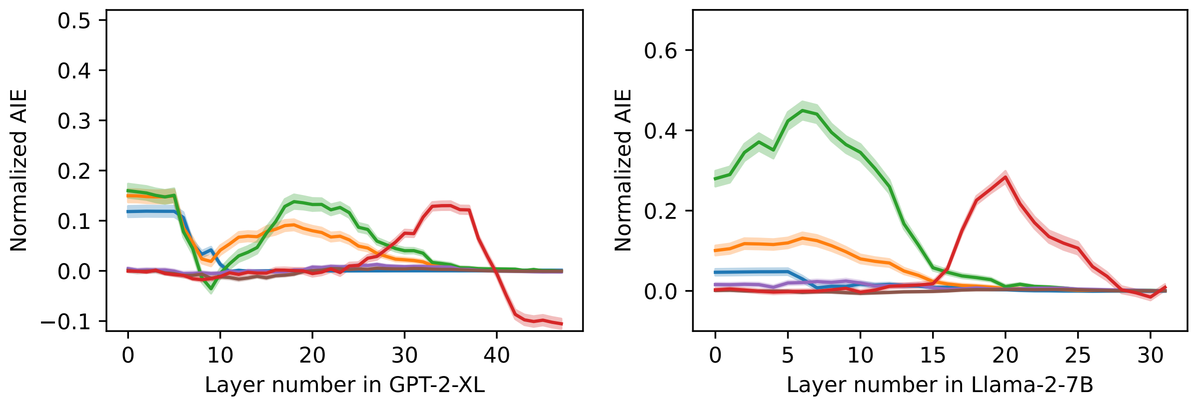


(a) GPT-2 XL on generic language modelling samples. (b) Llama 2 7B on generic language modelling samples.

Figure 7: Normalized CT results for 1000 samples from PREPMECH designed to exemplify generic language modelling. Results are reported for both GPT-2 XL and Llama 2 7B.



(a) Prompt bias for GPT-2 XL. (b) Prompt bias for Llama 2 7B.



(c) Person name bias for GPT-2 XL. (d) Person name bias for Llama 2 7B.

Figure 8: Normalized CT results for sets of 1000 samples designed to exemplify each of the two main categories of the heuristics recall mechanism.