Supplemental Materials: A Property Induction Framework for Neural Language Models

Kanishka Misra
Purdue University
kmisra@purdue.edu

Julia Taylor Rayz
Purdue University
jtaylor1@purdue.edu

Allyson Ettinger
University of Chicago
aettinger@uchicago.edu

Code and analyses: https://github.com/kanishkamisra/lm-induction

1 Property Knowledge Re-annotation

Premise Datasets such as the CSLB (Devereux et al., 2014) naturally lend themselves to investigations that probe the conceptual knowledge of computational models and their representations. The CSLB dataset was collected by tasking 123 human participants to generate properties of a total of 638 concepts. For each property the authors then calculated its production frequency for all concepts for which it was generated, i.e., if the property *can fly* was generated for the concept ROBIN by 20 out of the 30 participants who were shown the concept, then its production frequency is 20. Note that the CSLB data set contains only positive property-concept associations. To construct negative samples, prior works that use CSLB as ground-truth to probe word representations typically use the set of concepts for which a given property was not generated, as negative (e.g. Lucy & Gauthier, 2017; Forbes et al., 2019; Da & Kasai, 2019; Bhatia & Richie, 2021). That is, negative samples are usually generated using concepts that have a production frequency of 0 for each property. Once a sufficient number of negative samples have been generated, the authors then train a probing classifier for every property, which predicts 1 if the production frequency of the property for that concept is nonzero, and 0 otherwise.

Limitation Since the task that was employed to construct the CSLB dataset was that of generation as opposed to validation, it is possible—and perhaps likely—that it resulted in inconsistent annotations, where some humans might have forgotten to generate *obvious* properties for certain concepts, or simply ignored them. For instance, the property *can breathe*, which is obviously applicable for all animals, was missing in 146 animal concepts within the dataset. This means that if one were to follow the standard negative-sampling method described earlier, they would consider all 146 of these animals as concepts for which the property *can breathe* does not hold true, which is incorrect. We conjecture that humans fail to generate features that are *obviously valid* for certain concepts (e.g., *can breathe, can grow, is a living thing* for animals) because they may be operating under Grice's maxim of quantity (Grice, 1989), by only eliciting non-trivial or *truly* informative properties for concepts in order to avoid redundancy. While we leave the testing of the hypotheses within this conjecture for future work, this limitation of incomplete data raises questions about the extent to which we should trust the results and conclusions of prior work which are crucially affected by this problem, which we summarize using the aphorism: *absence of evidence is not evidence of absence*.

Manual re-annotation of missing property-concept pairs To mitigate the limitation discussed above, we first selected the categories (hand-annotated by Devereux et al., 2014, e.g., BIRD, VEHICLE, TREE, etc.) that had at least 9 concepts in the dataset and were not labeled as "miscellaneous," resulting in 23 different categories with a total of 529 unique noun concepts, and 4,970 unique properties. Next, we manually removed concepts and properties that contained proper nouns (e.g., ROLLS-ROYCE, *is in Harry Potter*),

stereotypical or subjective data (e.g., *is meant for girls*, *is ugly*), and explicit mentions of similarity or relatedness (e.g., *is similar to horse*). We further normalized properties that were paraphrases of each other (e.g., *is used to flavor, does flavor* $\rightarrow$ *is used to flavor*). This resulted in 521 concepts and 3,735 properties. Again through manual search, we further identified a total of 365 properties that were incompletely annotated (i.e., those that were associated with certain concepts but were omitted for many relevant concepts during data collection—e.g., the property *can grow* was missing for all invertebrates, despite being associated with all of them). We manually extended the coverage for these properties by adding in entries for concepts for which they had not been elicited. For instance, for the property *can breathe*, which was generated for 6 out of 152 animals in the original dataset, we further add the remaining 146 concepts as additional positively associated concepts, increasing its coverage from 6 to 152. While the total number of incompletely annotated properties is small (10% of the valid properties), our re-annotation process greatly increases the total number of concept-property pairs (from 13,355 pairs in the original, unmodified dataset, to 23,107: an increase of 72%) since many of the incompletely labeled properties were applicable across several categories (e.g., *has a mouth, can grow,* etc). After applying this process to the CSLB dataset, we are left with 23,107 property-concept pairs, which we use in subsequent experiments. The re-annotated data can be found in the file `post_annotation_all.csv`[1] in the github repository.

Final thoughts The re-annotation process described above was performed manually due to resource, time, and financial constraints. However, we recommend running a large-scale empirical validation studies for datasets such as CSLB and McRae, before using them for probing experiments. While this is non-ideal in terms of resource use, it is necessary in order to draw faithful and appropriate conclusions about the correspondence between conceptual knowledge in humans and machines. Finally, a manuscript describing this process in greater detail, a small validation experiment ($\approx$2400 annotations) with humans, as well as empirical implications of the limitations described herein is in the works.

2 Negative Sample generation using Taxonomies

Here we describe our algorithm to generate negative samples for our first experiment in the paper—the property judgment task, where LMs are fine-tuned to classify as True or False sentences that attribute properties to concepts. For instance, the sentence *a cat can fly* is labeled as False as CAT is a negative sample for the property *can fly*, whereas, *a robin can fly* is labeled as True. Briefly, for the set of positive samples for a given property, we sample an equal-sized set of negative samples that are maximally similar to the positive samples. We use a taxonomic similarity (described below) as our similarity measure as it is model-free. Below we describe useful notation involved in the process, and then describe the full algorithm.

2.1 Notation and Preliminaries

Table 1 describes the notation we follow to construct our property judgment dataset. Our goal here is to generate 23,107 negative samples and then take the entire set of 46,214 concept-property pairs and their labels to carry out the property-judgment experiment.

In order to generate negative samples, we first tag the senses of all our 521 concepts using the WordNet (Miller, 1995) taxonomy, and also retrieve the sub-tree from WordNet that perfectly contains our concepts and use this as our ground-truth taxonomy on the basis of which we carry out subsequent experiments. We generate our negative samples by choosing a measure derived primarily from the Wu-Palmer similarity (Wu

[1] https://github.com/kanishkamisra/lm-induction/data/post_annotation_all.csv

Notation	Meaning	Remarks
$\mathcal{C}$	The set of all concepts in our experiments. These are also at the lowest level of the taxonomy—i.e., its leaf nodes.	$\lvert\mathcal{C}\rvert = 521$
$\mathcal{P}$	The set of all unique properties used in our experiments.	$\lvert\mathcal{P}\rvert = 3735$
$\mathcal{Q}_{P_i}$	The set of concepts that possess the property P_i.	$\mathcal{Q}_{P_i} \subset \mathcal{C}, \lvert\mathcal{Q}_{P_i}\rvert = k$
$\neg\mathcal{Q}_{P_i}$	The set of concepts that do not possess the property P_i, i.e., $\neg\mathcal{Q}_{P_i} = \mathcal{C} - \mathcal{Q}_{P_i}$	$\lvert\neg\mathcal{Q}_{P_i}\rvert = 521 - k$
$\delta(\neg\mathcal{Q}_{P_i}, k)$	A function that extracts k negative samples from $\neg\mathcal{Q}_{P_i}$ using the method described in Algorithm 1 (lines 6–9).	$\lvert\delta(\neg\mathcal{Q}_{P_i}, k)\rvert = k$,

Table 1: Notation for various artifacts involved in the paper.

& Palmer, 1994). This similarity can be computed over any taxonomy using the following operations:

$$sim_{\texttt{wup}}(c_i, c_j) = \frac{2 \times \texttt{depth}(\texttt{lcs}(c_i, c_j))}{\texttt{depth}(c_i) + \texttt{depth}(c_j)}, \tag{1}$$

where $\texttt{lcs}(x_1, x_2)$ is a function that computes the least-common subsumer[2] of the two[3] concepts, and $\texttt{depth}(x)$ computes the length of the path between the input concept and the root node of the hierarchy. We consider a generalized form of this measure (denoted as $sim_{\texttt{gwup}}$), to compute the similarity of a single concept to a set of concepts:

$$sim_{\texttt{gwup}}(c_1, \ldots, c_n) = \frac{n \times \texttt{depth}(\texttt{lcs}(c_1, \ldots, c_n))}{\texttt{depth}(c_1) + \cdots + \texttt{depth}(c_n)} \tag{2}$$

For every property P_i, we use this measure in algorithm 1 to sample k concepts from $\neg\mathcal{Q}_{P_i}$, based on their sim_{gwup} with $\mathcal{Q}_{P_i} = \{c_1, \ldots, c_k\}$. For example, consider the property *has striped patterns on its body*, the corresponding artifacts would be:

$$\begin{aligned} \mathcal{Q} &= \{\text{ZEBRA}, \text{TIGER}, \text{BEE}, \text{WASP}\} \\ \neg\mathcal{Q} &= \mathcal{C} - \mathcal{Q} \\ &= \{\text{ACCORDION}, \ldots, \text{YO-YO}\} \\ \text{NS} = \delta(\neg\mathcal{Q}, 4) &= \{\text{HORSE}, \text{LION}, \text{ANT}, \text{BEETLE}\} \end{aligned}$$

$$\begin{aligned} \mathcal{D} = \{&[\textit{a zebra has striped patterns on its body}, \mathsf{True}], \\ &\ldots, \\ &[\textit{a beetle has striped patterns on its body}, \mathsf{False}]\} \end{aligned}$$

Note that we follow the method outlined by Bhatia & Richie (2021) to convert concept-property pairs into sentences, which we denote as *sentencizer*() in Algorithm 1.

[2] a node in the hierarchy that is a hypernym/parent of the input concepts with minimum depth. For instance, $\texttt{lcs}(\text{ROBIN}, \text{BAT}) =$ VERTEBRATE.

[3] although in practice it can be applied for multiple concepts.

Algorithm 1 Algorithm to generate the dataset, $\mathcal{D}$, for the property judgment task

Input: $\mathcal{C} = \{c_1, \ldots, c_n\}$: Set of all concepts, $n = 521$.
$\mathcal{P} = \{P_1, \ldots, P_m\}$: Set of all properties, $m = 3735$.

1. $\mathcal{D} \leftarrow [\,]$ ▷ *the final set of stimuli for the property judgment task.*
2. **for** $i = 1, \ldots, m$:
3. $\quad \mathcal{Q}_{P_i} \leftarrow [c_1, \ldots, c_k]$ ▷ *set of k concepts that possess the property* P_i
4. $\quad \neg\mathcal{Q}_{P_i} \leftarrow \mathcal{C} - \mathcal{Q}_{P_i}$
5. $\quad$ ▷ *Lines 6–9 compute* $\delta(\neg\mathcal{Q}_{P_i}, k)$
6. $\quad \text{NS}_{P_i} \leftarrow [\,]$ ▷ *set of negative samples for the property* P_i
7. $\quad \neg\tilde{\mathcal{Q}}_{P_i} \leftarrow \text{argsort}(\neg\mathcal{Q}_{P_i}, sim_{\texttt{gwup}})$ ▷ *sort* $\neg\mathcal{Q}_{P_i}$ *based on* $sim_{gwup}(c_1, \ldots, c_k, x_j)\, \forall x_j \in \neg\mathcal{Q}_{P_i}$
8. $\quad$ **for** $j = 1, \ldots, k$:
9. $\quad\quad \text{NS}_{P_i}.\text{append}(\neg\tilde{\mathcal{Q}}_{P_i}[j])$ ▷ *take the top k concepts from* $\neg\mathcal{Q}_{P_i}$ *as negative samples*
10. $\quad$ ▷ *the following pairs the positive and negative samples with their labels, and appends them to* $\mathcal{D}$
11. $\quad$ **for** $j = 1, \ldots, k$:
12. $\quad\quad$ ▷ *sentencize() constructs a sentence using a concept and a property-phrase (see Bhatia & Richie, 2021).*
13. $\quad\quad \mathcal{D}.\text{append}([\textit{sentencize}(\mathcal{Q}_{P_i}[j], P_i), \text{True}])$
14. $\quad\quad \mathcal{D}.\text{append}([\textit{sentencize}(\text{NS}_{P_i}[j], P_i), \text{False}])$
15. **return** $\mathcal{D}$

3 Linear Mixed Effects Model Results

We use a linear-mixed effects models to test the connection between LMs' generalization behavior and the overlap in training data properties For each model, we use the following LMER specification (Bates et al., 2015):

$$\texttt{G} \sim \texttt{n + overlap * sim + (1|property) + (1|trial)},$$

where:

- `G` is the generalization score (see Eq. 1 in the paper).
- `n` is the number of adaptation concepts (i.e., the number of premise statements).
- `overlap` is the property overlap between the adaptation and the generalization set in each trial, calculated as the jaccard similarity between the binary property-vectors of each concept.
- `sim` is the cosine similarity between the embeddings (from the pre-contextualized layer in each model) of the concepts in the adaptation and generalization sets in each trial. Note that this is a model-dependent measure.
- `property` is the novel property (one out of 8) that is projected in the trial.
- `trial` is the individual trial.

In what follows, we report results from fitting this model to the results and statistics of our three property-induction models. We use Satterthwaite's method (Kuznetsova et al., 2017) to perform significance testing.

Fixed-effect	β	SE	t	df	p
`n`	0.0544	0.0035	15.41	304.59	$< 2e-16$
`overlap`	0.3951	0.0229	17.26	7123.98	$< 2e-16$
`sim`	0.1102	0.0180	6.11	6751.08	$1e-9$
`overlap * sim`	0.8583	0.2263	3.79	7170.65	0.0001

Table 2: Results for ALBERT-xxl

Fixed-effect	β	SE	t	df	p
`n`	0.0589	0.0046	12.76	396.84	$< 2e-16$
`overlap`	0.4731	0.0245	19.28	7088.53	$< 2e-16$
`sim`	0.1696	0.0487	3.48	6509.68	0.0005
`overlap * sim`	0.7429	0.3415	2.18	7180.52	0.03

Table 3: Results for BERT-large

Fixed-effect	β	SE	t	df	p
`n`	0.0555	0.0051	10.79	404.43	$< 2e-16$
`overlap`	0.3851	0.0269	14.32	7187.79	$< 2e-16$
`sim`	0.3631	0.0665	5.46	6180.44	$4.9e-8$
`overlap * sim`	-1.1735	0.4164	-2.82	7186.19	0.005

Table 4: Results for RoBERTa-large

References

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. doi: 10.18637/jss.v067.i01

Bhatia, S., & Richie, R. (2021). Transformer networks of human concept knowledge. *Psychological Review*.

Da, J., & Kasai, J. (2019). Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing* (pp. 1–12). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D19-6001 doi: 10.18653/v1/D19-6001

Devereux, B. J., Tyler, L. K., Geertzen, J., & Randall, B. (2014). The Centre for Speech, Language and the Brain (CSLB) concept property norms. *Behavior Research Methods*, *46*(4), 1119–1127.

Forbes, M., Holtzman, A., & Choi, Y. (2019). Do Neural Language Representations Learn Physical Commonsense? In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society.*

Grice, P. (1989). *Studies in the way of words*. Harvard University Press.

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, *82*(13), 1–26. doi: 10.18637/jss.v082.i13

Lucy, L., & Gauthier, J. (2017). Are distributional representations ready for the real world? evaluating word vectors for grounded perceptual meaning. In *Proceedings of the first workshop on language grounding for robotics* (pp. 76–85). Vancouver, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W17-2810 doi: 10.18653/v1/W17-2810

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Wu, Z., & Palmer, M. (1994). Verb Semantics and Lexical Selection. In *32nd Annual Meeting of the Association for Computational Linguistics* (pp. 133–138). Las Cruces, New Mexico, USA: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P94-1019 doi: 10.3115/981732.981751