# Probing Language Models on Their Knowledge Source

Zineddine Tighidet, Andrea Mogini, Jiali Mei, Benjamin Piwowarski, Patrick Gallinari

# Probing Language Models on Their Knowledge Source

**Zineddine Tighidet[1,2], Andrea Mogini[1], Jiali Mei[1], Benjamin Piwowarski[2], Patrick Gallinari[2,3]**

[1]BNP Paribas, Paris, France
[2]Sorbonne Université, CNRS, ISIR, F-75005 Paris, France
[3]Criteo AI Lab, Paris, France

*firstname.lastname@{isir.upmc.fr, bnpparibas.com}*

## Abstract

Large Language Models (LLMs) often encounter conflicts between their learned, internal (parametric knowledge, PK) and external knowledge provided during inference (contextual knowledge, CK). Understanding how LLMs models prioritize one knowledge source over the other remains a challenge. In this paper, we propose a novel probing framework to explore the mechanisms governing the selection between PK and CK in LLMs. Using controlled prompts designed to contradict the model's PK, we demonstrate that specific model activations are indicative of the knowledge source employed. We evaluate this framework on various LLMs of different sizes and demonstrate that mid-layer activations, particularly those related to relations in the input, are crucial in predicting knowledge source selection, paving the way for more reliable models capable of handling knowledge conflicts effectively.

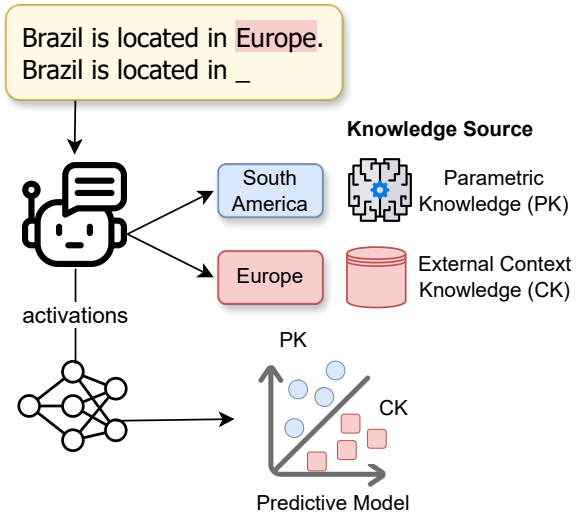


Figure 1: Illustration of our method for probing knowledge sources in LLMs. We present the model with a prompt containing contradictory information to its learned knowledge to test whether it uses parametric knowledge (PK) or contextual knowledge (CK). The resulting activations are used to train a classifier to distinguish between PK and CK.

## 1 Introduction

Large Language Models (LLMs) have demonstrated remarkable proficiency in memorizing and retrieving massive amounts of information. Despite these strengths, LLMs often struggle when exposed to novel information not seen during training (Ovadia et al., 2019) or when there is a conflict between their **parametric knowledge (PK)** and the **context knowledge (CK)** provided at inference (Xie et al., 2024). Such discrepancies can lead to erroneous outputs, a phenomenon that remains a significant challenge in LLMs applications (Ji et al., 2023). While several approaches, such as reinforcement learning and retrieval-augmented generation, have been proposed to mitigate these issues (Ziegler et al., 2020; Lewis et al., 2021), the mechanisms by which LLMs select and prioritize knowledge sources are not well understood, suggesting a gap in current research methodologies.

This paper explores the internal dynamics of LLMs, and more precisely decoder-only layers, focusing on their decision-making processes regarding the use of CK versus PK. By prompting the LLM in a way that contradicts its PK, we probe the model's knowledge-sourcing behaviors. By training a linear classifier on model activations, our experiments reveal that certain activations correlate with determining whether context or parametric knowledge predominates in the generated outputs.

In this paper, we make the following key findings and contributions:

- We define a framework that characterizes the source of knowledge used by the model to generate its outputs – Sections 3 and 4. To facilitate further research and validation of our findings, we make our framework publicly

available on GitHub[1].

- Specific activations are indicative of the knowledge source: by applying our framework to LLMs of different sizes, we establish that specific activations correlate with the model's use of contextual or parametric knowledge.

## 2 Related Work

The understanding of the mechanisms and knowledge localization within transformers has progressed through various studies. On the one hand, some work investigated the PK-based outputs (factual setting) (Meng et al., 2023; Geva et al., 2021, 2023; AlKhamissi et al., 2022; Heinzerling and Inui, 2021). These works hypothesized that LLMs store parametric information in the Multi-Layer Perceptron (MLP), which acts as a key-value memory, subsequently accessed by the Multi-Head Self-Attention (MHSA) mechanisms. On the other hand, other studies focused on the CK-based outputs and concluded that processing CK, as opposed to PK, is not specifically localized in the LLM's parameters (Monea et al., 2024).

Yu et al. (2023) employed an attribution method (Wang et al., 2022; Belrose et al., 2023) to identify the most influential attention heads responsible for generating PK and CK outputs, and subsequently adjusted the weights of these heads to modify the source of knowledge. Their work however focuses exclusively on knowledge specific to capital cities and relies on causal tracing, which is costly to compute.

In contrast, our work utilizes a probing approach that uses a classifier on the LLM activations to identify the source of knowledge, leveraging the insights from previous research on the respective roles of MLPs and MHSAs in the inference process. We extend the scope of Yu et al. (2023) by incorporating a dataset with a broader range of knowledge categories (`ParaRel` (Elazar et al., 2021)), moving beyond just capital cities.

## 3 Methodology

We aim to show that specific activations correlate with the used knowledge source, parametric or context knowledge. In order to probe LLMs, we construct prompts that are composed of inputs representing information about a subject $s$ that contradicts what the model learned during training, followed by a query about the same subject (see Figure 1). If the model answers according to the prompt, then it uses context knowledge. On the other hand, if the model answers according to what it learned, then it is using its parametric knowledge. In the following two sections, we define more formally PK and CK.

### 3.1 Parametric Knowledge (PK)

We consider the parametric knowledge (PK) to be the information that the model learned during training. More specifically, we restrict this PK by using a knowledge base $KB = \{(s, r, o)\}$, i.e. a set of (subject, relation, object) triplets from the ParaRel dataset (Elazar et al., 2021). We then define $PK$ to be the set of objects that are generated by a LLM:

$$PK = \{(s, r, o^{'}) \mid \exists o \text{ s.t. } (s, r, o) \in KB \wedge o^{'} = G_\theta(q(s, r))\} \quad (1)$$

where $G_\theta$ is an LLM; $q(s, r)$ is a prompt in natural language corresponding to a subject-relation pair $(s, r)$; $o^{'}$ is the output of $G_\theta$ given the query prompt (e.g. *"Brazil is located in the continent of _"*).

Note that we use this method to define PK because we do not have access to the training data of LLMs in general, and, more importantly, we are interested in what the LLM infers by itself. If $o = G_\theta(q(s, r))$, that is, the object $o$ was generated by the model after providing an input query $q(s, r)$, we can conclude that the model learned to associate the object $o$ with the subject $s$ with the relation $r$ during training. Note also that, unlike previous work (Meng et al., 2023; Yu et al., 2023), even when $o$ is factually incorrect (e.g. *"Paris is the capital of Italy"*), we still consider it in our study as our only interest is the parametric knowledge and not the external world factual truth[2].

#### 3.1.1 Knowledge Base (ParaRel)

We extend the ParaRel dataset (Elazar et al., 2021) for constructing a parametric knowledge base. ParaRel dataset consists of triplets, each composed of a subject, a relation, and an object. Table 1 illustrates a sample of the raw ParaRel dataset.

While the majority of the triplets adhere to the subject-relation-object structure, some deviate

[1]Link to the code and dataset: `https://github.com/Zineddine-Tighidet/knowledge-probing-framework`

[2]This behavior happens when the subjects are unpopular and the LLM was not trained on enough examples. We discuss this further in Section 6.

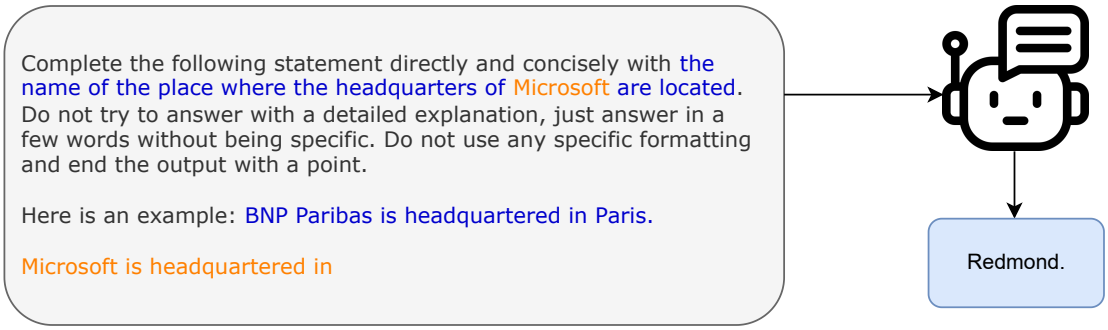


Figure 2: Example of the template used to generate the parametric knowledge dataset. The blue text is proper to the relation and the orange is specific to a subject-relation example in the ParaRel dataset (Elazar et al., 2021).

from this format. To ensure consistency, a preprocessing step was applied on the raw ParaRel dataset using Mistral-Large[3]. Specifically, the goal was to transform triplets where the subject precedes the relation (e.g., *"The official language of France is French."*) into triplets where the subject is placed directly before the relation (e.g., *"France's official language is French."*). We selected Mistral-Large because it is open-weight, enabling reproducibility, and its capabilities are very close to those of GPT-4.

#### 3.1.2 Parametric Knowledge Query Format

To guide the studied LLMs towards generating parametric knowledge objects that are coherent with the relation and to help specifying the type of object that is expected when there are multiple possible answers (for example in *"Napoleon passed away in"* the LLM can generate the place of death *"Longwood"* or the year of death *"1821"*) we propose to use a template prompt that is illustrated in Figure 2. The prompt specifies the requested type of object with a brief description as well as an example (one-shot learning) to help the LLM understand the kind of object that is intended (illustrated in blue in Figure 2). The description and example were manually created for each relation. The prompt also tries to guide the LLM towards generating a concise output as these models tend to give a long explanation that is irrelevant in our study (e.g. *Amazon is headquartered in the city of Seattle where Starbucks is also headquartered...*).

#### 3.1.3 Subject/Object Bias

The subject can sometimes provide relevant information about the object which can bias our definition of parametric knowledge (e.g. *Princeton University Press is located in Princeton.* or *Niger shares the border with Nigeria*). To avoid this, we removed examples where the subject is similar to the object, utilizing the Jaro-Winkler string distance (Jaro–Winkler) with a threshold empirically fixed at 0.8. This method is advantageous for our dataset, as it assigns closer distances to subjects with the same prefix as the objects, which is common in cases like *"Croatia's official language is Croatian"* where *"Croatia"* and *"Croatian"* have the same prefix.

### 3.2 Context Knowledge (CK)

In our framework, we perturb the LLM by providing a CK that contradicts the PK, which we name *counter-PK* and denote $\overline{PK}$. It is challenging to test what the model does not know (Yin et al., 2023). One way to build these inputs is to contradict what the model learned during training by taking $(s, r, o) \in PK$ and replacing $o$ with another object $\bar{o} \in O_r$ that shares the same relation $r$ to keep semantic consistency (e.g. *"Elvis Presley is a citizen of Japan"*, here we replaced *"the USA"* with a country name: *"Japan"*). More specifically, the set of tuples $\overline{PK}$ that represents the counter-PK is defined as follows:

$$\overline{PK} = \bigcup_{(s,r,o) \in PK} \texttt{Counter-PK}_k(s, r, o) \quad (2)$$

where:

$$\texttt{Counter-PK}_k(s, r, o) = \{(s, r, o, \bar{o}) \mid \bar{o} \in O_r \wedge \bar{o} \neq o \wedge \mathit{rank}_\theta(\bar{o} \mid s, r) \leq k\} \quad (3)$$

where $k$ is the number of counter-knowledge triplets per triplet $(s, r, o)$ in PK; $\mathit{rank}_\theta(o \mid s, r)$

[3] https://mistral.ai/news/mistral-large/

| subject | rel-lemma | object | query |
|---|---|---|---|
| Newport County A.F.C. | is-headquarter | Newport | Newport County A.F.C. is headquartered in |
| Norway | capital-city-of | Oslo | Norway's capital city, |
| WWE | is-headquarter | Stamford | WWE is headquartered in |
| Princeton University Press | is-headquarter | Princeton | Princeton University Press is headquartered in |
| Internet censorship | is-subclass | censorship | Internet censorship is a subclass of |
| McMurdo Station | part-of-continent | Antarctica | McMurdo Station is a part of the continent of |
| Windows Update | product-manufacture-by | Microsoft | Windows Update, a product manufactured by |
| Nintendo | located-in | Kyoto | The headquarter of Nintendo is located in |
| Microsoft Windows SDK | product-manufacture-by | Microsoft | Microsoft Windows SDK, a product manufactured by |
| Harare | capital-of | Zimbabwe | Harare, the capital of |

**Table 1:** A sample of the raw ParaRel dataset (Elazar et al., 2021)

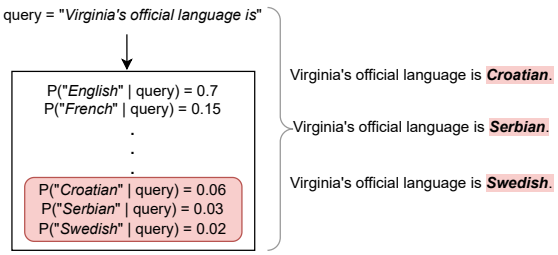


Figure 3: Example of 3 counter-knowledge objects that were associated to a parametric knowledge element. The probability distribution is ranked in an descendant order and we selected the objects with the lowerst probabilities.

is the rank of $\bar{o}$ among the $O_r$ ordered by the increasing probability $p(\hat{o} \mid q(s,r))$ of the LLM to generate an object $\hat{o} \in O_r$ given the prompt $q(s,r)$. We also make sure that the model has not learned the $(s,r,\bar{o})$ association by considering the objects $\hat{o}$ with the $k$ lowest ranks ($rank_\theta \leq k$) – indicating that the LLM is very unlikely to use its parametric knowledge to generate $\bar{o}$.

Figure 3 illustrates the counter-knowledge objects that were generated by Phi-1.5 for a parametric knowledge example.

### 3.3 Models

We consider decoder-only Transformer models. Between layer $l$ and $l-1$, the hidden state $X^{(l-1)}$ is updated by:

$$X^{(l)} = \gamma(X^{(l-1)} + A^{(l)})) + M^{(l)} \qquad (4)$$

where $A^{(l)}$ and $M^{(l)}$ are the outputs of the MHSA and MLP modules respectively, and $\gamma$ is a non-linearity.

The MLP module is a two-layer neural network parameterized by matrices $W^{(l)}_{mlp} \in \mathbb{R}^{d \times d_{mlp}}$ and $W^{(l)}_{proj} \in \mathbb{R}^{d_{mlp} \times d}$:

$$M^{(l)} = \sigma(X^{(l)}_{mlp} W^{(l)}_{mlp}) W^{(l)}_{proj} \in \mathbb{R}^{n \times d} \qquad (5)$$

where $\sigma$ is a non-linearity function (e.g. GeLU) and $X^{(l)}_{mlp}$ is the input of the MLP. We refer the reader to Vaswani et al. (2017) for more details on the architecture.

In our probing set-up (Section 4), we use the following activations: $\sigma(X^{(l)}_{mlp} W^{(l)}_{mlp})$ the first layer of the MLP (referred as MLP-L1 in this paper), $\sigma(X^{(l)}_{mlp} W^{(l)}_{mlp}) W^{(l)}_{proj}$ the output of the MLP (i.e. second layer, referred as MLP-L2 in this paper), and $A^{(l)}$ the output of the MHSA. We consider the first and second MLP layers activations, based on Geva et al. (2021) work, and also the MHSA activations as the attentions play a crucial role in information selection from the MLP memory (Geva et al., 2023).

We evaluate our method on several LLMs with different sizes: `Phi-1.5` with 1.3B parameters (Li et al., 2023), `Pythia-1.4B` with 1.4B parameters (Biderman et al., 2023), `Mistral-7B` with 7B parameters (Jiang et al., 2023), and `Llama3-8B` with 8B parameters (AI@Meta, 2024). Table 2 gives characteristics about the LLMs' modules dimensions.

| Model | MLP-L2 | MLP-L1 | MHSA |
|---|---|---|---|
| `Phi-1.5` | 2048 | 8192 | 2048 |
| `Pythia-1.4B` | 2048 | 8192 | 2048 |
| `Llama3-8B` | 4096 | 14336 | 4096 |
| `Mistral-7B` | 4096 | 14336 | 4096 |

**Table 2:** Activation dimensions for Phi-1.5, Pythia-1.4B, Llama3-8B and Mistral-7B for the different considered modules (MLP-L2, MLP-L1 and MHSA)

**Decoding strategy** As the generated sequences are short, we use a greedy decoding strategy and limit the number of generated tokens to 10.

| Relation Group ID | Relations | #Examples |
|---|---|---|
| geographic-geopolitic-language | *is-headquarter, located-in, headquarters-in, locate, share-border, is-twin-city-of, located, border-with, is-located, work-in-area, which-is-located, capital-city-of, part-of-continent, capital-of, headquarter, belong-to-continent, based-in, is-citizen-of, that-originate-in, originate-in, is-in, found-in, share-common-border, is-native-to, is-originally-from, pass-away-in, born-in, hold-citizenship-of, have-citizenship-of, citizen-of, start-in, formulate-in, legal-term, tie-diplomatic-relations, maintains-diplomatic-relations, have-diplomatic-relations, native, mother-tongue, original-language-is, the-official-language, communicate* | 2815 |
| corporate-products-employment | *product-manufacture-by, develop-by, owned-by, product-develope-by, product-release-by, create-by, product-of, produce-by, owner, is-product-of, is-part-of, who-works-for, employed-by, who-employed-by, works-for, work-in-field, profession-is, found-employment* | 1217 |
| media | *premiere-on, to-debut-on, air-on-originally, debut-on* | 128 |
| religion | *official-religion* | 249 |
| hierarchy | *is-subclass* | 183 |
| naming-reference | *is-call-after, is-name-after, is-name-for* | 6 |
| occupy-position | *play-in-position, who-holds* | 77 |
| play-instrument | *play-the* | 13 |

**Table 3:** All the relation groups with their corresponding relations and number of examples.

## 4 Probing Set-up

To build our probing dataset, we associate each tuple $(s, r, o, \bar{o}) \in \overline{PK}$ with a prompt $prompt(s, r, \bar{o})$ that corresponds to a natural language statement of $(s, r, \bar{o})$ followed by $q(s, r)$ (see Figure 1). Each prompt is associated with a label among CK, PK, and ND, where **CK** if $G_\theta(prompt(s, r, \bar{o})) = \bar{o}$, **PK** if $G_\theta(prompt(s, r, \bar{o})) = o$, and with ND (Not Defined) otherwise. In this work, we discard tuples associated with ND.

We specifically probe the activations $\bar{o}$ of the object, $s_q$ of the subject in the query, and $r_q$ the relation in the query. As each of these elements may have multiple tokens, we use their last tokens as their representative (e.g. for *"Washington"* $\rightarrow$ *["Wash", "inghton"]*, we consider the activations of the token *"inghton"*). The fact that this token representation summarizes the entity is intuitively true for decoder-only models and has been experimentally validated in (Meng et al., 2023; Geva et al., 2023).

Note that our first probe targets $\bar{o}$ as this is where the knowledge conflict starts (e.g. *Bill Gates is the founder of* ***Apple***$(\bar{o})$***. Bill Gates***$(s_q)$ ***is the founder of***$(r_q)$ _).

### 4.1 Control experiments

We also probe the activations of the first token to measure how much of the prediction can be attributed to the subject representation itself. Since the knowledge perturbation starts with the first object token, the first token activations should not indicate the knowledge source. For instance, in *Paris is located in Italy* the representation of the first token (*Paris*) should not contain information about the knowledge source as the perturbation starts at *Italy*.

### 4.2 Relation Groups

To avoid syntactic and semantic biases related to the type of relation when training a classifier, we grouped the relations that are similar into relation groups. The relation groups are illustrated in Table 3.

### 4.3 Evaluation

We use each relation group as a test set and train on the rest of the relation groups. We make sure that the train and test sets do not share similar subjects and objects to avoid biases related to the syntax or the nature of the relation and subject. We ensure the train set is balanced (equal number of CK and PK), as current LLMs are more likely to use context information (CK) than their parametric knowledge Xie et al. (2024). This is illustrated by Figure 4 (and Figure 7 in appendix for a breakdown by relation), where we can see that the considered LLMs mostly generate CK-based outputs.

We also ensure that the test set is balanced so

we can use the success rate (accuracy) as the main metric — with 50% being the performance of a random classifier. We compute the success rate $p_i$ for each group of relations. As $p_i$ follows a binomial distribution, we used a binomial proportion confidence interval to compute the weighted standard error (WSE – see formula 6) around the average success rate (see formula 9) with a 95% confidence interval to assess the significance of the resulting classification scores for each layer and token. We used the following formula in order to propagate the errors across the relation groups:

$$\text{WSE} = \sqrt{\sum_{i=1}^{G} \left(\frac{n_i}{N} \times \text{SE}_i\right)^2} \tag{6}$$

Where $\text{SE}_i$ is the standard error for the $i^{th}$ relation group, defined as:

$$\text{SE}_i = \sqrt{\frac{p_i \times (1 - p_i)}{n_i}} \tag{7}$$

$G = 8$ is the number of relation groups; $n_i$ is the number of test examples for the $i^{th}$ relation group; $N$ is the total number of test examples across all the relation groups.

The error bars are finally computed using a $z$-score of 1.96 for a confidence interval of 95%:

$$\text{Error Bars} = [P - 1.96 \times \text{WSE}, P + 1.96 \times \text{WSE}] \tag{8}$$

Where $P$ is the average success rate across all the relation groups:

$$P = \frac{\sum_{i=1}^{G} n_i \times p_i}{N} \tag{9}$$

Figure 5 presents the success rates for classifiers trained on activations from object, subject, and relation tokens, with the first token used as a control (see Section 4.1 for more details on the control experiment.) Results are reported for Mistral-7B, Phi-1.5, Llama3-8B, and Pythia-1.4B. Solid lines represent the average success rates across relation groups, while shaded areas denote the weighted standard error with a 95% confidence interval.

## 5 Results and Discussion

In Figure 5, we can first observe that the features linked to $\bar{o}$, the subject $s_q$ and the relation $r_q$ exhibit a correlation with the used knowledge source for MLP-L2, MLP-L1, and MHSA activations.

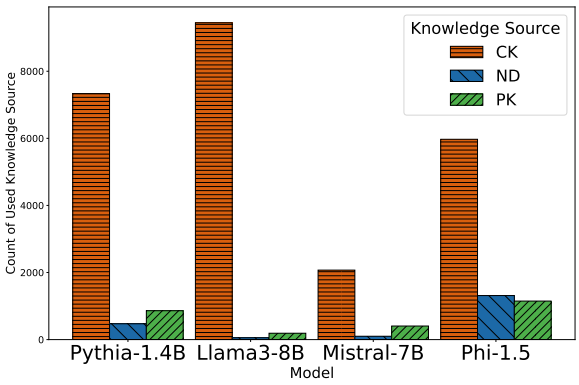


Figure 4: Count of used knowledge sources by each model (CK, PK, and ND). ND refers to outputs where the knowledge source is not defined.

Our framework successfully transfer the learned knowledge source patterns from one relation group to another, generalizing well from one group to another. The most predictive features are those of $r_q$, i.e. the relation token in the query. On certain layers, the success rate increases significantly, reaching 87% for Pythia-1.4b on the 15th layer at the relation token position.

This finding is consistent with prior research, which indicates that LLMs primarily store knowledge in the MLPs (Meng et al., 2023; Geva et al., 2021). Moreover, it supports Geva et al. (2023)'s insights on the information extraction process, where the relation token retrieves attributes from $s_q$'s MLPs using the MHSA (a process referred to as *Attribute Extraction*).

Additionally, it is noteworthy that the knowledge source can be detected directly starting from the perturbing object $\bar{o}$. This shows that detecting a potentially harmful conflict knowledge statement is possible early in the LLM inference process. MHSA activations are less connected to the used knowledge source than MLP-L2 and MLP-L1 activations.

The results of the control experiments conducted on the first token of the input indicate that the learned patterns in the object, subject, and relation are not arbitrary. The success rates of all LLMs for the first token appear to be random (about 0.5).

Finally, compared to (Yu et al., 2023), we show in this work that it is possible to predict the knowledge source based on the sole activations of an LLM, and, even more importantly, that we predict this for multiple relations rather than being limited to a single relation.

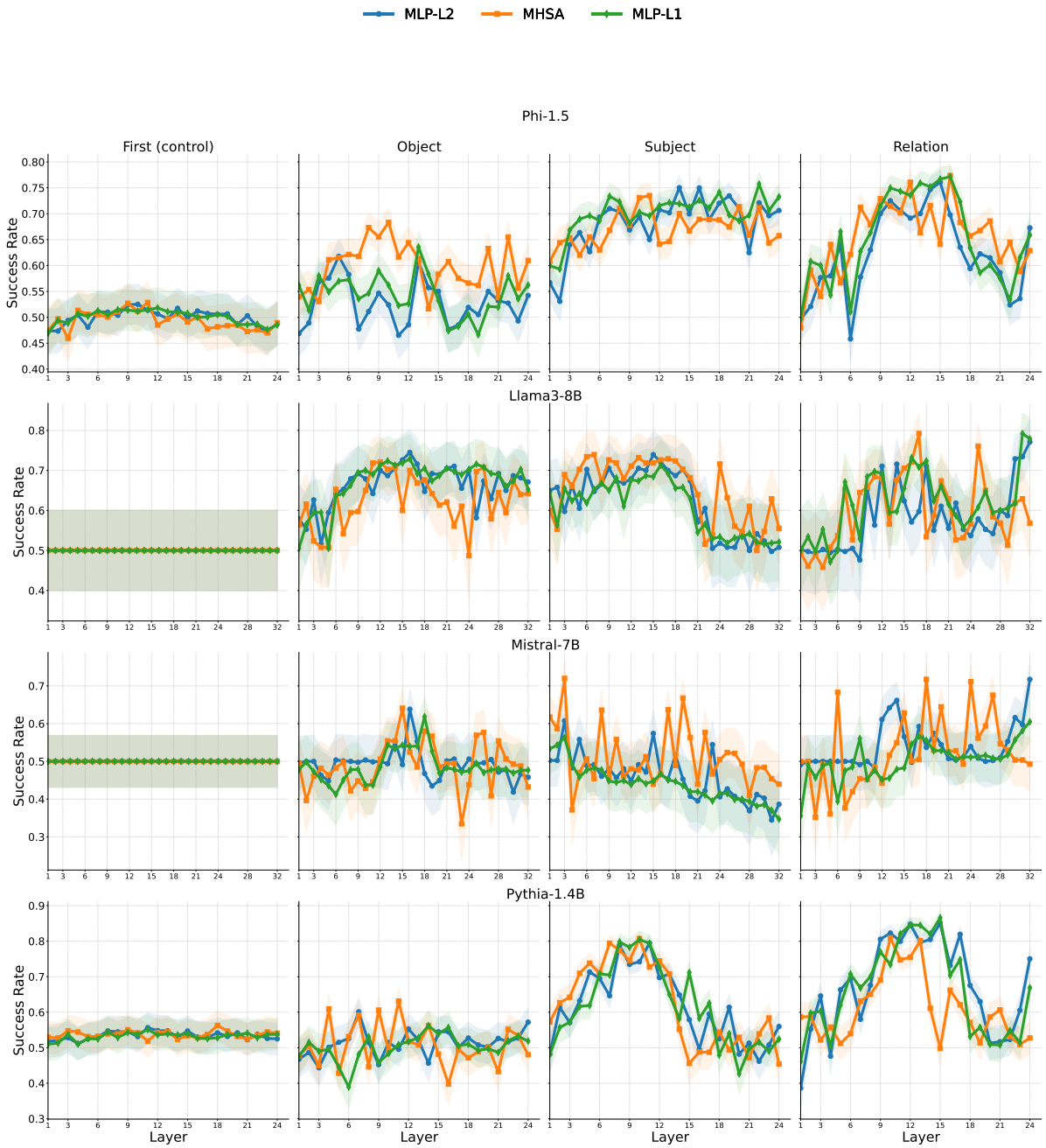


Figure 5: Performance of the linear classifier in identifying knowledge sources across different layers and modules (MLP-L2, MLP-L1, MHSA). The plots show success rates for classifiers trained on activations from object, subject, and relation tokens, with the first token used as a control (see Section 4.1 for more details on the control experiment.) Results are reported for the Mistral-7B, Phi-1.5, Llama3-8B, and Pythia-1.4B models. Solid lines represent the average success rates across relation groups, while shaded areas denote the weighted standard error with a 95% confidence interval. See Section 4.3 for further details on the evaluation methodology.

## 6 Subject frequency Vs. Knowledge Source

To understand what makes an LLM select the CK object over the PK object, we observed the subject frequency in The Pile corpus (Gao et al., 2020) for CK, PK, and ND outputs as illustrated in Figure 6 – We use The Pile as an approximation of what the LLMs might have learned except for Pythia-1.4B for which it is the training data. We used the infini-gram API made available by Liu et al. (2024) in order to get the frequencies. A Mann-Whitney U test reveals that the subject frequency distribution for PK outputs is significantly higher than for CK and ND outputs, except in the case of Pythia-1.4B, where PK is only higher than CK but not ND. This

| Input Prompt | Knowledge Source | PK Object | Model |
|---|---|---|---|
| *Harney County has its capital city in Taiwan. Harney County has its capital city in* ***Burns.*** | ND | Oregon | Llama3-8B |
| *Lisa Appignanesi has citizenship of Finland. Lisa Appignanesi has citizenship of* ***France.*** | ND | the UK | Llama3-8B |
| *Craiova is located in the continent of India. Craiova is located in the continent of* ***Romania.*** | ND | *Europe* | Pythia-1.4B |
| *The Kingdom of Hungary had its capital as Connecticut. The Kingdom of Hungary had its capital as* ***Connecticut.*** | CK | *Budapest* | Mistral-7B |
| *The Wii U system software is a product that was manufactured by Square. The Wii U system software is a product that was manufactured by* ***Square.*** | CK | *Nintendo* | Llama3-8B |
| *The Centers for Disease Control and Prevention is headquartered in Lyon. The Centers for Disease Control and Prevention is headquartered in* ***Lyon.*** | CK | *Atlanta* | Llama3-8B |
| *Harare is the capital city of Florida. Harare is the capital city of* ***Zimbabwe.*** | PK | *Zimbabwe* | Pythia-1.4B |
| *Goodreads is owned by Microsoft. Goodreads is owned by* ***Amazon.*** | PK | *Amazon* | Phi-1.5 |
| *OneDrive is owned by Toyota. OneDrive is owned by* ***Microsoft.*** | PK | *Microsoft* | Mistral-7B |

**Table 4:** Examples of final probing prompts, including their knowledge source, the LLM, and the corresponding parametric knowledge (PK) object. Bold text indicates the generated object, while underlined text represents the counter-knowledge object.

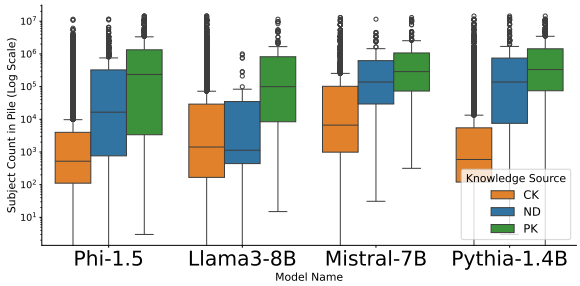


Figure 6: Subject frequency in the training dataset (The Pile) for CK, PK, and ND outputs. We use The Pile as an approximation of what the LLMs might have learned except for Pythia-1.4B for which it is the actual training data.

suggests that as a model gains more knowledge about a subject, it becomes more likely to select PK over CK objects.

## 7 Probing Dataset Examples

Table 4 illustrates some examples of the final probing prompts with their knowledge source, the LLM, and the corresponding PK object.

## 8 Conclusion

In this study, we introduced a novel probing framework to investigate if we can detect when LLMs switch from PK to CK. Our findings reveal that specific model activations are significantly correlated with the used knowledge source with a success rate reaching 87% for Pythia-1.4b. Additionally, our framework is able to transfer the learned knowledge source patterns from one relation group to another. This opens the door for future work investigating the mechanism at play when using CK or PK, and finally to building models that can better control this behavior.

## 9 Limitations

Our current framework is designed to probe LLMs by introducing contradictions to their learned knowledge, effectively identifying the source of knowledge. However, this controlled experimental setting does not account for many other situations, e.g. where the knowledge remains unperturbed. Future work should extend the framework to handle cases where both the parametric knowledge (PK) and the contextual knowledge (CK) are consistent or not related, providing a more comprehensive understanding of LLM behavior. Additionally, our study primarily measures the correlation between specific activations and the use of PK or CK, which, while providing valuable insights, does not establish an explanation of the underlying process. Further research is needed to uncover the underlying mechanisms that govern knowledge source selection in LLMs, possibly through experimental designs that manipulate specific model parameters or activations to observe resulting behavioral

changes.

It might also be interesting to employ a variety of prompt structures to mitigate biases associated with the conventional subject-relation-object format. Exploring alternative combinations, such as relation-subject-object (e.g., *The official language of Italy is Italian*), could yield valuable insights.

## 10 Ethical Considerations

Our probing framework of LLMs for their knowledge-sourcing behaviors only uses publicly available, non-personal datasets to ensure privacy and security. We recognize the potential for misuse of our findings. The insights derived from our research could be exploited to generate misleading information or make the models more susceptible to adversarial attacks. Therefore, we emphasize the importance of the ethical application of our work. Researchers and practitioners must implement robust safeguards to prevent the misuse of these technologies and ensure they are used to benefit society. Developing and deploying robust security measures is essential to protect against these vulnerabilities and maintain the integrity of information generated by LLMs. While we recognize inherent biases in LLMs, our commitment to transparency is demonstrated through the public release of our framework, facilitating reproducibility and further research.

## 11 Acknowledgements

We would like to thank BNP Paribas and the French National Association for Research and Technology (ANRT) who founded this project under the CIFRE program (2023/1673). We also thank Etienne Boisseau for his help on this paper.

## References

AI@Meta. 2024. Llama 3 model card.

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases. *Preprint*, arXiv:2204.06031.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *Preprint*, arXiv:2303.08112.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Preprint*, arXiv:2304.01373.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Preprint*, arXiv:2102.01017.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *Preprint*, arXiv:2304.14767.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. *Preprint*, arXiv:2012.14913.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

Jaro–Winkler. Jaro–winkler distance — Wikipedia, the free encyclopedia.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *Preprint*, arXiv:2309.05463.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. 2024. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *arXiv preprint arXiv:2401.17377*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kıcıman, Hamid Palangi, Barun Patra, and Robert West. 2024. A glitch in the matrix? locating and detecting language model grounding with fakepedia. *Preprint*, arXiv:2312.02073.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Preprint*, arXiv:1906.02530.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *Preprint*, arXiv:2211.00593.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. *Preprint*, arXiv:2305.13300.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. Characterizing mechanisms for factual recall in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9924–9959, Singapore. Association for Computational Linguistics.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

## A Data Characteristics

The `ParaRel` (Elazar et al., 2021) dataset includes 5313 unique subject-relation pairs, leading to the formation of the same number of PK triplets. After removing the examples where the subject is similar to the parametric object (see Section 3.1.3) we are left with approximately 3600 examples depending on the LLMs' parametric knowledge. We take $k = 3$ for `Counter-PK`$_k$ which gives approximately counter-PK 10k triplets. After undersampling, we are left with approximately 2000 balanced prompts depending on the LLM.

## B Data Undersampling Seed Impact

To examine the impact of random seed selection in undersampling for balanced CK and PK classes, we conducted an experiment with various seeds to determine if seed choice influenced model performance. Changing the seed for the undersampling of the majority class introduces significant variations in the success rate of our classifiers. This effect can be explained by the fact that the minority class (PK) has fewer samples than the majority class (CK), meaning there are very few CK examples in common between the datasets generated by two different seeds. In some cases, we observed a standard deviation of up to 11 points of accuracy in the success rate when varying the seed. However, the results stayed consistent with our conclusions across all choices of seeds.

## C Hardware and Software

Text generation tasks were performed using A100 GPUs, each equipped with 80 GB of memory. The process of generating the outputs spanned around 100 GPU hours. Our framework was constructed utilizing PyTorch (Paszke et al., 2019) and the HuggingFace Transformers library (Wolf et al., 2020).

## D License

**Model weights.** Llama3-8B weights are released under the license available at https://llama.meta.com/llama3/license/. Mistral-7B and Pythia-1.4B weights are released under an Apache 2.0 license. Mistral-Large weights are released under the licence available at https://mistral.ai/licenses/MRL-0.1.md. Phi-1.5 weights are released under a MIT license.

**Data.** The ParaRel dataset we used is released under a MIT License.

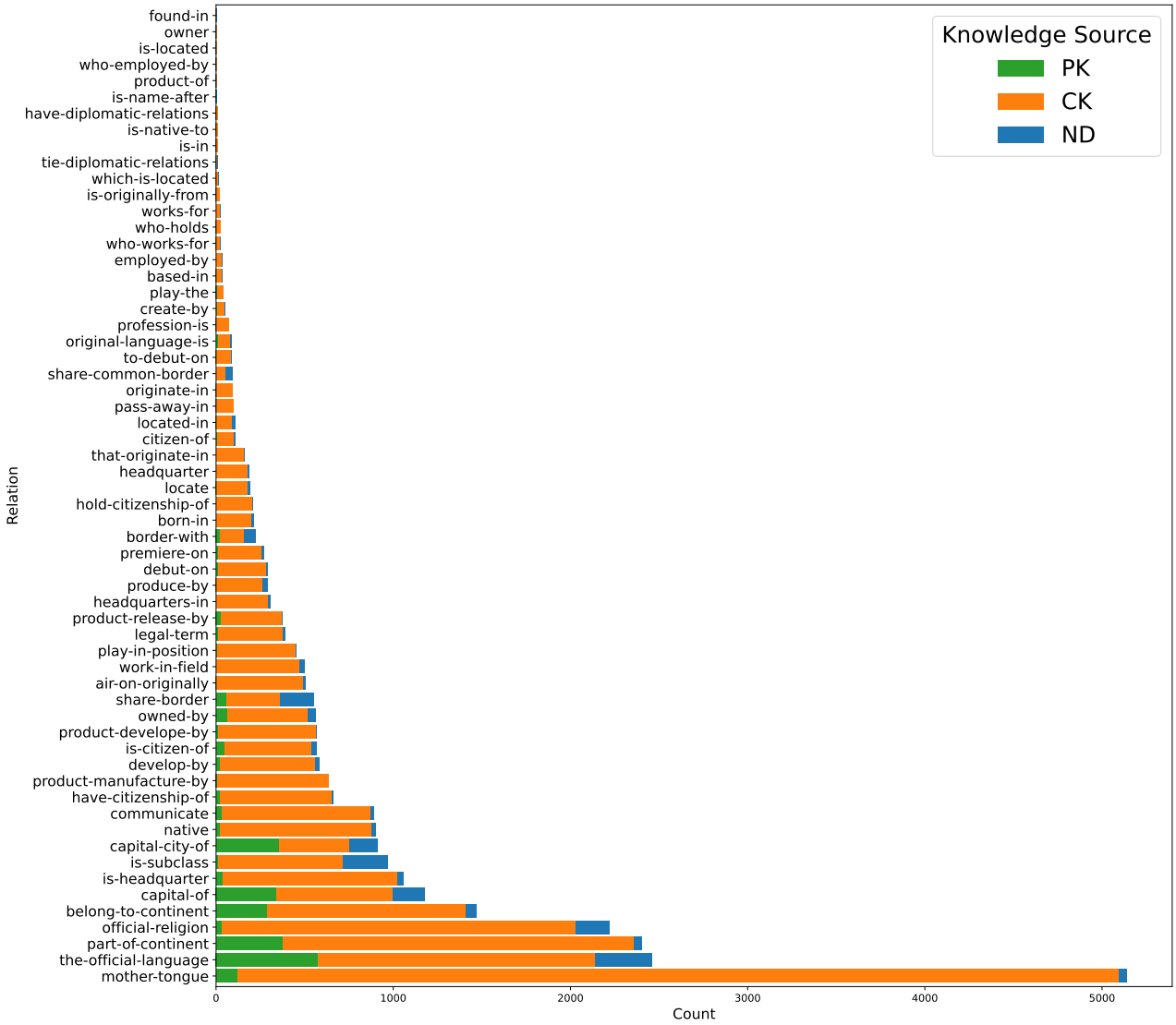


Figure 7: All the considered relations with the number of outputs that used CK (orange), PK (green), and ND (blue) sources (the counts include all the considered LLMs).