

# COPEN: Probing Conceptual Knowledge in Pre-trained Language Models

Hao Peng<sup>1,2\*</sup>, Xiaozhi Wang<sup>1,2\*</sup>, Shengding Hu<sup>1,2</sup>, Hailong Jin<sup>1,2</sup>, Lei Hou<sup>1,2†</sup>, Juanzi Li<sup>1,2</sup>, Zhiyuan Liu<sup>1,2</sup>, Qun Liu<sup>3</sup>

<sup>1</sup>Department of Computer Science and Technology, BNRist;

<sup>2</sup>KIRC, Institute for Artificial Intelligence,

Tsinghua University, Beijing, 100084, China

<sup>3</sup>Huawei Noah’s Ark Lab

{peng-h21, wangxz20}@mails.tsinghua.edu.cn

## Abstract

Conceptual knowledge is fundamental to human cognition and knowledge bases. However, existing knowledge probing works only focus on evaluating factual knowledge of pre-trained language models (PLMs) and ignore conceptual knowledge. Since conceptual knowledge often appears as implicit commonsense behind texts, designing probes for conceptual knowledge is hard. Inspired by knowledge representation schemata, we comprehensively evaluate conceptual knowledge of PLMs by designing three tasks to probe whether PLMs organize entities by conceptual similarities, learn conceptual properties, and conceptualize entities in contexts, respectively. For the tasks, we collect and annotate 24k data instances covering 393 concepts, which is COPEN, a CONceptual knowledge Probing bENChmark. Extensive experiments on different sizes and types of PLMs show that existing PLMs systematically lack conceptual knowledge and suffer from various spurious correlations. We believe this is a critical bottleneck for realizing human-like cognition in PLMs. COPEN and our codes are publicly released at <https://github.com/THU-KEG/COPEN>.

## 1 Introduction

Pre-trained language models (PLMs) have achieved superior performance on most NLP tasks requiring substantial world knowledge (Qiu et al., 2020; Han et al., 2021). It is interesting and meaningful to *probe* the extent and scope of world knowledge within PLMs. Existing knowledge probing works have evaluated PLMs’ knowledge about entities (Broscheit, 2019; Tenney et al., 2019a) and their relations (Petroni et al., 2019; Jiang et al., 2020; Roberts et al., 2020), i.e., factual knowledge, but ignore conceptual knowledge.

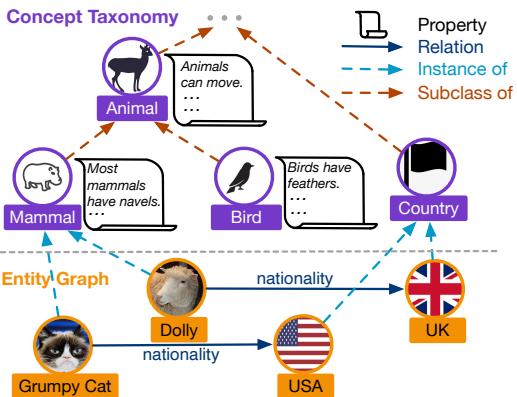


Figure 1: An example knowledge graph. Entities are organized by concepts through the *Instance of* relation and concepts are organized into a taxonomy through the *Subclass of* relation. Each concept has certain properties. Existing work only probes factual knowledge in entity graphs, ignoring conceptual knowledge in the concept taxonomy and *Instance of* relation.

Conceptual knowledge, especially the abstraction ability, is fundamental to all kinds of human cognition (Carey, 1991; Collins and Olson, 2014) including language processing (Waxman and Markow, 1995; Wellsby and Pexman, 2014). Just as the quote of psychologist Gregory Murphy, *concepts are the glue that holds our mental world together* (Murphy, 2004). Moreover, knowledge bases (Suchanek et al., 2007; Auer et al., 2007; Vrandečić, 2012) organize massive entities via concept taxonomies as illustrated in Figure 1, which enable broad applications (Lv et al., 2018; Zhou et al., 2021). Therefore, probing whether PLMs have human-like conceptual knowledge is necessary in knowledge probing.

Inspired by the conceptual schema in knowledge representations (Sowa, 1976; Decker et al., 2000; McGuinness et al., 2004; Antoniou and Van Harmelen, 2004), we comprehensively evaluate the conceptual knowledge of PLMs by asking three questions: Do PLMs organize entities by conceptual

\* Equal contribution

† Corresponding author: L.Hou

similarities? Do PLMs know the properties of concepts? Can PLMs correctly conceptualize entities in contexts? In this paper, we design three probing tasks for these questions: (1) **The conceptual similarity judgment (CSJ)** task studies whether PLMs organize entities by conceptual similarities, which is the basis of understanding concepts. Given a query entity, CSJ requires PLMs to choose the most conceptually similar entity among candidate entities. For example, in Figure 1, given Dolly as the query entity, although UK has a direct relation and more co-occurrences with it, PLMs should choose Grumpy Cat. (2) **The conceptual property judgment (CPJ)** task probes whether PLMs have the knowledge of conceptual properties, which are the generic abstractions of factual knowledge. Given a statement about a specific property, such as “*have feathers*”, CPJ requires PLMs to judge whether it is true for a specific concept and also a concept chain, which evaluates whether PLMs understand the property transitivity among a chain of hierarchical concepts. (3) **The conceptualization in contexts (CiC)** task evaluates the abilities of PLMs to correctly conceptualize entities within contexts. Given an entity mentioned in a specific context, PLMs are required to choose the most appropriate concept in a concept taxonomy according to its context. CiC requires not only disambiguating entity mentions, but also distinguishing superordinate and subordinate concepts. For instance, given the context “*Dolly is running on the grassland*”, PLMs should conceptualize Dolly as an Animal since there is no enough evidence for Mammal.

Based on the above considerations, we construct a conceptual knowledge probing benchmark, COPEN, which contains a concept taxonomy with 446 concepts and high-quality data of 24K instances for the three probing tasks. The concept taxonomy is curated by experts based on DBpedia (Auer et al., 2007) and Wikidata (Vrandečić and Krötzsch, 2014) to form a well-defined hierarchy and cover broad entities. The data instances for three tasks are collected by aligning entities in Wikidata and sentences in GenericsKB (Bhakthavatsalam et al., 2020), Wikipedia<sup>1</sup>, and Simple Wikipedia<sup>2</sup> into the concept taxonomy and then manually annotated by crowd-sourcing annotators.

We conduct extensive experiments on COPEN to evaluate various widely-used language mod-

els (LMs), which include three types: masked LMs (Devlin et al., 2019; Liu et al., 2019b), autoregressive LMs (Radford et al., 2019; Black et al., 2021), and sequence-to-sequence LMs (Lewis et al., 2020; Raffel et al., 2020). We conduct the experiments in three settings: (1) zero-shot probing, which reformulates the probing tasks into pre-training objectives and lets PLMs score answers without any training (Petroni et al., 2019); (2) linear probing, which only tunes additional linear classification heads and uses them to handle probing tasks with the frozen representations produced by PLMs; (3) fine-tuning, which tunes all the PLM parameters. Experiments show that existing PLMs achieve non-trivial performance but still significantly underperform ordinary persons on all three probing tasks. Further analyses show that PLMs suffer from spurious correlations like word co-occurrences and out-of-context predictions, and increasing model scale brings marginal improvements.

To summarize, our contributions are three-fold: (1) We propose to probe PLMs for conceptual knowledge, which has long been ignored, and design three probing tasks inspired by the knowledge representation works. (2) We construct COPEN, a probing benchmark containing high-quality concept taxonomy and probes. (3) We empirically show that existing PLMs systematically lack conceptual knowledge and analyze the reasons. We hope our benchmark and findings could facilitate further research on concept-aware PLMs and human-like language understandings.

## 2 COPEN Benchmark

In this session, we introduce our COPEN benchmark, including the construction of the concept taxonomy (§ 2.1) and the datasets for three probing tasks (§§ 2.2 to 2.4). More construction and annotation details are shown in appendix D.

### 2.1 COPEN Concept Taxonomy

Designing the three probing tasks takes inspiration from concept schemata in knowledge representations (Decker et al., 2000; McGuinness et al., 2004), which are widely used in knowledge graphs (Suchanek et al., 2007; Auer et al., 2007; Vrandečić, 2012). In general, it uses the instance of relation to link the entities (specific instances) into abstract concepts, and uses the subclass of relation to organize the concepts into a taxonomy. Each concept has certain properties describing it as

<sup>1</sup><https://en.wikipedia.org/>

<sup>2</sup><https://simple.wikipedia.org/>

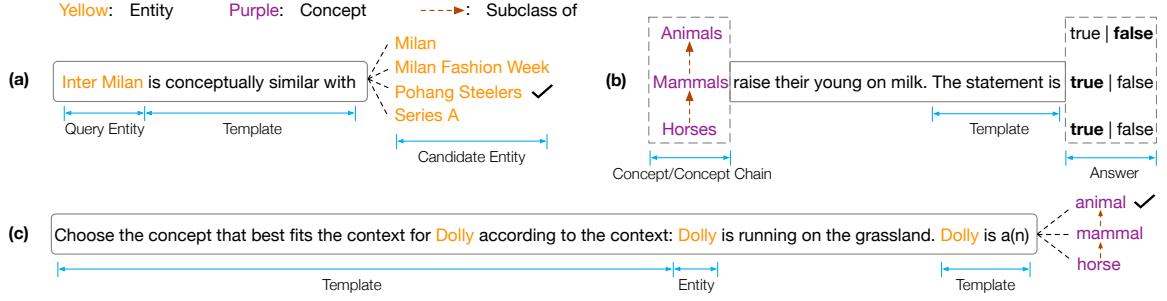


Figure 2: Examples for casting the data of three probing tasks into natural language prompts in zero-shot probing. The names of entities or concepts are the text looked up in Wikidata using their IDs. In Figure (b), **texts in bold** (true or false) denote answers. In Figure (b) and (c), the concept chain is Horse → Mammal → Animal. In Figure (c), for entities with multiple concept chains, each concept will be scored independently by PLMs, i.e., the PLMs make concept-level predictions only. There is no dedicated chain selection procedure.

the example shown in Figure 1.

To support probing dataset construction, we manually curate a concept taxonomy based on DBpedia (Auer et al., 2007) and Wikidata (Vrandečić and Krötzsch, 2014) in 3 steps: (1) Obtain a basic taxonomy from DBpedia. We extract the frequent concepts of DBpedia, which are the concepts with at least 5 instances, and keep the subclass of relations between them. (2) Align DBpedia and Wikidata. For each DBpedia concept, we manually find its equivalent Wikidata item and then use the subclass of (P279) relations in Wikidata to expand the concept taxonomy and use the instance of (P31) relations to link massive Wikidata entities into the concepts. (3) Simplify the taxonomy. We further remove some unusual concepts to simplify the taxonomy by the guidance from Schema.org (Guha et al., 2016). For example, Person is a sub-concept of Animal, Eukaryote, and Species in DBpedia, which is reasonable but inconvenient for real-world applications. Following Schema.org, we set Person as a top-level concept in the taxonomy. Finally, we achieve a tree-structure concise concept taxonomy, which contains 446 concepts covering 45 million Wikidata entities. There are 23 top-level concepts, and we use 11 of them and their sub-concepts for constructing training and development datasets as well as the other concepts for the testing datasets.

## 2.2 Conceptual Similarity Judgment

The conceptual similarity judgment (CSJ) task is a multiple-choice classification task, which probes whether PLMs organize entities by conceptual similarities, i.e., whether PLMs learn the instance of relation. Given a query entity, CSJ requires PLMs to choose the most conceptually similar en-

		Train	Dev	Test
CSJ	#Instance	4,462	1,116	3,909
	#Concept	84	84	90
CPJ	#Instance	3,274	823	4,758
	#Concept	215	195	178
CiC	#Instance	2,888	722	2,368
	#Concept	193	184	155

Table 1: COPEN data statistics for three probing tasks.

tity (instance of the same superordinate concept) among some candidates. As in Figure 2 (a), PLMs should choose Pohang Steelers for Inter Milan since they are both football clubs, although Milan and Inter Milan co-occur more frequently. The conceptual similarity here is similar to the *cohyponym* relation in lexical semantics (Cruse, 1986), which has been shown to be distinct from but easily influenced by spurious co-occurrence associations (Hill et al., 2015). Thus we need to control the influence of co-occurrences to get faithful results.

**Data Collection** The data for CSJ is collected in two steps: (1) Automatic collection. We first sample 174 concepts that are not subordinates to each other. Then we retrieve 50 Wikidata entities most frequently showing up in the Wikipedia corpus for each concept, and then build data instances by combining them. Each instance consists of a query entity, an answer entity of the same concept, and 20 distractor entities, among which 5 are hard distractors of concepts sharing superordinates with the concept of query entity. To check the data quality, we sample 200 instances and find little noise. (2) Co-occurrence-based filtering. To reduce the influence of co-occurrences, we need to filter out the instances that can be easily solved with co-

occurrences. Lastra-Díaz et al. (2019) show that Glove word embedding (Pennington et al., 2014) contains rich word co-occurrence information but limited cohyponym knowledge. Hence we use it to filter out instances with higher word similarity between the query and answer entity than distractor entities. We finally get 9,487 instances, each including a query entity and 21 candidate entities. The statistics of data subsets are shown in Table 1.

### 2.3 Conceptual Property Judgment

The conceptual property judgment (CPJ) task is a binary sentence classification task, which probes whether PLMs know the *properties* of concepts. Given a statement describing a certain conceptual property, PLMs are required to judge whether it is true. For example in Figure 2 (b), PLMs should predict “true” for the statement instance *Mammals raise their young on milk*.

Besides evaluating CPJ at instance level, which reflects the PLMs’ knowledge about properties for different individual concepts, we also set a **chain-level** evaluation, in which a PLM correctly judges a property if and only if it correctly judges the property for every concept in a *concept chain*. As the example in Figure 2 (b), a concept chain is a chain of concepts connected with the subclass of relation in order. The chain-level evaluation evaluates whether PLMs understand the transitivity of conceptual properties. It means that a property holds for a concept also holds for its subordinate concepts, but may not hold for its superordinate concepts like the case in Figure 2 (b).

**Data Collection** The data for CPJ is collected in two steps: (1) Automatic collection. For each concept in our taxonomy, we align it with the statements of GenericsKB (Bhakthavatsalam et al., 2020), a high-quality knowledge base for naturally occurring generic statements, by lexical matching so as to get positive instances. Then we replace the concept mention with other concept names to obtain negative instances. (2) Human annotation. To ensure data quality, we invite annotators to check whether the instances are correctly labeled, grammatically correct, and describing concept properties. All annotators are well-trained and pass a qualification before annotation. We finally get 8,855 instances for CPJ and the statistics of data subsets are shown in Table 1. Additionally, the final test data includes 102 concept chains and corresponding properties used for chain-level evaluation.

### 2.4 Conceptualization in Contexts

The conceptualization in contexts (CiC) task is a multiple-choice classification task, which probes whether PLMs can correctly conceptualize entities within contexts. Given an entity mentioned in a specific sentence, PLMs are required to choose the most appropriate concept among a concept chain, which is a chain of concepts connected with the subclass of relation in order. This requires PLMs to understand the subclass of relation and capture the subtle differences of different-level concepts in a hierarchy. For example in Figure 2 (c), given the sentence *Dolly is running on the grassland*, and a concept chain Horse → Mammal → Animal, PLMs shall choose Animal for Dolly since the context do not support more fine-grained concepts. Sometimes the entity is of multiple concept chains, for example, Jimmy Carter is both a Writer and a Politician, which additionally requires PLMs to disambiguate.

**Data Collection** The data for CiC is collected in two steps: (1) Sentence collection. For each concept, we first retrieve 10 Wikidata entities most frequently showing up in the Wikipedia corpus. Among the retrieved entities, we only keep the entities linked with the concept chains containing more than one concepts and collect 5 sentences for each of them from Wikipedia and SimpleWiki, which provides various contexts for conceptualization. A sentence, together with an entity mentioned in the sentence and concept chains of the entity, constitutes an instance. (2) Human annotation. We then organize crowd-sourcing annotation to obtain the labels. All annotators are well-trained and qualified. We finally get 5,978 instances for CiC and the statistics of data subsets are shown in Table 1.

## 3 Evaluation Setup

We introduce the various widely-used PLMs investigated in our experiments (§ 3.1) and the three adopted probing methods (§ 3.2).

### 3.1 Investigated PLMs

We investigate three mainstream types of PLMs: (1) **Masked LM**, including BERT (Devlin et al., 2019), which is pre-trained with the bidirectional masked language modeling and next sentence prediction objectives, and RoBERTa (Liu et al., 2019b), which is a robustly optimized version of BERT. (2) **Autoregressive LM**, including GPT-2 (Radford et al.,

Model	CSJ			CPJ						CiC		
				Instance-Level			Chain-Level					
	ZP	LP	FT	ZP	LP	FT	ZP	LP	FT	ZP	LP	FT
Random	4.8	4.8	4.8	50.0	50.0	50.0	7.2	7.2	7.2	27.7	27.7	27.7
BERT <sub>BASE</sub>	<b>20.3</b>	<b>16.1</b> <sub>0.21</sub>	27.3 <sub>0.86</sub>	49.4	61.6 <sub>0.28</sub>	68.1 <sub>0.98</sub>	<b>22.5</b>	<b>24.2</b> <sub>1.22</sub>	<b>23.2</b> <sub>1.22</sub>	37.6	34.3 <sub>0.59</sub>	49.5 <sub>0.60</sub>
RoBERTa <sub>BASE</sub>	15.5	12.0 <sub>0.21</sub>	22.3 <sub>0.51</sub>	49.2	61.9 <sub>0.13</sub>	72.0 <sub>0.54</sub>	21.6	13.1 <sub>1.67</sub>	18.3 <sub>1.22</sub>	31.4	30.0 <sub>1.98</sub>	52.6 <sub>1.02</sub>
GPT-2 <sub>BASE</sub>	7.9	4.3 <sub>0.24</sub>	20.1 <sub>0.23</sub>	51.5	64.8 <sub>1.14</sub>	70.4 <sub>0.72</sub>	14.7	14.4 <sub>0.92</sub>	20.3 <sub>2.01</sub>	32.3	34.5 <sub>2.08</sub>	54.2 <sub>0.12</sub>
GPT-Neo125M	7.9	11.0 <sub>0.20</sub>	18.3 <sub>0.42</sub>	52.2	62.2 <sub>0.21</sub>	68.2 <sub>0.62</sub>	22.5	15.0 <sub>2.01</sub>	19.0 <sub>2.81</sub>	32.6	39.6 <sub>0.93</sub>	47.4 <sub>0.25</sub>
BART <sub>BASE</sub>	14.4	8.4 <sub>0.10</sub>	21.0 <sub>0.50</sub>	48.7	58.5 <sub>0.27</sub>	68.2 <sub>0.86</sub>	20.6	10.5 <sub>1.22</sub>	16.7 <sub>0.80</sub>	33.6	<b>43.7</b> <sub>1.19</sub>	51.3 <sub>1.56</sub>
T5 <sub>BASE</sub>	15.2	4.9 <sub>0.21</sub>	<b>27.9</b> <sub>0.60</sub>	<b>55.9</b>	<b>66.9</b> <sub>0.25</sub>	<b>72.5</b> <sub>0.28</sub>	22.5	18.0 <sub>0.46</sub>	18.0 <sub>3.95</sub>	<b>42.3</b>	24.7 <sub>0.66</sub>	53.2 <sub>0.18</sub>
Human	79.5	79.5	79.5	91.4	91.4	91.4	91.2	91.2	91.2	85.6	85.6	85.6

Table 2: Accuracies (%) of various PLMs on the three tasks using different probing methods. ZP: Zero-shot probing. LP: Linear probing. FT: Fine-tuning. LP and FT results are Mean<sub>standard deviation</sub> over three random trials. Human performance is obtained by ordinary people trained with a few instances.

2019), which is pre-trained with the unidirectional left-to-right language modeling objective, and GPT-Neo (Black et al., 2021), which adopts the same objective but improves some implementation details. (3) **Sequence-to-sequence LM**, which adopts the encoder-decoder architecture. This type includes BART (Lewis et al., 2020), which is pre-trained with the text infilling and sentence permutation objectives, and T5 (Raffel et al., 2020), which is pre-trained with the span-corruption objective and multiple downstream tasks.

In § 4, we report the results of the frequently-used BASE versions of these PLMs, and results for the other versions are shown in appendix C.

### 3.2 Probing Method

**Zero-Shot Probing** reformulates probing tasks to the format of pre-training language modeling objectives (Liu et al., 2021a) so that PLMs can do these tasks without any training. It is widely adopted by knowledge probing work (Petroni et al., 2019; Tenney et al., 2019a) since it prevents PLMs from learning new knowledge from training data so that the achieved performance reflects PLMs’ intrinsic knowledge. Hence the performance of zero-shot probing is commonly interpreted as the *lower bound* of PLMs’ knowledge (Jiang et al., 2020).

As illustrated in Figure 2, for each data instance of the three probing tasks, we cast its choices into natural language prompts by filling them into manually designed templates, and then let PLMs score the prompts by the likelihood of language modeling. The choice with the highest score is regarded as the predicted answer of PLMs. Some implementation details like taking which parts of the prompts into scoring calculation may influence the PLMs’ performance. We search these details with preliminary trials and only report the performance of the

best configuration in experiments.

**Linear Probing** adds an additional shallow linear classifier on top of the output contextualized representations of PLMs, and only trains the additional classifier while keeping the PLMs’ parameters fixed. Since the model capacity of the shallow linear classifier is too limited to fit the tasks, the achieved performance shall mainly come from the knowledge in the PLMs’ representations (Alain and Bengio, 2017). Hence linear probing is widely used in knowledge probing (Tenney et al., 2019b; Hewitt and Manning, 2019).

**Fine-Tuning** is the standard method to adapt PLMs to downstream tasks, which trains all the PLMs’ parameters on the training data with task-specific objectives. Considering the strong model capacity of the PLMs, PLMs will inevitably fit the probing tasks through the information in training data rather than only resort to their intrinsic knowledge. Hence the fine-tuning performance shall serve as an *upper bound* of the PLMs’ conceptual knowledge in our experiments.

For CSJ and CiC, we take the filled prompts of identical templates in zero-shot probing as inputs and train PLMs with the cross-entropy loss. For CPJ, we take the property statements as inputs and use the binary cross entropy loss.

More detailed implementations about three probing methods are shown in appendix A.

## 4 Experiment and Analysis

We first introduce the overall results in § 4.1 and conduct detailed analyses on the three probing tasks (§§ 4.2 to 4.4), respectively. We then analyze the performance at different model scales (§ 4.5). More observations and discussions on experimental results are placed in appendix B.

Model	Hard Distractor	Easy Distractor
BERT <sub>BASE</sub>	25.1	15.7
RoBERTa <sub>BASE</sub>	25.3	15.7
GPT-2 <sub>BASE</sub>	21.1	17.0
GPT-Neo <sub>125M</sub>	20.7	17.1
BART <sub>BASE</sub>	24.2	16.0
T5 <sub>BASE</sub>	24.6	15.9

Table 3: Mean reciprocal ranks (%) for hard distractors and easy distractors on CSJ in zero-shot probing results of various PLMs. Larger values for higher ranks.

## 4.1 Overall Results

The overall experimental results are shown in Table 2, from which we can observe that: (1) All the PLMs can achieve non-trivial (better than random guess) performance on all the probing tasks with zero-shot probing or linear probing, which indicates that existing PLMs capture a certain conceptual knowledge with pre-training on massive texts. (2) However, even with fine-tuning, all PLMs’ accuracies are still well below human performance, which urges further efforts on concept-aware pre-training. (3) The accuracies of PLMs using different types of pre-training objectives are generally on the same level. It suggests that any existing pre-training objective has no special advantages in understanding concepts and further improvements may come from targeted pre-training design. We provide some analyses in the following sections to help targeted concept-aware PLMs development.

## 4.2 Conceptual Similarity Judgment

We analyze the predictions and performance of various PLMs on CSJ, and find that:

**PLMs better distinguish coarse-grained concepts.** As mentioned in § 2.2, among 20 distractor entities, 5 of them are hard distractors of concepts sharing superordinates with the concept of the query entity, and the others are easy distractors. For example, if the query entity is of Mammal concept, the entities of Bird concept are hard distractors and the entities of Country concept are easy distractors. Table 3 shows the mean reciprocal ranks of these two kinds of distractors. We can see that the hard distractors are significantly ranked higher than easy distractors, which indicates that PLMs generally better distinguish coarse-grained concepts, such as telling the differences between Animal and Country, but fail in distinguishing fine-grained concepts. It suggests that future methods should focus more on how to capture the subtle

BERT	RoBERTa	GPT-2	GPT-Neo	BART	T5
78.0	72.5	64.6	52.5	65.9	58.3

Table 4: Percentage (%) of false positive predictions among all incorrect predictions in fine-tuning results of various PLMs on the CPJ dataset.

differences between fine-grained concepts.

## 4.3 Conceptual Property Judgment

We analyze the error cases on CPJ and find that:

**Conceptual transitivity challenges PLMs.** Table 2 shows that PLMs can achieve high instance-level accuracies, but all perform poorly in the chain-level evaluation. It suggests that PLMs can relatively well recall the properties for individual concepts like recalling the facts about entities in factual knowledge probing, but hardly understand the hierarchical relations of concepts and the property transitivity. It suggests that further PLM works should not only focus on better memorizing knowledge but also consider how to better organize knowledge.

**PLMs have conceptual hallucination.** It has been observed that PLMs frequently generate non-sensical and unfaithful outputs, which are factually incorrect, and previous work (Rohrbach et al., 2018; Reiter, 2018; Ji et al., 2022) dubs this phenomenon as *hallucination*. In our experiments, we observe that many PLMs’ failure cases on CPJ task can be described as *conceptual hallucination*, i.e., PLMs hallucinate that concepts have certain properties while they actually do not. As shown in Table 4, the errors of most PLMs are generally mainly from making false positive predictions, i.e., taking false conceptual property statements as true. It suggests that PLMs tend to hallucinate the false conceptual properties as true rather than cannot recall the true conceptual properties, which is interesting and we further explore whether there are certain spurious correlations causing this.

**Word co-occurrence causes conceptual hallucination.** We hypothesize that the word co-occurrence in the pre-training corpora causes PLMs’ conceptual hallucination. For example, if a PLM has seen the text “*The temple’s Jufu Hall was included in the 1998 World Monuments Watch by the World Monuments Fund (WMF) ...preservation of the painted decoration*”<sup>3</sup>, it may be more

<sup>3</sup>[https://en.wikipedia.org/wiki/Temple\\_of\\_Agriculture](https://en.wikipedia.org/wiki/Temple_of_Agriculture)

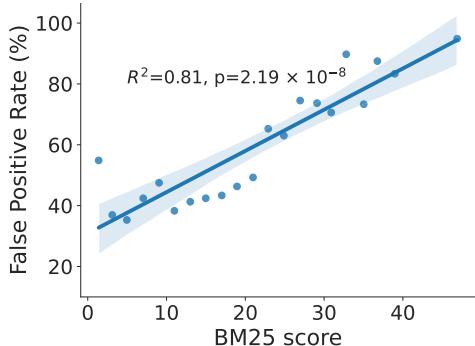


Figure 3: The false positive rate of BERT’s fine-tuning results on CPJ negative instances with different BM25 scores. Results of other PLMs are left in appendix C.1.

likely to predict the statement “*Monuments are used for decoration*” as true. We empirically find pieces of evidence supporting this hypothesis. For each CPJ instance, to assess the word co-occurrence in pre-training corpora, we retrieve the most similar document of it from Wikipedia, which is a widely-used corpus in pre-training, with the BM25 (Robertson et al., 1995) algorithm implemented in Whoosh (Mchaput, 2016), and use the BM25 score of the top one of retrieved documents as the indicator of this CPJ instance’s word co-occurrence rate in pre-training corpus. We divide the negative instances of CPJ dataset into different subsets by their BM25 scores and observe the false positive rate of BERT’s fine-tuning predictions on them. The results are plotted in Figure 3, from which we can see that the false positive prediction rates, indicating conceptual hallucination, have strong positive correlations to the BM25 scores, indicating word co-occurrence. This suggests that the conceptual hallucination of PLMs comes from capturing the spurious correlations of word co-occurrence in pre-training, and further pre-training work shall explore to fix it.

#### 4.4 Conceptualization in Contexts

We analyze the error cases on CiC and find that:

**PLMs conceptualize entities over-relying on memories.** In CiC, we find that if we remove the contexts, PLMs can still predict a possibly correct concept, which is similar to previous works (Petroni et al., 2019; Roberts et al., 2020; Cao et al., 2021) showing that PLMs memorize a certain knowledge about entities’ types. We dub these predictions *out-of-context predictions*, which can be regarded as the PLMs’ memories obtained in pre-training. What we evaluate in CiC is the

BERT	RoBERTa	GPT-2	GPT-Neo	BART	T5
72.9	75.9	76.7	60.4	71.8	59.2

Table 5: Percentage (%) of out-of-context predictions among all incorrect predictions in zero-shot probing results of various PLMs on the CiC dataset.

in-context conceptualization abilities rather than the memorized knowledge about the concepts of entities, which is evaluated by CSJ. Hence relying on the memories and making out-of-context predictions are wrong for handling CiC. However, as shown in Table 5, in most of the error cases, PLMs wrongly conceptualize the entities within contexts as the default out-of-context predictions. It demonstrates that PLMs conceptualize entities by over-relying on memories rather than understanding the contexts, which reflects the lack of genuine conceptualization abilities. We encourage future works to study whether the memories inhibit learning to conceptualize during pre-training.

**Understanding hierarchy is more difficult than disambiguation.** In Table 6, we analyze the two error types on CiC task. *Disambiguation* indicates the PLM selects a wrong concept chain for the given entity and *Wrong Level* indicates the PLM selects a wrong-level concept in the correct chain. In the analysis, we only consider entities with more than one concept chain. The *Wrong Level* errors take up the majority, which shows that understanding concept hierarchy is more difficult than disambiguation for PLMs and how to teach the PLMs to understand it is essential.

#### 4.5 Analysis on Model Scale

Inspired by recent advances showing the superior advantages of large-scale models (Kaplan et al., 2020; Lester et al., 2021), we explore how the model scale influences PLMs’ conceptual knowledge. We investigate the family of three representative PLMs: BERT, GPT-2 and T5. Since fine-tuning extremely-large PLMs is too computationally expensive, for models with more than 2.5 billion parameters, we instead adopt BitFit (Zaken et al., 2022), which can achieve similar performance to fine-tuning (He et al., 2021) but requires much less computation. The results are shown in Figure 4, and we have following observations: (1) Larger-scale PLMs generally achieve better performance on all the probing tasks, which suggests that increasing model scale can store more conceptual

Error Type	Context	Concept Chains
Disambiguation 29.0%	He was nominated by President <i>Jimmy Carter</i> to the court.	Person → <u>BusinessPerson</u> Person → Writer Person → <b>Politician</b>
Wrong Level 71.0%	Dolly is running on the grassland.	Horse → <u>Mammal</u> → <b>Animal</b>

Table 6: Error examples sampled from zero-shot probing results of BERT<sub>BASE</sub> on the CiC dataset. *Italics* denote entities. Underlines denote model predictions. **Texts in bold** denote answers.

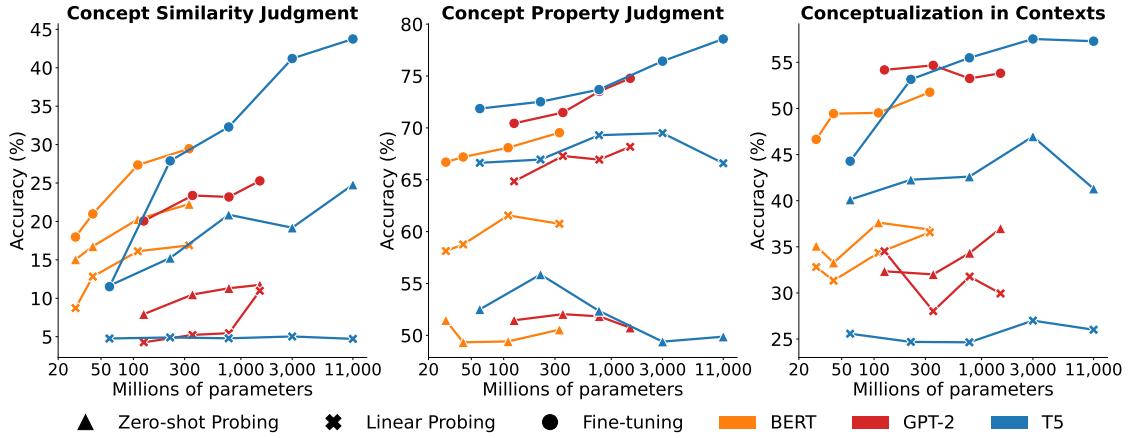


Figure 4: Accuracies (%) of various PLMs at different scales. The accuracies on CPJ are instance-level.

knowledge. However, the improvements brought by increasing model scale are generally marginal, especially on CiC task, and the improvements in zero-shot probing and linear probing results are not so obvious like in fine-tuning, which poses a question that whether the fine-tuning improvements come from the intrinsic knowledge of PLMs. (2) The fine-tuning accuracies of T5<sub>11B</sub> with 11 billion parameters, are still well below ordinary people, which demonstrates that acquiring conceptual knowledge is quite challenging for existing pre-training methods, which encourages further efforts on building concept-aware PLMs.

## 5 Related Work

**Knowledge Probing** To understand the success of PLMs, extensive works explore to know what PLMs know, and find PLMs have strong linguistic knowledge (Liu et al., 2019a; Hewitt and Manning, 2019; Tenney et al., 2019b; Vulić et al., 2020). Moreover, it has been shown that PLMs have a certain world knowledge, which is typically stored in world knowledge bases, such as the knowledge about entities (Broscheit, 2019; Tenney et al., 2019a) and their relationships (Petroni et al., 2019; Roberts et al., 2020; Jiang et al., 2020; Bouraoui et al., 2020; Zhong et al., 2021). However, these ex-

plorations are limited in the scope of factual knowledge, ignoring the conceptual knowledge, which is essential for both knowledge bases (Wu et al., 2012; Ji et al., 2019) and intelligence (Carey, 1991; Collins and Olson, 2014). Hence we explore the conceptual knowledge probing in this paper.

**Conceptual Knowledge in PLMs** Previous works also explore the *concept* in PLMs (Michael et al., 2020; Talmor et al., 2020; Aspíllaga et al., 2021; Dalvi et al., 2021), which study principally similar topics with us. However, the *concept* they refer to is essentially *word sense*. They focus on whether PLMs discover the word senses and recognize their hierarchical relations. While in this work, we study the concepts defined in knowledge bases to abstract real-world entities, which support broader applications (Lv et al., 2018; Zhou et al., 2021; Zeng et al., 2021), and probe knowledge about conceptual similarity and properties of concepts as well as PLMs’ conceptualization ability.

## 6 Conclusion and Future Work

In this paper, we systematically analyze the conceptual knowledge in existing PLMs by constructing a high-quality conceptual knowledge probing benchmark (COPEN). Extensive experiments show that

existing PLMs have a certain conceptual knowledge, but are significantly worse than humans, even with billions of parameters. We further find that PLMs fail in distinguishing fine-grained concepts and understanding concept hierarchy, and suffer from conceptual hallucination caused by word occurrence and out-of-context bias. In the future, inspired by works infusing factual knowledge, we will try to develop conceptual knowledgeable PLMs by exploring concept-aware pre-training objectives and knowledge-enhanced architectures.

## Limitations

In the section, we discuss the limitations of this work: (1) **COPEN benchmark**. COPEN only involves English corpora, which limits the use of the benchmark to PLMs pre-trained on other languages. In the future, we will consider more languages and construct multilingual COPEN. (2) **Large PLMs**. We do not experiment on very large PLMs, such as GPT-3 (Brown et al., 2020) and PaLM (Chowdhery et al., 2022), due to our limited access to them. We conduct experiments on T5<sub>11B</sub> with 11 billion parameters instead. Experimental results demonstrate that acquiring conceptual knowledge is quite challenging for existing pre-training methods, which urges concept-aware pre-training objectives and model architectures. (3) **Environmental impact**. In this paper, we conduct a lot of experiments with various PLMs, some of which even contain several billions of parameters. It consumes large amounts of energy and causes large amounts of carbon dioxide emissions, which incurs negative influence to our environment (Strubell et al., 2019). But the experiments are necessary for drawing faithful and comprehensive conclusions. We hope our findings could facilitate further research on more powerful PLMs with fewer parameters.

## Ethical Considerations

We discuss the ethical considerations and broader impact of this work in this section: (1) **Intellectual property**. The Wikipedia, Simple Wikipedia corpora, and Wikidata are obtained from the Wikimedia dump<sup>4</sup>, which is shared under the CC BY-SA 3.0 license<sup>5</sup>. The DBpedia<sup>6</sup> is shared under the CC BY-SA 3.0 license and GNU Free Docu-

mentation License<sup>7</sup>. The GenericsKB corpus<sup>8</sup> is shared under the CC BY 4.0 license<sup>9</sup>. These are all public and established resources, which are intended to support broad artificial intelligence and NLP research. We believe these resources are well desensitized and anonymized. (2) **Data annotation**. We invite 19 annotators without background of expertise to annotate our datasets and produce human performance. They are all employed by commercial data production companies. The invited annotators are fairly paid according to agreed working hours and prices. The annotators are all informed about how the data will be processed, used, and released, and this is confirmed in the data production contract. (3) **Intended use**. COPEN is a high-quality benchmark used for evaluating conceptual knowledge in PLMs and developing concept-knowledgeable PLMs. Researchers can use COPEN to assess new concept-aware objectives and conceptual-knowledge-enhanced architectures. (4) **Misuse risks**. Considering COPEN is built on top of a limited scope of natural texts and the probing methods are inevitably influenced by some spurious correlations, a good enough performance on COPEN cannot fully guarantee that the developed methods really understand concepts and shall not be used to support relevant commercial and political claims. (5) **Potential risks control**. The texts in COPEN are from public data and do not involve private information, sensitive topics and social issues. The three tasks in COPEN also do not involve sensitive topics or social issues. We manually check some randomly sampled instances in COPEN and find no sensitive information or other risky issues. Hence we believe that COPEN does not create additional risks.

## Acknowledgements

This work is supported by the Key-Area Research and Development Program of Guangdong Province (2019B010153002), the Institute for Guo Qiang, Tsinghua University (2019GQB0003), and Huawei Noah’s Ark Lab. The authors thank all the anonymous reviewers for their detailed and valuable comments and suggestions. The authors also thank all the annotators for their substantial efforts in the annotation process.

<sup>4</sup><https://dumps.wikimedia.org/>

<sup>5</sup><https://creativecommons.org/licenses/by-sa/3.0/>

<sup>6</sup>[www.dbpedia.org](http://www.dbpedia.org)

<sup>7</sup><https://www.gnu.org/licenses/fdl-1.3.html>

<sup>8</sup><https://allenai.org/data/genericskb>

<sup>9</sup><https://creativecommons.org/licenses/by/4.0/>

## References

- Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *Proceedings of ICLR*.
- Grigoris Antoniou and Frank Van Harmelen. 2004. *A semantic web primer*. MIT press.
- Carlos Aspillaga, Marcelo Mendoza, and Alvaro Soto. 2021. Inspecting the concept knowledge graph encoded by modern language models. In *Findings of ACL-IJCNLP*, pages 2984–3000.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.
- Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. GenericsKB: A knowledge base of generic statements. *CoRR*, abs/2005.00660.
- Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh Tensorflow. Zenodo.
- Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from BERT. In *Proceedings of AAAI-IAAI-EAAI*, pages 7456–7463.
- Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of CoNLL*, pages 677–685.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*, pages 1877–1901.
- 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 ACL-IJCNLP*, pages 1860–1874.
- Susan Carey. 1991. Knowledge acquisition: Enrichment or conceptual change. *The epigenesis of mind: Essays on biology and cognition*, pages 257–291.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyuntaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.
- Jessica A. Collins and Ingrid R. Olson. 2014. Knowledge is power: How conceptual knowledge transforms visual cognition. *Psychonomic Bulletin & Review*, 21:843–860.
- David Alan Cruse. 1986. *Lexical semantics*. Cambridge university press.
- Fahim Dalvi, Abdul Rafee Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. 2021. Discovering latent concepts learned in BERT. In *Proceedings of ICLR*.
- Stefan Decker, Sergey Melnik, Frank van Harmelen, Dieter Fensel, Michel C. A. Klein, Jeen Broekstra, Michael Erdmann, and Ian Horrocks. 2000. The semantic web: The roles of XML and RDF. *IEEE Internet Comput.*, 4(5):63–74.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- Ramanathan V Guha, Dan Brickley, and Steve Macbeth. 2016. Schema.org: Evolution of structured data on the web. *Communications of the ACM*, 59(2):44–51.
- Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, et al. 2021. Pre-trained models: Past, present and future. *Proceedings of AI Open*.
- Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.
- John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of NAACL-HLT*, pages 4129–4138.

- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. **Simlex-999: Evaluating semantic models with (genuine) similarity estimation.** *Comput. Linguistics*, 41(4):665–695.
- Lei Ji, Yujing Wang, Botian Shi, Dawei Zhang, Zhongyuan Wang, and Jun Yan. 2019. **Microsoft concept graph: Mining semantic concepts for short text understanding.** *Data Intelligence*, 1(3):238–270.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. **Survey of hallucination in natural language generation.** *CoRR*, abs/2202.03629.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. **How can we know what language models know?** *Trans. Assoc. Comput. Linguistics*, 8:423–438.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. **Scaling laws for neural language models.** *arXiv preprint arXiv:2001.08361*.
- Juan J. Lastra-Díaz, Josu Goikoetxea, Mohamed Ali Hadj Taieb, Ana García-Serrano, Mohamed Ben Aouicha, and Eneko Agirre. 2019. **A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art.** *Eng. Appl. Artif. Intell.*, 85:645–665.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. **The power of scale for parameter-efficient prompt tuning.** In *Proceedings of EMNLP*, pages 3045–3059.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. **BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.** In *Proceedings of ACL*, pages 7871–7880.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. **Linguistic knowledge and transferability of contextual representations.** In *Proceedings of NAACL-HLT*, pages 1073–1094.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. **Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.** *CoRR*, abs/2107.13586.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. **GPT understands, too.** *CoRR*, abs/2103.10385.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. **RoBERTa: A robustly optimized BERT pretraining approach.** *CoRR*, abs/1907.11692.
- Xin Lv, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. **Differentiating concepts and instances for knowledge graph embedding.** In *Proceedings of EMNLP*, pages 1971–1979.
- Deborah L McGuinness, Frank Van Harmelen, et al. 2004. **Owl web ontology language overview.** *W3C recommendation*, 10(10):2004.
- Mchaput. 2016. **Mchaput/whoosh: Pure-python full-text search library.** GitHub.
- Julian Michael, Jan A. Botha, and Ian Tenney. 2020. **Asking without telling: Exploring latent ontologies in contextual representations.** In *Proceedings of EMNLP*, pages 6792–6812.
- Gregory Murphy. 2004. **The big book of concepts.** MIT press.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. **Glove: Global vectors for word representation.** In *Proceedings of EMNLP*, pages 1532–1543.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. **Language models as knowledge bases?** In *Proceedings of EMNLP-IJCNLP*, pages 2463–2473.
- Lutz Prechelt. 1996. **Early stopping-but when?** In Genevieve B. Orr and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, pages 55–69. Springer.
- Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. **Pre-trained models for natural language processing: A survey.** *Science China Technological Sciences*, 63(10):1872–1897.
- 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.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. **Exploring the limits of transfer learning with a unified text-to-text transformer.** *J. Mach. Learn. Res.*, 21:140:1–140:67.
- Ehud Reiter. 2018. **Hallucination in Neural NLG.** Ehud Reiter’s Blog.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. **How much knowledge can you pack into the parameters of a language model?** In *Proceedings of EMNLP*, pages 5418–5426.

- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. *Okapi at TREC-3*. *Nist Special Publication Sp*, 109:109.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. *Object Hallucination in Image Captioning*. In *Proceedings of EMNLP*, pages 4035–4045.
- John F Sowa. 1976. *Conceptual graphs for a data base interface*. *IBM Journal of Research and Development*, 20(4):336–357.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. *Energy and policy considerations for deep learning in NLP*. In *Proceedings of ACL*, pages 3645–3650.
- Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. *Yago: a core of semantic knowledge*. In *Proceedings of WWW*, pages 697–706.
- Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. *oLMpics - On what Language Model Pre-training Captures*. *Trans. Assoc. Comput. Linguistics*, 8:743–758.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. *BERT Rediscovers the Classical NLP Pipeline*. In *Proceedings of ACL*, pages 4593–4601.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. *What do you learn from context? Probing for sentence structure in contextualized word representations*. In *Proceedings of ICLR*.
- Denny Vrandečić. 2012. *Wikidata: A new platform for collaborative data collection*. In *Proceedings of WWW*, pages 1063–1064.
- Denny Vrandečić and Markus Krötzsch. 2014. *Wikidata: A free collaborative knowledgebase*. *Communications of the ACM*, 57(10):78–85.
- Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. *Probing Pretrained Language Models for Lexical Semantics*. In *Proceedings of EMNLP*, pages 7222–7240.
- Sandra R. Waxman and Dana Markow. 1995. *Words as Invitations to Form Categories: Evidence from 12- to 13-Month-Old Infants*. *Cognitive Psychology*, 29:257–302.
- Michele Wellsby and Penny M. Pexman. 2014. *Developing embodied cognition: Insights from children’s concepts and language processing*. *Frontiers in Psychology*, 5.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pieric Cistac, Tim Rault, Remi 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 Rush. 2020. *Transformers: State-of-the-art natural language processing*. In *Proceedings of EMNLP*, pages 38–45.
- Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. *Probbase: A probabilistic taxonomy for text understanding*. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492.
- Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. *Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models*. In *Proceedings of ACL*, pages 1–9.
- Kaisheng Zeng, Chengjiang Li, Yan Qi, Xin Lv, Lei Hou, Guozheng Peng, Juanzi Li, and Ling Feng. 2021. *Encoding the meaning triangle (object, entity, and concept) as the semantic foundation for entity alignment*. In *Proceedings of WISE*, pages 227–241.
- Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. *Factual probing is [MASK]: Learning vs. learning to recall*. In *Proceedings of NAACL-HLT*, pages 5017–5033.
- Jie Zhou, Shengding Hu, Xin Lv, Cheng Yang, Zhiyuan Liu, Wei Xu, Jie Jiang, Juanzi Li, and Maosong Sun. 2021. *KACC: A multi-task benchmark for knowledge abstraction, concretization and completion*. In *Findings of ACL-IJCNLP*, pages 1751–1763.

## Appendices

Model	model_name
BERT <sub>SMALL</sub>	prajjwal1/bert-small
BERT <sub>MEDIUM</sub>	prajjwal1/bert-medium
BERT <sub>BASE</sub>	bert-base-uncased
BERT <sub>LARGE</sub>	bert-large-uncased
RoBERTa <sub>BASE</sub>	roberta-base
GPT-2 <sub>BASE</sub>	gpt2
GPT-2 <sub>MEDIUM</sub>	gpt2-medium
GPT-2 <sub>LARGE</sub>	gpt2-large
GPT-2 <sub>XL</sub>	gpt2-xl
GPT-Neo <sub>125M</sub>	EleutherAI/gpt-neo-125M
BART <sub>BASE</sub>	facebook/bart-base
T5 <sub>SMALL</sub>	t5-small
T5 <sub>BASE</sub>	t5-base
T5 <sub>LARGE</sub>	t5-large
T5 <sub>3B</sub>	t5-3b
T5 <sub>11B</sub>	t5-11b

Table 7: The corresponding model\_names in Transformers library (Wolf et al., 2020) for different PLMs.

## A Implementation Details

We use the implementation code and pre-trained parameters of PLMs released in HuggingFace Transformers library (Wolf et al., 2020) to run our experiments. The model\_names we used in Transformers for different PLMs are shown in Table 7. We run experiments for large models (T5<sub>3B</sub>, and T5<sub>11B</sub>) on NVIDIA V100 GPUs, which approximately consumes 160 GPU hours, and the other PLMs on Nvidia GEFORCE RTX 3090 GPUs, which consumes about 300 GPU hours. We will introduce the implementation details for zero-shot probing (appendix A.1), linear probing (appendix A.2), and fine-tuning (appendix A.3).

### A.1 Zero-Shot Probing

As mentioned in § 3.2, we take different text parts of the prompts into scoring calculation. Table 8 shows the text parts used by various PLMs to score prompts on the three datasets.

### A.2 Linear Probing

We use the final outputs of specific tokens as the features extracted by PLMs: [CLS] for BERT; <s> for RoBERTa; the last token for GPT-2, GPT-Neo, and BART; the first token for T5. We then tune a lightweight linear classifier on the fixed features for BERT, RoBERTa, GPT-2, GPT-Neo, BART and tune the final vocabulary classification head for T5. Moreover, we reformulate the original instances into the text-to-text format for T5, and the input and output formats are shown in Table 9.

Model	CSJ	CPJ	CiC
BERT <sub>BASE</sub>	Query Entity	Concept	All
RoBERTa <sub>BASE</sub>	Query Entity	Concept	Concept
GPT-2 <sub>BASE</sub>	All	All	Concept
GPT-Neo <sub>125M</sub>	All	Concept	Concept
BART <sub>BASE</sub>	Query Entity	Concept	Concept
T5 <sub>BASE</sub>	Query Entity	Concept	All

Table 8: The text parts used to calculate scores of prompts in zero-shot probing on the three datasets. All: use the negative perplexities of prompts as scores. The meanings of the other text parts are shown in Figure 2.

**Hyperparameters** We set the learning rate as  $1 \times 10^{-3}$  and apply early stopping (Prechelt, 1996) on the accuracy on the development dataset with a patience of 20 epochs. We keep the other hyperparameters the same as in Table 10.

### A.3 Fine-Tuning

We follow the fine-tuning methods in original papers to fine-tune BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), GPT-2 (Radford et al., 2019), GPT-Neo (Black et al., 2021), and BART (Lewis et al., 2020). As in appendix A.2, we reformulate the original instances into the text-to-text format for T5 (Raffel et al., 2020), and the input and output formats are shown in Table 9.

**Hyperparameters** We follow the hyperparameters mostly used in previous literature. The hyperparameters are shown in Table 10. And we apply early stopping (Prechelt, 1996) on the accuracy on the development dataset.

### Parameter-efficient Tuning for Big Models

Due to the limits of computation, we consider the parameter-efficient tuning for models with more than 2.5 billion parameters (T5<sub>3B</sub> and T5<sub>11B</sub>). Previous works (He et al., 2021) have proven that parameter-efficient tuning methods can save GPU memory, accelerate training for PLMs, and achieve comparable performance to fine-tuning all parameters, especially at large scales. Therefore, we adopt BitFit (Zaken et al., 2022) implemented by Open-Delta<sup>10</sup> to fine-tune big models.

## B More Discussions on Experimental Results

In the section, we discuss some detailed and interesting observations.

<sup>10</sup><https://github.com/thunlp/OpenDelta>

Conceptual Similarity Judgment
<b>Original Query:</b> Inter Milan
<b>Original Candidates:</b> Milan, Milan Fashion Week, Pohang Steelers, Series A
<b>Original Label:</b> Pohang Steelers
<b>Processed Input:</b> choose the most similar entity to Inter Milan: (A) Milan, (B) Milan Fashion Week, (C) Pohang Steelers, (D) Series A.
<b>Processed Label:</b> C
Conceptual Property Judgment
<b>Original Statement:</b> Mammals raise their young on milk.
<b>Original Label:</b> True
<b>Processed Input:</b> verify: Mammals raise their young on milk.
<b>Processed Label:</b> true
Conceptualization in Contexts
<b>Original Context:</b> Dolly is running on the grassland.
<b>Concept Chain:</b> Horse → Mammal → Animal
<b>Original Label:</b> Animal
<b>Processed Input:</b> select concept: <entity> Dolly </entity> is running on the grassland. Select a contextually related concept for Dolly from (A) Horse, (B) Mammal, (C) Animal.
<b>Processed Label:</b> C

Table 9: The input and output format used to linear probe and fine-tune T5 on the three datasets.

	CSJ		CPJ		CiC	
	The Others	T5	The Others	T5	The Others	T5
Learning Rate	$3 \times 10^{-5}$	$5 \times 10^{-5}$	$3 \times 10^{-5}$	$5 \times 10^{-5}$	$3 \times 10^{-5}$	$5 \times 10^{-5}$
Weight Decay	$1 \times 10^{-5}$					
Batch Size	4	16	64	32	4	16
Warmup Rate	0.1	0.1	0.1	0.1	0.1	0.1

Table 10: Hyperparameters used to fine-tune PLMs on COPEN.

	Query Entity	CSJ			CPJ			CiC	
		Candidate Entity	All	Concept	Answer	All	Concept	All	Concept
BERT <sub>SMALL</sub>	15.0	6.5	8.1	50.7	48.5	51.5	31.9	35.1	
BERT <sub>MEDIUM</sub>	16.8	7.2	10.0	49.3	46.7	49.2	29.6	33.3	
BERT <sub>BASE</sub>	20.3	7.5	11.3	49.4	47.2	49.2	32.6	37.6	
BERT <sub>LARGE</sub>	22.3	8.2	13.4	50.5	47.6	50.4	31.1	36.9	
RoBERTa <sub>BASE</sub>	15.5	5.1	10.0	49.2	46.7	47.6	31.4	25.5	
GPT-2 <sub>BASE</sub>	2.9	6.6	7.9	49.4	48.4	51.5	32.3	31.1	
GPT-2 <sub>MEDIUM</sub>	3.7	8.6	10.5	52.0	47.2	47.2	30.3	32.0	
GPT-2 <sub>LARGE</sub>	4.6	9.0	11.3	51.8	47.3	47.2	34.3	33.8	
GPT-2 <sub>XL</sub>	3.9	9.6	11.7	50.7	47.2	47.1	35.3	37.0	
GPT-Neo <sub>125M</sub>	2.6	6.6	7.9	52.2	47.2	47.6	32.6	28.8	
BART <sub>BASE</sub>	14.4	5.0	7.1	48.7	48.4	48.0	33.6	27.4	
T5 <sub>SMALL</sub>	11.6	5.4	6.5	52.5	47.6	53.2	34.9	40.1	
T5 <sub>BASE</sub>	15.2	7.2	10.3	55.9	47.2	49.5	39.1	42.3	
T5 <sub>LARGE</sub>	20.9	7.8	14.0	52.4	47.2	49.8	40.5	42.6	
T5 <sub>3B</sub>	19.2	7.9	14.1	49.4	47.7	49.4	38.6	47.0	
T5 <sub>11B</sub>	24.8	7.8	14.5	46.7	46.7	49.9	37.2	41.3	

Table 11: Overall zero-shot probing accuracies (%) of using different text parts to score prompts on COPEN.

Model	Linear Probing					Fine-tuning				
	Seed=42	Seed=43	Seed=44	Mean	Std	Seed=42	Seed=43	Seed=44	Mean	Std
Conceptual Similarity Judgment										
BERT <sub>SMALL</sub>	9.1	8.2	8.9	8.7	0.37	17.6	17.1	19.2	18.0	0.91
BERT <sub>MEDIUM</sub>	13.1	12.3	13.1	12.8	0.35	20.3	21.1	21.6	21.0	0.57
BERT <sub>BASE</sub>	16.3	16.3	15.8	16.1	0.21	28.5	26.6	26.9	27.3	0.86
BERT <sub>LARGE</sub>	16.5	16.9	17.3	16.9	0.31	28.7	30.2	29.5	29.5	0.61
RoBERTa <sub>BASE</sub>	11.8	12.0	12.3	12.0	0.21	22.8	21.6	22.4	22.3	0.51
GPT-2 <sub>BASE</sub>	4.6	4.1	4.1	4.3	0.24	19.7	20.1	20.3	20.1	0.23
GPT-2 <sub>MEDIUM</sub>	5.3	5.2	5.2	5.2	0.02	24.9	22.2	23.0	23.4	1.15
GPT-2 <sub>LARGE</sub>	4.0	6.8	5.6	5.5	1.13	22.2	24.0	23.4	23.2	0.77
GPT-2 <sub>XL</sub>	7.8	15.0	10.1	11.0	3.00	25.9	24.2	25.7	25.3	0.75
GPT-Neo <sub>125M</sub>	11.1	10.7	11.2	11.0	0.20	18.8	18.4	17.8	18.3	0.42
BART <sub>BASE</sub>	8.5	8.3	8.4	8.4	0.10	20.4	21.0	21.7	21.0	0.50
T5 <sub>SMALL</sub>	4.8	4.8	4.7	4.8	0.05	10.1	17.6	6.9	11.5	4.48
T5 <sub>BASE</sub>	5.2	4.8	4.7	4.9	0.21	27.4	27.5	28.7	27.9	0.60
T5 <sub>LARGE</sub>	4.7	4.9	4.8	4.8	0.09	31.0	33.4	32.5	32.3	1.01
T5 <sub>3B</sub>	5.0	4.9	5.2	5.0	0.11	41.0	40.6	42.0	41.2	0.61
T5 <sub>11B</sub>	4.7	4.7	4.7	4.7	0.01	43.7	43.6	43.8	43.7	0.08
Conceptual Property Judgment										
BERT <sub>SMALL</sub>	57.8	58.8	57.8	58.1	0.47	66.3	66.5	67.2	66.7	0.39
BERT <sub>MEDIUM</sub>	58.2	59.6	58.5	58.8	0.59	66.7	67.5	67.3	67.2	0.35
BERT <sub>BASE</sub>	61.2	61.9	61.5	61.6	0.28	66.8	68.3	69.2	68.1	0.98
BERT <sub>LARGE</sub>	61.6	61.7	59.0	60.8	1.26	67.8	69.6	71.2	69.5	1.41
RoBERTa <sub>BASE</sub>	61.7	62.0	61.9	61.9	0.13	71.4	72.7	71.8	72.0	0.54
GPT-2 <sub>BASE</sub>	65.2	63.3	66.0	64.8	1.14	71.3	69.5	70.5	70.4	0.72
GPT-2 <sub>MEDIUM</sub>	67.0	67.4	67.4	67.3	0.17	73.0	68.6	72.9	71.5	2.07
GPT-2 <sub>LARGE</sub>	66.2	67.8	66.8	66.9	0.62	74.5	72.7	73.4	73.5	0.74
GPT-2 <sub>XL</sub>	67.8	68.1	68.6	68.2	0.36	74.5	75.1	74.7	74.8	0.22
GPT-Neo <sub>125M</sub>	61.9	62.4	62.1	62.2	0.21	68.9	68.4	67.4	68.2	0.62
BART <sub>BASE</sub>	58.8	58.2	58.7	58.5	0.27	68.5	69.2	67.1	68.2	0.86
T5 <sub>SMALL</sub>	67.7	67.2	65.0	66.6	1.18	71.3	72.2	72.1	71.9	0.40
T5 <sub>BASE</sub>	67.3	66.8	66.8	66.9	0.25	72.6	72.1	72.8	72.5	0.28
T5 <sub>LARGE</sub>	68.9	69.7	69.3	69.3	0.33	72.5	73.4	75.2	73.7	1.10
T5 <sub>3B</sub>	69.2	69.7	69.5	69.5	0.22	76.6	76.6	76.2	76.4	0.19
T5 <sub>11B</sub>	67.3	66.5	66.0	66.6	0.53	78.2	78.3	79.2	78.6	0.46
Conceptualization in Contexts										
BERT <sub>SMALL</sub>	32.4	32.7	33.3	32.8	0.38	44.6	47.0	48.4	46.6	1.55
BERT <sub>MEDIUM</sub>	31.6	31.2	31.1	31.3	0.22	49.4	49.1	49.8	49.4	0.31
BERT <sub>BASE</sub>	33.6	34.5	35.0	34.3	0.59	49.3	48.9	50.3	49.5	0.60
BERT <sub>LARGE</sub>	35.4	38.9	35.3	36.6	1.67	50.7	53.0	51.6	51.8	0.92
RoBERTa <sub>BASE</sub>	27.3	32.0	30.7	30.0	1.98	51.3	52.6	53.8	52.6	1.02
GPT-2 <sub>BASE</sub>	31.7	36.7	35.1	34.5	2.08	54.0	54.2	54.3	54.2	0.12
GPT-2 <sub>MEDIUM</sub>	29.3	25.6	29.1	28.0	1.69	54.6	54.5	54.9	54.7	0.14
GPT-2 <sub>LARGE</sub>	32.8	28.8	33.7	31.8	2.16	53.4	52.7	53.6	53.3	0.36
GPT-2 <sub>XL</sub>	27.7	32.2	29.9	29.9	1.83	52.6	54.4	54.4	53.8	0.88
GPT-Neo <sub>125M</sub>	38.9	38.9	40.9	39.6	0.93	47.6	47.0	47.5	47.4	0.25
BART <sub>BASE</sub>	44.1	42.1	44.9	43.7	1.19	50.8	49.7	53.5	51.3	1.56
T5 <sub>SMALL</sub>	25.7	26.1	24.9	25.6	0.53	43.5	44.4	45.0	44.3	0.64
T5 <sub>BASE</sub>	25.5	23.9	24.7	24.7	0.66	53.2	53.3	52.9	53.2	0.18
T5 <sub>LARGE</sub>	24.3	24.3	25.3	24.6	0.49	52.4	56.9	57.2	55.5	2.21
T5 <sub>3B</sub>	26.7	27.5	26.8	27.0	0.35	59.2	57.5	55.9	57.5	1.35
T5 <sub>11B</sub>	25.1	26.6	26.4	26.0	0.66	56.7	58.7	56.5	57.3	0.97

Table 12: Overall linear probing and fine-tuning accuracies (%) of all PLMs on COPEN. We run experiments 3 times using three seeds: 42, 43, 44. Mean: mean accuracy of the three trials; Std: standard deviation.

**Comparison of Pre-training Method** In Figure 2, we can observe that: (1) For PLMs using the same architecture, T5 generally outperforms BART, and BERT generally outperforms RoBERTa. The differences may come from the different pre-training corpora. (2) Autoregressive LMs (GPT-2, GPT-Neo) perform worse on CSJ, which is consistent with the observations on factual knowledge probing (Liu et al., 2021b). As we are the first to study conceptual knowledge in PLMs, we focus on the general question “to what extent do current PLMs understand conceptual knowledge?” and provide more general conclusions in the paper. We leave the detailed and in-depth analysis of a specific PLM, e.g., layer-wise analysis (Dalvi et al., 2021), in future works.

**Comparison of Probing Method** Intuitively, zero-shot probing reflects the *lower bound* of PLMs’ knowledge (Jiang et al., 2020), while linear probing learns a task-specific linear classifier and performs better than zero-shot probing, and fine-tuning reflects the *upper bound* of PLMs’ knowledge. However, as shown in Figure 2, linear probing sometimes underperforms zero-shot probing, especially in CSJ and chain-level CPJ. The reason may be that the concepts used for training and testing are disjoint, and linear probing involves trainable parameters, which may learn spurious or shallow correlations on training sets and hence struggles on generalization. Meanwhile, fine-tuning still performs poorly, which demonstrates that existing PLMs systematically lack conceptual knowledge.

**Comparison of Instance-Level and Chain-Level CPJ** For chain-level, BERT performs the best, but for instance-level performs worse than T5. The reason may be that BERT better understands concept transitivity (i.e., making more consistent predictions) but stores fewer conceptual properties overall. A thorough and comprehensive analysis is needed on this phenomenon and we leave it in future works.

## C Additional Experimental Results

Table 11 shows overall zero-shot probing results on COPEN. The experimental results of linear probing and fine-tuning are obtained at 3 random trials using seeds 42, 43, 44. Table 12 shows overall linear probing and fine-tuning results on COPEN. And we provide additional results for the analytical experiments: analysis of *conceptual hallucination*

Model	Disambiguation	Wrong Level
BERT <sub>BASE</sub>	29.0%	71.0%
RoBERTa <sub>BASE</sub>	12.8%	87.2%
GPT-2 <sub>BASE</sub>	12.5%	87.5%
GPT-Neo <sub>125M</sub>	11.9%	88.1%
BART <sub>BASE</sub>	11.5%	88.5%
T5 <sub>BASE</sub>	32.0%	68.0%

Table 13: The proportion of different error types of zero-shot probing results on the CiC dataset. We only consider the entities with more than one concept chain.

on the CPJ dataset (appendix C.1), error analysis on the CiC dataset (appendix C.2), and analysis on avoiding dataset artifacts (appendix C.3).

### C.1 Conceptual Hallucination on CPJ

Figure 5 shows the false negative rates on subsets with different BM25 scores for various PLMs. We can observe that the false positive rates, which indicates conceptual hallucination, have strong positive correlations to the BM25 scores, which indicates word co-occurrence.

### C.2 Error Analysis on CiC

Table 13 shows the proportions of different error types. We can observe that in most wrong predictions, PLMs select concepts of wrong levels. It indicates that PLMs lack a comprehensive understanding of concept hierarchy and fail to conceptualize entities according to contexts.

### C.3 Analysis on Avoiding Dataset Artifacts

Dataset artifacts leak shallow information and cause the PLMs to learn spurious correlations rather than exhibit inner knowledge. When construct COPEN, we avoid two kinds of artifacts:

**Lexical Overlap** means that the query and the answer have word overlap, which may enable PLMs to make correct predictions using spurious correlations without the correct knowledge. For example, in CSJ, if the query entity is Stanford University and the answer entity is University of California; in CiC, if the context is *She graduated from Stanford University* and the answer concept is University; they have lexical overlap.

We conduct experiments on the data with lexical overlap. As shown in Table 14, on the data with lexical overlap, PLMs perform much better. But this should be interpreted as they learn shallow clues leaked by artifacts since they cannot achieve similar performance on data without lexical over-

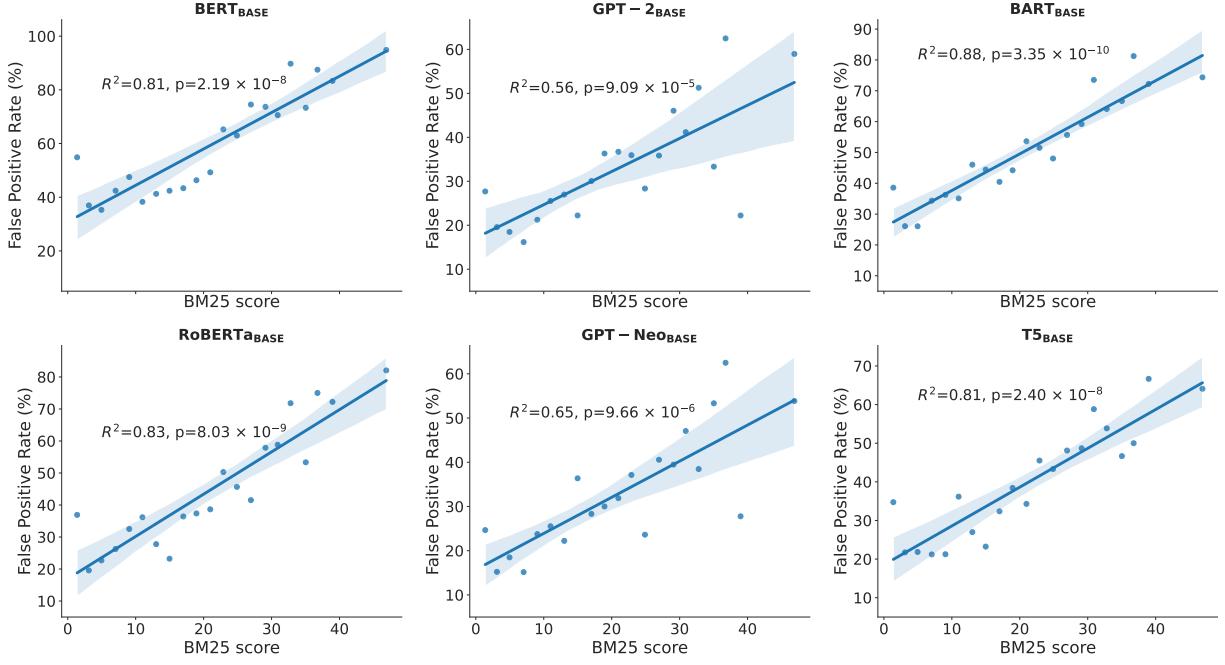


Figure 5: The false positive rate of various PLMs’ fine-tuning results on negative instances of the CPJ dataset with different BM25 scores.

Model	CSJ		CiC	
	w/ LO	w/o LO	w/ LO	w/o LO
BERT <sub>BASE</sub>	68.9	20.3	52.5	37.6
RoBERTa <sub>BASE</sub>	62.2	15.5	48.5	31.4
GPT-2 <sub>BASE</sub>	34.2	7.9	43.8	32.3
GPT-Neo <sub>125M</sub>	34.0	7.9	52.4	32.6
BART <sub>BASE</sub>	75.9	14.4	53.2	33.6
T5 <sub>BASE</sub>	69.2	15.2	62.7	42.3

Table 14: Zero-shot probing accuracies (%) of PLMs on data with lexical overlap (w/ LO) and without lexical overlap (w/o LO). We collect 688 and 1,200 instances with lexical overlap for CSJ and CiC, respectively.

lap. Hence, we filter out all instances with lexical overlap in COPEN to avoid this kind of artifact.

**Concept Overlap** is that the same concepts show up in both training and test datasets, which may leak conceptual knowledge, i.e., the PLMs may learn some knowledge from training data. In COPEN, as mentioned in § 2.1, we split different top-level concepts and their subconcepts into different sub-datasets, so as to avoid concept overlap. To empirically show the influence of concept overlap, we randomly re-split the datasets into same-size training, development, and test sets and see the fine-tuning performance on the new split.

The results of fine-tuning BERT are shown in Figure 6, and the results of fine-tuning and linear

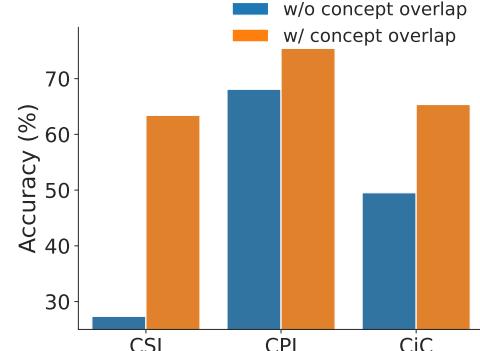


Figure 6: Fine-tuning accuracies of BERT<sub>BASE</sub> on data with and without concept overlap.

probing for all PLMs are shown in Table 15. Fine-tuning on datasets with concept overlap achieves much higher accuracies, especially on CSJ. It indicates that if we do not avoid concept overlap, PLMs can easily learn conceptual knowledge from training data and lead to false optimistic conclusions.

## D COPEN

We provide a detailed introduction to COPEN.

### D.1 COPEN Taxonomy

**Disjoint Concepts** We divide all the concepts into two disjoint sets: one set containing 11 top-level concepts together with all their sub-concepts for constructing training and development datasets,

Model	CSJ		CPJ		CiC	
	w/ CO	w/o CO	w/ CO	w/o CO	w/ CO	w/o CO
Linear Probing						
BERT <sub>BASE</sub>	20.0	16.1	64.1	61.6	46.5	34.3
RoBERTa <sub>BASE</sub>	12.3	12.0	65.9	61.9	45.4	30.0
GPT-2 <sub>BASE</sub>	5.2	4.3	67.2	64.8	39.0	34.5
GPT-Neo <sub>125M</sub>	15.4	11.0	64.6	62.2	58.3	39.6
BART <sub>BASE</sub>	9.4	8.4	62.6	58.5	50.2	43.7
T5 <sub>BASE</sub>	4.7	4.9	68.8	66.9	33.9	24.7
Fine-tuning						
BERT <sub>BASE</sub>	63.4	27.3	75.4	68.1	65.4	49.5
RoBERTa <sub>BASE</sub>	61.0	22.3	77.0	72.0	66.6	52.6
GPT-2 <sub>BASE</sub>	49.9	20.1	72.7	70.4	65.4	54.2
GPT-Neo <sub>125M</sub>	44.3	18.3	71.2	68.2	62.5	47.4
BART <sub>BASE</sub>	54.7	21.0	73.1	68.2	67.4	51.3
T5 <sub>BASE</sub>	50.6	27.9	77.6	72.5	67.6	53.2

Table 15: Accuracies (%) of linear probing and fine-tuning on data with concept overlap (w/ CO) and without concept overlap (w/o CO).

	#Concepts	Top-Level Concepts
Training& Development	248	Organisation, Name, Award, MeanOfTransportation, Colour, Language, Person, Holiday, Work, Currency, EthnicGroup
Testing	198	AnatomicalStructure, Species, Food, Event, TimePeriod, ChemicalSubstance, Place, Device, Disease, Activity, Biomolecule, SportsSeason

Table 16: The top-level concepts and the number of concepts used for training, development, and testing.

and the other set containing the other concepts for testing datasets. As shown in Table 16, there are 248 concepts including 11 top-level concepts for training and development datasets and 198 concepts including 12 top-level concepts for testing.

**Concept Hierarchy** We present the concepts for training and development datasets in Figure 7 and the concepts for testing datasets in Figure 8. Object is a virtual concept for visualization and is not included in the overall 446 concepts.

## D.2 Concept Similarity Judgment

**Human Performance** We sample 1,000 instances from the testing dataset and invite annotators with no linguistic background to perform the CSJ task. All the annotators are trained with a few instances before the evaluation.

**Co-occurrence-based Filtering** We filter out instances of which query entities and answer entities have a high association, which are estimated by cosine similarity of their Glove word embeddings. Specifically, for a query entity, we sample 5 answer entities and select the entity with the lowest association with the query entity as the answer entity. Then we choose distractor entities iteratively fol-

lowing the rules: (1) Sample a distractor entity, if the entity has a higher association with the query entity than the answer entity, then select the distractor entity as a candidate entity. (2) If not, select the distractor entity as a candidate entity with a 20% probability, otherwise start the next iteration until the number of distractor entities reaches 20.

## D.3 Conceptual Property Judgment

**Human Annotation** We invite annotators with no linguistics background to check whether the instances are correctly labeled, grammatically correct, and describing concept properties. All annotators are well-trained and required to pass a qualification before the annotation. The instances originally labeled as false are annotated 4 times, and the other instances are annotated once. During the annotation, an author of the paper and another experienced annotator separately sample 10% of the instances to check the quality of annotation.

The acceptance criterion of the annotation is that the percentage of obvious annotation errors in the sampled instances (e.g., label the statement *The sun has two eyes* as true) does not exceed 3%, and the inter-annotator agreement rates exceed 85% for the instances annotated 4 times. Major voted results of

the instances annotated 4 times together with the instances annotated once constitute the CPJ dataset.

**Human Performance** We use the 2,159 instances that are annotated 4 times in the testing dataset to evaluate human performance. We conduct a 4-round evaluation: take the major voted results of 3 annotators as labels and the other one as human predictions to calculate the accuracy of the round. The mean accuracy of 4 rounds is reported as the human accuracy on the CPJ dataset.

#### D.4 Conceptualization in Contexts

**Human Annotation** We invite annotators with no linguistics background to annotate the dataset. To ensure quality, all annotators are well-trained and required to pass a qualification before the annotation. All instances are annotated four times. Moreover, during the annotation, an author of the paper and another experienced annotator separately sample 10% of the examples to check the quality of annotation. The acceptance criterion of the annotation is that the percentage of obvious annotation errors (e.g., Select Horse for Dolly according to the context *Dolly is running on the grassland.*) does not exceed 3%, and the inter-annotator agreement rates exceed 80%. Major voted results of the 4 annotated results constitute the final CiC dataset.

**Human Performance** We use all instances in the testing dataset, which are annotated 4 times, to evaluate human performance. We conduct a 4-round evaluation: take the major voted results of 3 annotators as labels and the other one as human predictions to calculate the accuracy of the round. The mean accuracy of 4 rounds is the human accuracy.

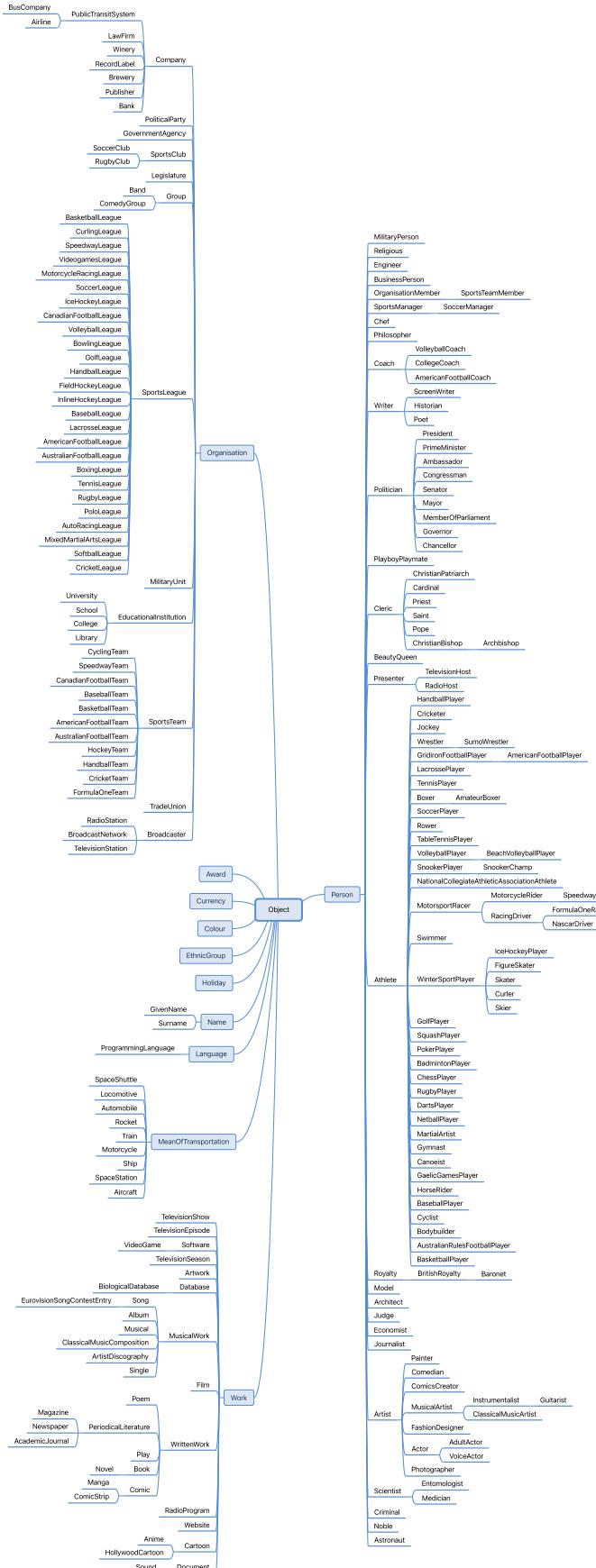


Figure 7: Concept taxonomy for training and development datasets. **Object** is a virtual concept without annotated instances.

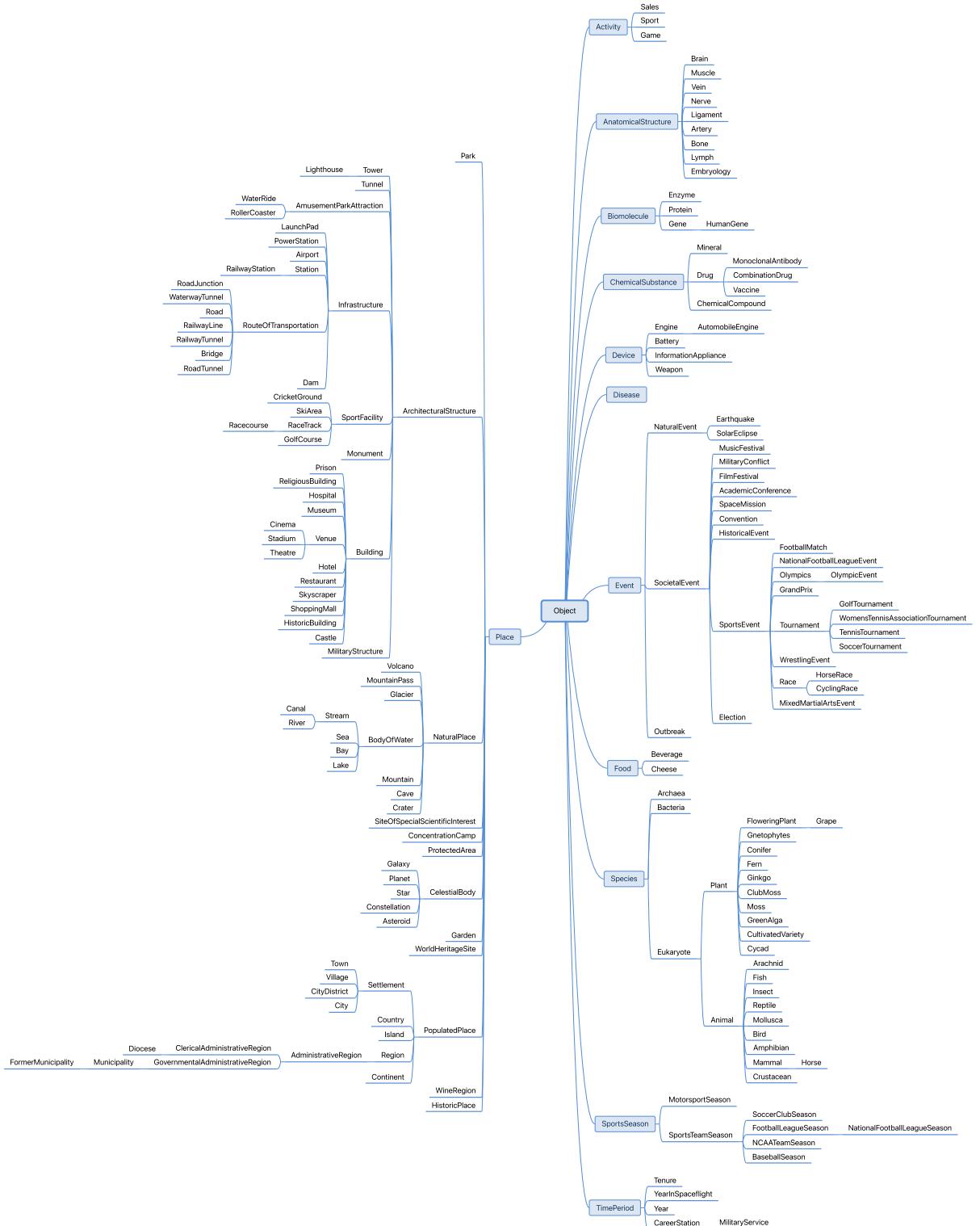


Figure 8: Concept taxonomy for testing datasets. **Object** is a virtual concept without annotated instances.