Is a Peeled Apple Still Red? Evaluating LLMs' Ability for Conceptual Combination with Property Type

Seokwon Song[⋆,1] **Taehyun Lee**[⋆,1] **Jaewoo Ahn**[1]
Jae Hyuk Sung[2,♯] **Gunhee Kim**[1]
[1]Seoul National University [2]Korea University
{seokwon.song, taehyun.lee, jaewoo.ahn}@vision.snu.ac.kr, okaybody10@korea.ac.kr
gunhee@snu.ac.kr

Abstract

Conceptual combination is a cognitive process that merges basic concepts, enabling the creation of complex expressions. During this process, the properties of combination (e.g., the whiteness of a peeled apple) can be inherited from basic concepts, newly emerge, or be canceled. However, previous studies have evaluated a limited set of properties and have not examined the generative process. To address this gap, we introduce the Conceptual Combination with Property Type dataset (CCPT), which consists of 12.3K annotated triplets of noun phrases, properties, and property types. Using CCPT, we establish three types of tasks to evaluate LLMs for conceptual combination thoroughly. Our key findings are threefold: (1) Our automatic metric grading property emergence and cancellation closely corresponds with human judgments. (2) LLMs, including OpenAI's o1, struggle to generate noun phrases which possess given emergent properties. (3) Our proposed method, inspired by cognitive psychology model that explains how relationships between concepts are formed, improves performances in all generative tasks. The dataset and experimental code are available at https://github.com/seokwon99/CCPT.git.

1 Introduction

The conceptual combination is a fundamental cognitive process that synthesizes multiple basic concepts into a novel concept (Wisniewski, 1997; Thagard, 1984; Coutanche et al., 2019; Ward, 2001). Combining concepts is potentially limitless and plays an important role in various fields such as engineering, science (Hampton, 1997), figurative language and literature (Ward, 2001). For example, 'a wilted flower' evokes our sadness because we internally compare a flower's former beauty and current feeble state. A skilled writer knows it and kindles diverse imagery through conceptual combinations. The potential for such conceptual combinations is vast, so large language models (LLMs) should be able to interpret or generate meaningful insights even for rare or novel combinations.

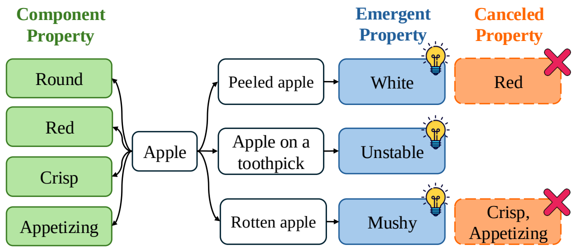


Figure 1: Three types of properties derived from conceptual combination with an example of "apple". Different concepts are formed by adding other concepts to "apple". The green properties are *component properties* of the basic concept "apple". The blue and orange are *emergent* and *canceled properties*, respectively.

Several works have explored concept knowledge of language models, as outlined in Table 1. Basic Concept Probing examines whether language models understand individual concepts at the entity (e.g., "horses are mammals") and event levels (e.g., "drinking tea leads to feeling refreshed"). However, this approach mainly focuses on single concepts rather than combinations. Noun Compound Interpretation studies how models generate plausible meanings for combinations of nouns (e.g., "dog house" means "a house for a dog"), but does not explore deeper properties of combinations. In conceptual combination works, Big-Bench-CC investigates emergent properties. FakeReef delves into a special type of canceled property (membership inference). Still, none of them addresses all of emergent, component, or canceled properties nor generative tasks related to creating combinations or property.

It is essential to understand not only properties that emerge from a combination, but also whether

⋆ Authors equally contributed.
♯ Work done during internship at Seoul National University.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 9028–9048
April 29 - May 4, 2025

DATASET	# Q	PROPERTY			MC
		COM	EME	CAN	
Basic Concept Probing					
BERT-IS-A (2022)	11,868	✓	✗	✗	✗
COPEN (2022)	10,624	✓	✗	✗	✗
PLM-Ontology (2023)	9,687	✓	✗	✗	✗
Noun Compound Interpretation					
SemEval-2013-Task4 (2013)	355	✗	✗	✗	✓
ChocolateBunny (2023)	105	✗	✗	✗	✓
COMPUN (2024)	918	✗	✗	✗	✓
Conceptual Combination					
BIG-bench-CC (2023)	103	△	✓	✗	✓
FakeReef (2024)	854	✓	✗	✓	✓
CCPT (ours)	12,315	✓	✓	✓	✓

Table 1: Comparison of our dataset with previous concept property probing and noun compound interpretation datasets. Each column denotes as follows. # Q: the number of questions with gold answers. COM, EME, and CAN denotes component, emergent and canceled property, respectively. MC: whether the combinations of multiple concepts are considered. In Big-bench-CC, △ indicates that component properties are present in a dataset as wrong options, but it does not specify what properties are component properties.

existing properties maintain or disappear (Springer and Murphy, 1992; Gagné et al., 2017; Estes and Glucksberg, 2020). Following Springer and Murphy (1992), we identify three types of properties in conceptual combination, which are illustrated using the example of an "apple" in Figure 1: (1) *Component property* refers to a property inherent to individual concepts (e.g., "red" for "apple on a toothpick"), (2) *Emergent property* is newly from the combination of concepts (e.g., "unstable" for "apple on a toothpick"), and (3) *Canceled property* is negated due to the combination (e.g., "crisp" for "rotten apple"). These three types form a complete set of properties related to conceptual combination since they indicate *already present*, *newly created*, and *newly disappeared*.

In this study, we aim to fill this gap by proposing the **C**onceptual **C**ombination with **P**roperty **T**ype (**CCPT**) dataset. Our dataset contains 12.3K annotated triplets of noun phrase comprised of two basic concepts, property and property type. The basic concepts in noun phrases are comprised of 9K unique uni-gram concepts from ConceptNet (Speer et al., 2017).

To evaluate LLMs' ability on conceptual combination, we introduce two generative and one classification tasks, detailed in Table 2 and §3.1. Based on the prior works in cognitive psychology and linguistics, each task is designed to evaluate cognitive processes related to conceptual combination as follows. (1) *Property induction*: imagine the property of noun phrase (Wilkenfeld and Ward, 2001; Estes and Ward, 2002), (2) *Noun phrase completion*: create a noun phrase to represent certain property, highlighted as a critical point of creativity (Kohn et al., 2011), and (3) *Property type prediction*: predict a type of property to identify the origin of property (Gagné et al., 2017).

Extensive experiments show that: (1) our automatic evaluation metric, based on the LLM-as-a-judge (Zheng et al., 2023) method to detect property emergence and cancellation scores, closely matches human evaluations; (2) current LLMs, including o1 (OpenAI, 2024c), struggle to create noun phrases with truly emergent properties, often defaulting to results where one component already has the target property; and (3) our proposed method, motivated by the spread activation model (Anderson, 1983; Collins and Loftus, 1975) in cognitive psychology, improves performance across all generative tasks, indicating that considering diverse relationships between concepts enhances conceptual combination.

2 Related Work

2.1 Conceptual Combination

Conceptual combination is a mental process of combining concepts. Unique features often arise from the combination, especially when the combinations of elements are unusual (Estes and Ward, 2002; Ward, 2007). People come up with novel ideas to merge conflicts; thus, it is a prominent part of creativity (Hampton, 1997). In scientific discovery, conceptual combination creates new scientific concepts (Thagard, 1984), such as "light wave." Even if nobody could observe light waves, the properties shared with waves, such as reflection, led to a theory that light is a wave, leading to much progress in science.

2.2 Conceptual Combination in NLP

In NLP, understanding conceptual combination has been studied in two different lines. For instance, interpreting "pet bird" as "a bird kept as a pet" exemplifies *noun compound interpretation*. Alternatively, imagining "pet bird"'s attributes like "probably lives in a cage" illustrates *property verification*.

Noun Compound Interpretation focuses on generating plausible meanings for noun-noun compounds, such as interpreting "dog house" as "a house for a dog". The goal is to resolve ambiguity and provide clear interpretations of compound nouns. While related to combining nouns, it mainly emphasizes understanding the relationships between the words (Hendrickx et al., 2013; Shwartz and Waterson, 2018; Shwartz and Dagan, 2019; Coil and Shwartz, 2023; Rambelli et al., 2024).

The second line of research focuses on the properties of conceptual combinations. BIG-bench authors (2023) introduced a benchmark for identifying emergent properties in conceptual combinations. However, their approach has limitations: (1) it only identifies emergent properties, (2) the benchmark is limited to multiple-choice selection tasks without generative tasks, and (3) it does not involve creating conceptual combinations. In contrast, our approach (1) uses property types as part of the constraints and (2) includes both generative tasks and multiple-choice selection tasks. Concurrent work (Ross et al., 2024) deeply explores LLM's understanding of modifier adjective-noun conceptual combination with a membership inference question such as 'Is a fake reef still a reef?'. The work shares interest with us about property cancellation. However, there's still a gap in evaluating LLMs for conceptual combination in generative way.

3 The CCPT Benchmark

3.1 Task Formulation

Before describing our task formulation, we begin with its key elements: $\{\mathcal{N}(\mathcal{H}, \mathcal{M}), \mathcal{P}, \mathcal{T}\}$, with two examples of "peeled apple" and "egg in the batter" for better understanding.

- **Noun phrase** $\mathcal{N}$ is noun phrase comprised of two concepts (e.g., "peeled apple" and "egg in the batter").
- **Head noun** $\mathcal{H}$ provides the central meaning of the combination $\mathcal{N}$ (e.g., "apple" and "egg").
- **Modifier** $\mathcal{M}$ adds additional meaning to the head noun $\mathcal{H}$ (e.g., "peeled" and "batter").
- **Property** $\mathcal{P}$ is attributed to the combination $\mathcal{C}$ (e.g., "white" for a peeled apple; "nutritious" for the egg in the batter).
- **Type of property** $\mathcal{T}$ introduces the origin of the property $\mathcal{P}$ (e.g., "emergent property" - "white" for a peeled apple, "component property" - "nutritious" for the egg in the batter)

Based on the above configuration, we devise three types of tasks by hiding one configuration and using the remaining configurations to predict it. This task formulation includes two generative tasks: property induction and noun phrase completion, and one classification task: property type prediction.

TASK 1: PROPERTY INDUCTION	
Inst.	Given a noun phrase and a property type, generate a property that satisfies both.
Input	Noun phrase: An apple on a toothpick Type: Emergent
Response1	Unstable
	➥ *"Apple on a toothpick" is unstable, but 'apple' or 'toothpick' is seldom related to it: good output (7/10)*
Response2	Delicious
	➥ *"Apple" is already delicious, and that makes 'apple on a tootkpick' also delicious: bad output (2/10)*
TASK 2: NOUN PHRASE COMPLETION	
Inst.	Given head noun, property and type, generate a noun phrase that satisfies both.
Input	Head noun: Apple Property: Rare Type: Emergent
Response1	A blue apple
	➥ *"A blue apple" has new emergent property 'rare' even if 'apple' is common and 'blue' is unrelated to 'rare': good output (8/10)*
Response2	"An overripe apple"
	➥ *"An overripe apple" is not considered rare at all: bad output (1/10)*
TASK 3: TYPE PREDICTION	
Inst.	Given noun phrase and property, choose a type of property. (1) Emergent (2) Canceled (3) Component (4) Others
Input	Noun phrase: A green apple Property: Good for health
Response1	(3) Component
	➥ *"Component property" is a good output, because 'apple' is already good for health.*

Table 2: Each example consists of a task instruction, input, and responses with their grading (Tasks 1 and 2) and correctness (Task 3). The explanations ➥ in italics are added by the authors for clarification. Note that LLMs are evaluated solely based on their responses.

3.1.1 Property Induction ($\mathcal{N}, \mathcal{T} \rightarrow \mathcal{P}$)

The interpretation of novel combinations by listing property has been a well-explored area in previous research on human cognition (Wilkenfeld and Ward, 2001; Estes and Ward, 2002), with a significant focus on how people comprehend novel expressions in sentences and discourse (Swinney et al., 2007). In this task, LLMs are instructed to identify the properties of combinations that align with the given property types (emergent, canceled property).

For example, in Table 2 (top), LLMs can identify an emergent property by finding a property that is not present in the individual concept but emerges in the noun phrase, such as "unstable" for "apple on a toothpick".

3.1.2 Noun Phrase Completion ($\mathcal{H}, \mathcal{P}, \mathcal{T} \rightarrow \mathcal{N}$)

Generating new concepts by combining existing ones is key to creativity (Kohn et al., 2011). In this task, LLMs generate noun phrases by adding modifiers to head nouns to represent emergent or canceled properties. For emergent properties, the modifier should not imply the property on its own, but the combination should. In contrast, for canceled properties, the modifier effectively negates the head noun's property.

For example, in Table 2 (middle), to represent the emergent property "rare" with the head noun "apple", LLMs may consider a noun phrase like "blue apple". The modifier "blue" does not directly suggest the property "rare", but when combined with "apple", it possesses the given property.

3.1.3 Property Type Prediction ($\mathcal{N}, \mathcal{P} \rightarrow \mathcal{T}$)

Understanding how combined concepts gain or lose certain properties is an essential process for concept theories (Gagné et al., 2017).

In this task, LLMs identify how a property relates to a noun phrase. For example, in Table 2, the property "good for health" in "a green apple" is an *component property*, as the apple is already good for health itself.

3.2 Data Collection

We propose the dataset, **C**onceptual **C**ombination with **P**roperty **T**ype (CCPT), to address the three tasks in § 3.1. We both use a automated methods and human filtering. Detailed sources of textual corpora are described in Appendix A.1.

Step 1. Extract and Filter Combinations.

First, we extract noun phrases from the corpus. To obtain noun phrases with a property hint in a sentence, we extract sentences that contain "like" or "as" which compare one concept to another. These comparisons explicitly highlight the characteristics (e.g., "our economy will be as unstable as an apple on a toothpick"). This yields 51.0M comparative sentences—8% of the paragraph in the original corpus contained such sentences.

To avoid common expressions, such as proper nouns or idioms, whose meanings can be memorized from training corpus, we exclude any N-grams found in ConceptNet. We utilize uni-gram concept set from ConceptNet and use it as basic concepts. In the end, we collected 136.0K comparative sentences containing combination made up of two uni-gram concepts.

Step 2. Extract and Filter Properties.

Emergent & Component Candidates. Property extraction by syntactic patterns such as "ADJ/ADV like C" or "as ADJ/ADV as C" often misses implicit properties (e.g., "the storm was almost like a raging bull") or multi-word properties (e.g., "they *crashed together* like a boat on the rocks").

To address this, we use GPT-4o-mini (OpenAI, 2024a) to extract 10 properties from given comparative sentence for each combination. Then VERA-T5-XXL (Liu et al., 2023) filters out unlikely properties (with an alignment score under 0.7), resulting in 41.6K noun phrases and 211.0K properties.

Then we extract candidates for each property type. For emergent properties, with GPT-4o-mini, we select instances where neither the head noun $\mathcal{H}$ nor the modifier $\mathcal{M}$ already possess the property $\mathcal{P}$ of the noun phrase $\mathcal{N}$. We then limit the selection to at most five instances per noun phrase, resulting in 3,851 candidates. For component properties, we randomly sample 10K candidates.

Canceled Candidates. Since we cannot gather canceled properties from the corpus, we use two additional sources to collect the properties of head nouns: one for the "HasProperty" relations from ConceptNet and another from GPT-4o-mini, following Step 2. We randomly sample 2K noun phrases and gather up to 10 properties for each head noun from each source. Then, VERA-T5-XXL filters out the most likely properties (with an alignment score above 0.7), resulting in 2K noun phrases and a total of 23K properties.

With GPT-4o-mini, we select instances where the noun phrase $\mathcal{N}$ no longer possesses the property $\mathcal{P}$ of the head noun $\mathcal{H}$. Then, we select at most 5 data with the same noun phrase. We then limit the selection to at most five instances per noun phrase, resulting in 4,457 candidates.

Step 3. Annotate Property Types. The annotation process consists of three stages: first, assign a 5-point Likert scale of relevance score ranging from Highly Negative to Highly Positive; second, annotate the property type; third, find the toxicity.

In total, we have 12,315 data points, the type of property $\mathcal{T}$ consisting of 2501 emergent prop-

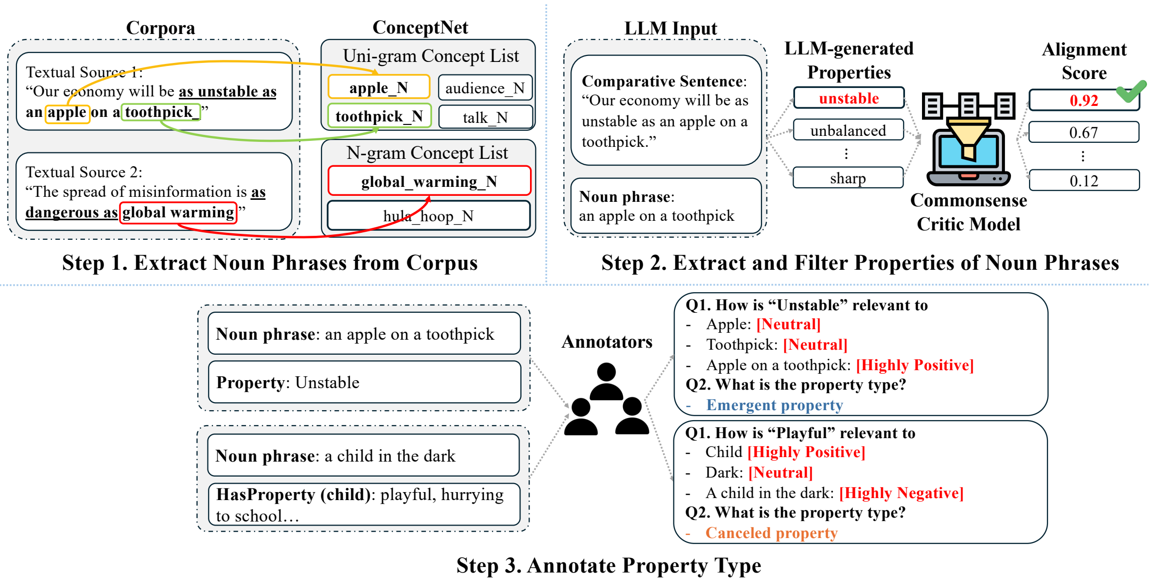


Figure 2: Overview of our data collection pipeline for conceptual combination through automated and human-driven data annotation.

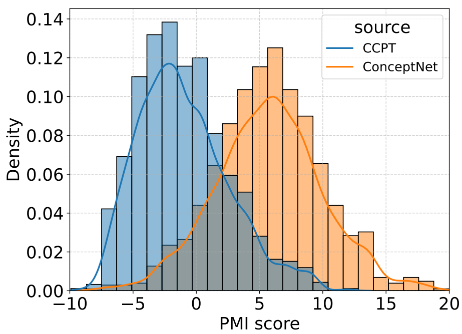


Figure 3: Distributions of Pointwise Mutual Information (PMI) on log-2 scale based on the Google Books N-gram Corpus.

erties, 1613 canceled properties, and 8201 component properties. Further details on the annotation procedure and inter-annotator agreement are provided in Appendix A.3.

3.3 Co-occurence of Concepts

To analyze how novel the noun phrases in our dataset is, we form a co-occurrence matrix based on PMI scores from the Google Books Ngram Corpus. The PMI formula is:

$$\mathrm{PMI}(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)},$$

where noun phrases with zero frequency are discarded.

When comparing our dataset to bi-gram concepts from ConceptNet, Figure 3 shows that combinations in our dataset have lower co-occurrence than those in ConceptNet. The average PMI for our dataset is -1.03, compared to 5.78 for ConceptNet, suggesting that combinations are more novel.

4 Benchmarking LLMs and Humans

We comprehensively assess 6 current LLMs with different architectures and sizes, including both open-source and closed-source models: LLaMa-3.1-70B-Instruct (Dubey et al., 2024), Qwen2.5-72B-Instruct (Team, 2024), GPT-4o-20240513 (OpenAI, 2024b), Claude-3.5-Sonnet (Anthropic, 2024), o1-mini-2024-09-12 (OpenAI, 2024d), and o1-2024-12-17 (OpenAI, 2024c). The test instances are a randomly sampled representative sample of data instances; N=200 for PROPERTY INDUCTION-EMERGENT, N=167 for PROPERTY INDUCTION-CANCELED, N=167 for NOUN PHRASE COMPLETION-EMERGENT.

4.1 Methods

To provide background knowledge on conceptual combinations, we use a consistent system prompt across all baselines and tasks. This prompt includes explanations of key concepts such as conceptual combinations, head nouns, modifiers, and different types of properties. Implementation details about model are explained in Appendix B.1. Prompts are detailed in Appendix B.2.

Base. Base prompting method is to evaluate LLMs' ability to generate their responses without reasoning steps.

Chain-of-Thought (CoT). We adopt chain-of-thought (Kojima et al., 2022) method to induce LLMs to generate the reasoning steps before pro-

Algorithm 1: SPREADING ACTIVATION

```
Input: Model 𝓜, Initial set of seed concepts C_0,
       Prompts {p_act, p_fil}, Concept graph G
Parameter: Maximum iterations T, Convergence
           threshold ϵ, Use language model U_L,
           Use concept graph U_C
Result: Related concept set C_T

# Retrieve relevant concepts A_c
Function(Activate (c)): begin
    A_c ← ∅
    if U_L then
        A_c ← A_c ∪ 𝓜(p_act, c)
    end
    if U_C then
        A_c ← A_c ∪ G.query(c)
    end
    return A_c
end

# Filter each concept in C_t w.r.t. C_0
Function(Filter (C_t, C_0)): begin
    return 𝓜(p_fil, C_t, C_0)
end

for iteration t = 0 to T do
    C_{t+1} ← C_t
    for each concept c ∈ C_t ∪ C_0 do
        A_c ← Activate(c)
        C_{t+1} ← C_{t+1} ∪ A_c
    end
    C_{t+1} ← Filter(C_{t+1}, C_0) \ C_0
    Δ = 1 − |C_t ∩ C_{t+1}| / |C_t ∪ C_{t+1}|
    if Δ < ϵ then
        break
    end
end
C_T ← C_{t+1}
return C_T
```

ducing the final answer by adding the phrase "Let's think step by step" at the end of the questions.

Spreading Activation (S.A.) (Ours) Spreading activation is a cognitive model to search through networks of related ideas or concepts (Anderson, 1983; Collins and Loftus, 1975). One concept in the mind activates another concept through the pathway of association. It becomes easier for people to combine these related concepts together and understand them as a relationship.

Motivated by spreading activation theory, we propose a novel method to iteratively explore relationships between concepts. In Algorithm 1, the initial concept set (C_0) and objective (O) are first defined. For example, generating the emergent property of a peeled apple is represented as $C_0 = \{\text{peeled, apple}\}$, with O defined as "find relationships between 'peeled' and 'apple'." Relevant concepts (A_c) are then activated for each component in the concept set (C_t) using either LLM or a graph-based approach such as ConceptNet. A filtering step selects components from C_t based on their relatedness to C_0. If there is no significant difference between C_t and C_{t+1}, the loop terminates. After iteratively expanding the set, the LLM generates the final answer based on the intermediate concept set C_T. Specifically, we set the maximum iteration steps (T) to 5 and the convergence threshold (ϵ) to 0.1.

Multi-Oracle. The best result among the multiple efforts can be seen as the upper-bound performance for each LLM. For the research purpose, we include this score by selecting the best score among the multiple solutions' (N=5) scores, presented with a gray background in Table 3.

Gold. We provide the score assigned to our annotated dataset as an upper-bound performance score for our tasks.

4.2 Human Responses

We recruited 5 native English speaker students through offline advertisement on a university campus. None of the students knew the researchers or had heard about conceptual combinations before. We verbally introduced the meaning of conceptual combination, and by e-mail, test sheets for each task were sent. The students solved the tasks in their own time and place. 1 to 3 students solved a single test sheet. We report the best result. Question format is provided in Appendix B.3.

4.3 Evaluation Metric

Generative Tasks. It can be challenging to determine whether emergence or cancellation occurs. For instance, an overripe apple may be more strongly associated with red than a regular apple, but this doesn't clearly indicate emergence. Similarly, determining cancellation presents the same difficulty. To address this, we propose two metrics based on a continuous relevance scoring. Both human judges and the LLM-as-a-judge approach (Zheng et al., 2023) are utilized to evaluate open-ended generative responses. These metrics are applied to the PROPERTY INDUCTION and NOUN PHRASE COMPLETION.

- The **emergence score** ($\mathcal{E}$) measures how suddenly properties arise when concepts are combined, compared to the properties of the individual concepts.
- The **cancellation score** ($\mathcal{C}$) reflects how much a property is diminished or canceled when

	Property Induction						Noun Phrase Completion		
	(1) Emergent			(2) Canceled			(3) Emergent		
	$R_{\mathcal{H},\mathcal{M},\mathcal{P}}\downarrow$	$R_{\mathcal{N},\mathcal{P}}\uparrow$	$\mathcal{E}\uparrow$	$R_{\mathcal{H},\mathcal{M},\mathcal{P}}\uparrow$	$R_{\mathcal{N},\mathcal{P}}\downarrow$	$\mathcal{C}\uparrow$	$R_{\mathcal{H},\mathcal{M},\mathcal{P}}\downarrow$	$R_{\mathcal{N},\mathcal{P}}\uparrow$	$\mathcal{E}\uparrow$
LLaMa3.1-70B									
Base	42.2 ± 1.8	81.4 ± 0.9	41.6 ± 0.6	64.1 ± 1.0	15.8 ± 1.2	50.3 ± 1.1	57.0 ± 1.3	**74.7 ± 0.7**	21.1 ± 1.9
CoT	41.8 ± 1.5	81.4 ± 0.7	42.5 ± 1.3	61.5 ± 0.5	14.0 ± 0.8	49.5 ± 0.7	52.1 ± 1.4	73.8 ± 0.5	24.8 ± 1.7
S.A.w/ LLM	**40.6 ± 0.3**	81.6 ± 0.5	43.3 ± 0.6	66.2 ± 1.4	13.9 ± 1.0	55.2 ± 1.4	**41.2 ± 1.5**	64.2 ± 1.0	**26.6 ± 2.1**
S.A.w/ ConceptNet	42.1 ± 0.5	**82.0 ± 0.5**	42.6 ± 0.7	**68.2 ± 0.9**	11.4 ± 0.7	57.9 ± 0.8	42.9 ± 0.7	66.0 ± 1.9	26.3 ± 1.3
S.A.w/ Both	**40.6 ± 0.6**	81.7 ± 0.7	**43.6 ± 0.3**	68.1 ± 1.3	**11.3 ± 1.4**	**58.3 ± 2.0**	43.9 ± 0.6	64.0 ± 1.8	24.3 ± 0.8
Multi-Oracle	30.7 ± 0.7	89.4 ± 0.4	54.6 ± 0.5	76.3 ± 0.5	6.8 ± 0.3	64.1 ± 0.6	44.8 ± 0.6	84.1 ± 0.3	32.4 ± 0.5
Qwen2.5-72B									
Base	46.6 ± 0.7	82.4 ± 0.4	40.2 ± 0.8	62.7 ± 0.7	11.8 ± 0.9	52.7 ± 1.2	54.6 ± 0.6	**75.5 ± 1.1**	24.5 ± 0.8
CoT	**44.4 ± 1.0**	79.3 ± 0.5	40.2 ± 0.7	62.0 ± 0.8	**9.9 ± 0.6**	54.1 ± 0.9	52.0 ± 0.9	75.0 ± 0.9	**26.5 ± 0.6**
S.A.w/ LLM	45.1 ± 0.4	84.3 ± 0.4	**42.9 ± 0.7**	**76.9 ± 1.4**	13.0 ± 0.7	**64.7 ± 1.2**	49.0 ± 0.8	68.6 ± 1.2	23.3 ± 0.3
S.A.w/ ConceptNet	44.8 ± 0.3	84.5 ± 0.4	**42.9 ± 0.2**	75.1 ± 1.8	13.5 ± 0.9	62.4 ± 1.6	**47.3 ± 1.6**	66.1 ± 1.8	22.4 ± 0.1
S.A.w/ Both	45.8 ± 0.8	**84.6 ± 0.5**	42.4 ± 1.1	67.6 ± 1.0	13.0 ± 0.7	56.3 ± 0.2	48.1 ± 0.3	69.0 ± 1.3	24.2 ± 1.2
Multi-Oracle	40.4 ± 0.8	87.4 ± 0.3	47.5 ± 0.9	71.5 ± 0.4	6.4 ± 0.2	63.4 ± 0.5	48.3 ± 0.2	82.4 ± 0.4	31.7 ± 0.7
GPT-4o									
Base	44.1 ± 0.6	83.3 ± 0.4	40.8 ± 0.7	67.5 ± 1.0	13.0 ± 0.7	55.5 ± 1.1	53.1 ± 2.0	69.8 ± 1.6	20.4 ± 1.5
CoT	43.0 ± 0.5	83.1 ± 0.7	42.4 ± 0.8	68.6 ± 0.6	**11.8 ± 1.4**	57.6 ± 0.7	52.0 ± 0.4	**72.1 ± 0.2**	23.1 ± 0.4
S.A.w/ LLM	42.9 ± 1.6	84.6 ± 0.7	43.8 ± 2.0	75.5 ± 1.2	15.1 ± 1.8	60.8 ± 1.5	46.8 ± 1.0	69.9 ± 0.3	25.5 ± 0.8
S.A.w/ ConceptNet	**41.5 ± 1.4**	85.8 ± 0.4	**45.8 ± 1.7**	**75.7 ± 1.2**	12.8 ± 0.6	**63.3 ± 1.6**	**46.3 ± 0.6**	70.8 ± 1.5	**26.9 ± 0.5**
S.A.w/ Both	43.0 ± 0.8	**86.6 ± 0.2**	45.2 ± 0.9	74.0 ± 0.4	13.0 ± 1.3	62.0 ± 1.3	47.0 ± 1.3	70.8 ± 1.5	26.8 ± 2.0
Multi-Oracle	28.9 ± 0.4	92.0 ± 0.4	57.8 ± 0.2	82.3 ± 0.6	4.5 ± 0.3	72.4 ± 0.9	35.7 ± 0.9	85.5 ± 0.7	38.9 ± 0.5
Sonnet-3.5									
Base	37.9 ± 0.7	78.9 ± 0.5	43.6 ± 0.9	71.1 ± 1.3	11.8 ± 0.7	59.8 ± 1.0	46.6 ± 0.5	67.8 ± 1.2	24.2 ± 1.0
Multi-Oracle	33.0 ± 0.8	81.2 ± 0.5	39.4 ± 0.6	84.6 ± 0.7	4.4 ± 0.1	75.2 ± 0.3	33.0 ± 0.8	81.2 ± 0.5	39.4 ± 0.6
o1-mini									
Base	42.1 ± 0.2	81.9 ± 0.5	42.1 ± 0.1	78.2 ± 0.1	17.8 ± 2.6	61.3 ± 2.1	47.7 ± 0.8	68.6 ± 1.2	24.5 ± 1.3
Multi-Oracle	26.0 ± 0.4	91.8 ± 0.2	60.3 ± 0.2	90.7 ± 0.9	5.3 ± 0.9	76.8 ± 0.9	31.3 ± 0.8	84.0 ± 0.2	44.0 ± 0.3
o1									
Base	37.3 ± 1.2	79.9 ± 0.9	43.5 ± 0.4	76.2 ± 1.9	7.9 ± 0.5	68.4 ± 0.9	43.8 ± 0.7	74.0 ± 1.7	32.6 ± 1.6
Multi-Oracle	21.9 ± 0.7	89.0 ± 0.2	60.9 ± 0.9	84.8 ± 0.9	2.3 ± 0.9	78.9 ± 0.9	30.7 ± 1.0	85.7 ± 0.7	49.7 ± 1.5
Human	37.7	85.0	49.2	69.5	11.1	58.6	36.4	72.5	38.0
Gold	29.2	87.4	58.4	83.2	14.2	69.5	27.5	87.2	59.9

Table 3: Generative results on test instances, reporting average scores with their standard error of the mean (SEM). In the Emergent Property scenario, better emergence corresponds to lower $R_{\mathcal{H},\mathcal{M},\mathcal{P}}\downarrow$ and higher $R_{\mathcal{N},\mathcal{P}}\uparrow$. In the Canceled Property scenario, better cancellation corresponds to higher $R_{\mathcal{H},\mathcal{M},\mathcal{P}}\uparrow$ and lower $R_{\mathcal{N},\mathcal{P}}\downarrow$. **Bold score** indicates the best score; underlined score is the second-best. Multi-Oracle in gray background represents upper-bound performance for each LLM and is not included in the rankings.

	Property Induction						Noun Phrase Completion		
	(1) Emergent			(2) Canceled			(3) Emergent		
GPT-4o	$R_{\mathcal{H},\mathcal{M},\mathcal{P}}\downarrow$	$R_{\mathcal{N},\mathcal{P}}\uparrow$	$\mathcal{E}\uparrow$	$R_{\mathcal{H},\mathcal{M},\mathcal{P}}\uparrow$	$R_{\mathcal{N},\mathcal{P}}\downarrow$	$\mathcal{C}\uparrow$	$R_{\mathcal{H},\mathcal{M},\mathcal{P}}\downarrow$	$R_{\mathcal{N},\mathcal{P}}\uparrow$	$\mathcal{E}\uparrow$
Base	49.2	84.1	36.3	70.7	29.4	43.9	54.3	69.0	20.4
CoT	45.4	84.1	41.3	68.0	21.5	47.0	53.0	**77.8**	26.7
S.A.w/ LLM	**43.0**	84.4	**43.3**	**79.0**	23.1	**57.6**	52.6	69.6	22.0
S.A.w/ ConceptNet	44.8	**86.1**	42.2	75.9	**19.6**	56.5	**48.7**	65.4	20.2
S.A.w/ Both	47.4	84.6	40.2	77.8	22.8	56.1	50.9	75.9	**29.6**
Human	42.8	84.8	44.3	73.5	22.4	52.6	43.7	79.8	39.0
Gold	34.4	87.2	52.8	79.6	21.1	59.0	33.0	90.6	57.6

Table 4: Human evaluation of generative results on sub-sampled test instances (N=30). The relevance scores are calculated by the average scores of two annotators. **Bold numbers** indicate the best scores, while underline numbers are the second-best.

concepts are combined, compared to their individual properties.

Specifically, each score is defined as

$$R_{\mathcal{H},\mathcal{M},\mathcal{P}} = \max\{R_{\mathcal{H},\mathcal{P}}, R_{\mathcal{M},\mathcal{P}}\}, \quad (1)$$

$$\mathcal{E} = \max\{R_{\mathcal{N},\mathcal{P}} - R_{\mathcal{H},\mathcal{M},\mathcal{P}}, 0\}, \quad (2)$$

$$\mathcal{C} = \max\{R_{\mathcal{H},\mathcal{M},\mathcal{P}} - R_{\mathcal{N},\mathcal{P}}, 0\}, \quad (3)$$

Here, $R_{\mathcal{X},\mathcal{P}}$ represents the relevance score, indicating how strongly concept $\mathcal{X}$ possesses property $\mathcal{P}$, measured by human judges and GPT-4o (OpenAI, 2024b) on a scale from 0 to 1. Exceptionally, for the Noun Phrase Completion, cancellation score is not measured since it easily occurs by adding antonym of given canceled property (e.g., to cancel "yellowness" of "banana", add "brown" to "banana"). Detailed instructions for both manual evaluation and LLM-as-a-judge are provided in Table 17 and Figure 8.

Classification Task. In this classification setup, we evaluate performance based on accuracy (%) in the PROPERTY TYPE PREDICTION. The classification involves four categories: "canceled property", "emergent property", "component property", and "others property". The "others property" category is specifically for properties unrelated to the combination and its components.

5 Results

5.1 Generative Task Result

Table 3 presents the experimental results in the generative setting for the two tasks, PROPERTY INDUCTION and NOUN PHRASE COMPLETION, evaluated based on LLM-as-a-judge. Table 4 shows the corresponding results based on human-judge evaluations for the same tasks.

Which property do the LLMs generate better: emergent or canceled? As shown in Table 3-(1) and (2), all baseline models find it more challenging to generate *emergent properties* than *canceled properties*. Humans outperform LLMs in generating emergent properties from noun phrases but perform worse in handling canceled properties. For emergent properties, GPT-4o-S.A. w/ ConceptNet achieves a significantly higher relatedness score between the property and each component, $\mathcal{R}_{\mathcal{H},\mathcal{M},\mathcal{P}}$, exceeding the gold score by 12.3 points. In contrast, the relatedness score between the property and the noun phrase, $\mathcal{R}_{\mathcal{N},\mathcal{P}}$, is only 1.6 points lower. While LLMs generate properties that align well with the noun phrase, they tend to rely on properties already associated with the individual components.

How well do LLMs create conceptual combinations? In Table 3-(3), it is difficult for all baselines to come up with a modifier $\mathcal{M}$ that lacks a given property on its own but exhibits that property when combined with a head noun. The emergence scores $\mathcal{E}$ for this task are 27 or lower across all models. Humans are better than LLMs at generating combinations that exhibit a given emergent property.

Which generative task do LLMs excel at? We compared two tasks: generating emergent properties (Table 3-(1)) and generating a noun phrase (Table 3-(1)). All baseline models achieve lower emergence scores ($\mathcal{E}$) in the noun phrase completion task. This suggests that LLMs find it more challenging to create a noun phrase that accurately captures an emergent property than to identify an emergent property from a noun phrase.

		Predicted Types			
		Emergent	Component	Canceled	Others
Actual Types	Emergent	**90.0**	4.4	2.0	3.6
	Component	59.6	**37.2**	1.2	2.0
	Canceled	13.6	15.6	**45.2**	25.6
	Others	26.0	5.6	15.2	**53.2**

Table 5: Classification results of GPT-4o for property type prediction based on 1000 sampled instances (250 samples per property type). We present the average accuracy (%). Cells with a green background indicate cases where the model correctly predicts that combinations possess a property (95.6% acc), while cells with a red background indicate cases where it correctly predicts that combinations do not possess a property (69.6% acc).

5.2 Classification Task Result

Table 5 presents the experimental results for the PROPERTY TYPE PREDICTION. Given a noun phrase and a property, GPT-4o classifies the property into one of four categories: emergent, component, canceled, or others (where others is unrelated to both the combination and its components).

Do LLMs identify the type of property well? In determining whether a noun phrase has a given property, GPT-4o achieves an accuracy of 82.6% $((95.6\% + 69.6\%) \div 2)$. However, in predicting the type of property, GPT-4o is correct only 56.4% of the time, falling behind human accuracy (81%), as noted in Appendix B.4. Compared to its accuracy in identifying emergent properties, its performance across other property types lags significantly. These results suggest room for improvement in understanding different property types.

6 Analysis

6.1 Relevance between LLM-as-a-judge metric and Manual Evaluation

As shown in Figure 4, we compare the LLM judge's metrics with human evaluations to verify the agreement between them. We randomly selected 300 pairs from CCPT, which consists of pairs from $\mathcal{H}-\mathcal{P}$, $\mathcal{M}-\mathcal{P}$, and $\mathcal{N}-\mathcal{P}$, covering both emergent properties (50 samples) and canceled properties (50 samples). Human raters, recruited through Amazon Mechanical Turk, as detailed in Section B.5, are asked to rate the relevance of each pair using the same instructions provided to the LLM judge. Each problem is rated by three different raters. We

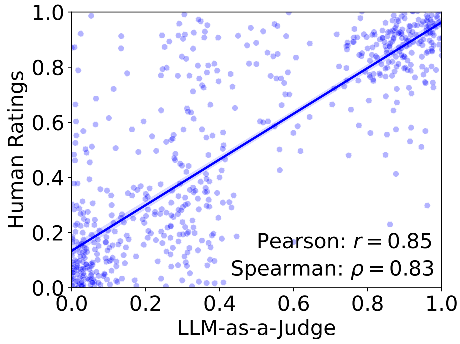


Figure 4: Correlation between LLM-as-a-judge and human ratings in relevance score, assessing how strong a concept $\mathcal{X}$ possesses a property $\mathcal{P}$. To avoid overlapping points, random jitters sampled from $\mathcal{N}(0, 0.05^2)$ are added to LLM-as-a-judge and human ratings after fitting the regression.

calculated the Pearson and Spearman correlation coefficients between the LLM judge's scores and human ratings, which are 0.85 and 0.83, respectively. These strong correlations indicate a high level of agreement, demonstrating the effectiveness of using LLM for relevance scoring.

6.2 Analysis for Spread Activation Method

First, to evaluate the impact of multiple iterations on performance, we adjust T from 1 to 5 and gather the answers for each C_T. Table 7 highlights that as the number of iterations increases, the performance of each task gradually improves. This indicates that iteratively propagating relevant concepts is effective in our conceptual combination tasks.

Second, our spread activation method consists of the functions `Activate`$(\cdot)$ and `Filter`$(\cdot, \cdot)$. To investigate the contribution of `Filter` on performance, we conducted an ablation study by comparing scores with and without `Filter` in the spread activation method. As shown in Table 6, performance decreases when `Filter` is removed. This suggests that iteratively eliminating distractors improves overall performance.

7 Conclusion

In conclusion, our work introduces CCPT, a conceptual combination dataset designed to evaluate LLMs' ability to process conceptual combinations. CCPT comprises 12,315 annotated instances of noun phrases, properties, and property types. Based on CCPT, we propose three downstream tasks: property induction, noun phrase completion, and property type prediction. To assess generative performance, we introduce two automatic evaluation metrics—emergence and cancellation scoring—which closely align with human evaluations. Additionally, we propose a novel evaluation method inspired by cognitive psychology models. Our findings indicate that (1) LLMs struggle more with generating emergent properties than with canceled ones. Furthermore, (2) generating a noun phrase that exhibits a given emergent property proves more challenging than generating the property itself. Notably, (3) GPT-4o struggles to determine property types compared to humans. Finally, (4) our proposed spread activation method achieves the highest performance among the evaluated approaches due to its iterative retrieval of relevant concepts and filtering process.

Task	C_1	C_2	C_3	C_4	C_5
PI-Eme.	45.1	45.4	45.2	**45.9**	45.8
PI-Can.	60.5	61.1	61.1	62.7	**63.3**
NPC-Eme.	26.0	26.2	**26.9**	26.7	**26.9**

Table 6: Iteration-wise score improvement of GPT-4o + S.A.w/ConceptNet. PI and NPC denote the PROPERTY INDUCTION and NOUN PHRASE COMPLETION, respectively. C_t represents the utilization of C_t to generate the final answer. Each score in the table corresponds to either $\mathcal{E}$ or $\mathcal{C}$, depending on the property type.

Task	w/o `Filter`	w/ `Filter`	Δ
PI-Eme.	44.6 ± 0.9	$\mathbf{45.8 \pm 1.7}$	$\uparrow 1.2$
PI-Can.	55.6 ± 1.0	$\mathbf{63.3 \pm 1.6}$	$\uparrow 7.7$
NPC-Eme.	26.0 ± 1.6	$\mathbf{26.9 \pm 0.5}$	$\uparrow 0.9$

Table 7: Impact of filtering in the spread activation method. The columns compare performance with and without filtering, highlighting the absolute score change (Δ).

Limitations

We acknowledge few potential limitations of our research. (1) There is fundamental diversity in people's mental representations of the world, especially across cultural contexts. The notions of "property" and "property type" in our dataset may implicitly reflect the commonsense knowledge of the annotators' demographic group. Moreover, the approval of our data through the MTurk study may primarily reflect the commonsense of the Turkers. Future work could further explore the relationship between conceptual combination understanding and cultural divergence in concepts. (2) Homonyms can introduce misleading effects on the evaluation process. If the grader misinterprets the definition of concept from the solver's intention, the solver's performance may not be fully captured. (3) Our data generation pipeline employs comparative sentences for efficiency. However, the inherent nature of comparative sentences may introduce skewness, favoring certain types of properties over others.

Ethics Statement

The authors checked all examples and found no personal identifying information (PII). As addressed in § 3.2, we also eliminated the offensive contents manually.

Acknowledgements

We thank the anonymous reviewers and Yunah Jang for their valuable comments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II191082, SW StarLab), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2023R1A2C2005573), the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(RS-2023-00274280), Korea Radio Promotion Association (Development of Intelligent Docent Service for Information-Disadvantaged Groups), and the SNU-Global Excellence Research Center establishment project. Gunhee Kim is the corresponding author.

References

John R Anderson. 1983. A spreading activation theory of memory. *Journal of verbal learning and verbal behavior*, 22(3):261–295.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

BIG-bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Albert Coil and Vered Shwartz. 2023. From chocolate bunny to chocolate crocodile: Do language models understand noun compounds? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2698–2710.

Allan M Collins and Elizabeth F Loftus. 1975. A spreading-activation theory of semantic processing. *Psychological review*, 82(6):407.

Marc N Coutanche, Sarah Solomon, and Sharon L Thompson-Schill. 2019. Conceptual combination in the cognitive neurosciences.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Zachary Estes and Sam Glucksberg. 2020. Relevance and feature accessibility in combined concepts. In *Proceedings of the Twenty-first Annual Conference of the Cognitive Science Society*, pages 149–154. Psychology Press.

Zachary Estes and Thomas B Ward. 2002. The emergence of novel attributes in concept modification. *Creativity Research Journal*, 14(2):149–156.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Christina L Gagné, Thomas L Spalding, and Matthew Kostelecky. 2017. Conceptual combination, property inclusion, and the aristotelian-thomistic view of concepts. *Compositionality and concepts in linguistics and psychology*, pages 223–244.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

James A Hampton. 1997. Emergent attributes in combined concepts.

Iris Hendrickx, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Stan Szpakowicz, and Tony Veale. 2013. SemEval-2013 task 4: Free paraphrases of noun compounds. In *Second Joint Conference on Lexical and Computational Semantics (*SEM),* *Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 138–143, Atlanta, Georgia, USA. Association for Computational Linguistics.

Nicholas W Kohn, Paul B Paulus, and Runa M Korde. 2011. Conceptual combinations and subsequent creativity. *Creativity Research Journal*, 23(3):203–210.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Ruixi Lin and Hwee Tou Ng. 2022. Does bert know that the is-a relation is transitive? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 94–99.

Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah A Smith, Yejin Choi, and Hannaneh Hajishirzi. 2023. Vera: A general-purpose plausibility estimation model for commonsense statements. *arXiv preprint arXiv:2305.03695*.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2024-07-18.

OpenAI. 2024b. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-05-13.

OpenAI. 2024c. Introducing openai o1. https://openai.com/o1/. Accessed: 2025-02-08.

OpenAI. 2024d. Openai o1-mini. https://openai.com/index/openai-o1-mini-advancing-cost-efficient-reasoning/. Accessed: 2024-10-12.

Hao Peng, Xiaozhi Wang, Shengding Hu, Hailong Jin, Lei Hou, Juanzi Li, Zhiyuan Liu, and Qun Liu. 2022. Copen: Probing conceptual knowledge in pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5015–5035.

Giulia Rambelli, Emmanuele Chersoni, Claudia Collacciani, and Marianna Bolognesi. 2024. Can large language models interpret noun-noun compounds? a linguistically-motivated study on lexicalized and novel compounds. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11823–11835.

Hayley Ross, Kathryn Davidson, and Najoung Kim. 2024. Is artificial intelligence still intelligence? LLMs generalize to novel adjective-noun pairs, but don't mimic the full human distribution. In *Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP*, pages 131–153, Miami, Florida, USA. Association for Computational Linguistics.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Vered Shwartz and Chris Waterson. 2018. Olive oil is made of olives, baby oil is made for babies: Interpreting noun compounds using paraphrases in a neural model. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 218–224.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Ken Springer and Gregory L Murphy. 1992. Feature availability in conceptual combination. *Psychological Science*, 3(2):111–117.

David Swinney, Tracy Love, Matthew Walenski, and Edward E Smith. 2007. Conceptual combination during sentence comprehension. *Psychological Science*, 18(5):397–400.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Paul Thagard. 1984. Conceptual combination and scientific discovery. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, volume 1984, pages 2–12. Cambridge University Press.

Thomas B Ward. 2001. Creative cognition, conceptual combination, and the creative writing of stephen r. donaldson. *American Psychologist*, 56(4):350.

Thomas B Ward. 2007. Creative cognition as a window on creativity. *Methods*, 42(1):28–37.

Merryl J Wilkenfeld and Thomas B Ward. 2001. Similarity and emergence in conceptual combination. *Journal of Memory and Language*, 45(1):21–38.

Edward J Wisniewski. 1997. When concepts combine. *Psychonomic bulletin & review*, 4:167–183.

Weiqi Wu, Chengyue Jiang, Yong Jiang, Pengjun Xie, and Kewei Tu. 2023. Do plms know and understand ontological knowledge? *arXiv preprint arXiv:2309.05936*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 5927–5934, Online. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Data Details

A.1 Textual Corpora

We utilize English textual corpora from datasets and websites across different domains including books and news. Our source is comprised of Toronto Book Corpus (Zhu et al., 2015), WritingPrompts (Fan et al., 2018), Gutenberg Corpus (Gerlach and Font-Clos, 2020), MediaSum News Dialogue (Zhu et al., 2021), Wikipedia-240823[1], r/FUNNY[2], r/FantasyWriters[3], and r/OCPoetry[4].

A.2 Dataset Construction Statistics

In Table 8, we describe the size of the dataset during the data collection pipeline.

A.3 Annotation Procedure

We hire skilled raters in Amazon Mechanical Turk (MTurk) to annotate the property type. For emergent properties and canceled properties, we assign three annotators per instance, and used majority label for the final property type. For component properties, due to cost constraints, we assign one annotator per instance.

Annotators were selected based on their success in a qualification task (Human Intelligence Task or HIT), which assessed their ability to distinguish property types. This qualification task included 10 thoroughly verified examples, with a payment of \$1.00. We required annotators to be from English-speaking countries (AU, CA, NZ, US, GB), have completed more than 10,000 HITs, and maintain a HIT approval rate greater than 98%.

After qualification, annotators received detailed instructions on conceptual combinations with examples. They answered three questions: (1) Relevance Rating – Annotators rated the relevance of each concept $(\mathcal{N}, \mathcal{H}, \mathcal{M})$ to a given property $(\mathcal{P})$ to encourage careful consideration of the data. (2). Property Type Annotation – For emergent and component properties, annotators chose: (1) emergent, (2) component, or (3) other. For canceled properties, they chose: (1) canceled, (2) component, or (3) other, as illustrated in Figures 5 and 6. For component property candidates there is a third question about locating the source of the property from the following options: (1) noun phrase $\mathcal{N}$, (2) head noun $\mathcal{H}$ and noun phrase $\mathcal{N}$, (3) modifier $\mathcal{H}$ and

[1]https://dumps.wikimedia.org/enwiki/latest/

[2]https://www.reddit.com/r/funny/

[3]https://www.reddit.com/r/fantasywriters/

[4]https://www.reddit.com/r/OCPoetry/

	# of Data
Corpora	~51,000,000
Extracting Comparative Sentences	136,300
Extracting and Filtering	41,660
Property Type Tagging (w/ Toxic Flitering)	12,315

Table 8: The size of the dataset during the data collection pipeline is represented by the number of remaining sentences or noun phrases at each step. The term "# of data" denotes the amount of data left after each processing stage.

noun phrase $\mathcal{N}$, (4) All of $\mathcal{N}, \mathcal{H}, \mathcal{M}$, (5) others. (3) Toxicity: The annotators are asked to check a box if the data is toxic. Finally, the Fleiss' κ score of three-way classification during the annotation phase is 0.312 for emergent property data and 0.410 for canceled property data. In CCPT, we contain disaggregated human annotations for all questions.

Next, we calculate inter-annotator result for the final dataset. For emergent and component properties, we randomly selected 100 samples for this task, each reviewed by two annotators who had not participated in the original annotation. The Fleiss' κ score for this binary classification task was 0.498, indicating agreement levels ranging from "moderate" to "substantial." For canceled properties, we also selected 100 samples, with two annotators reviewing each example. The Fleiss' κ score for this binary classification is 0.505, indicating agreement between "moderate" and "substantial" levels.

B Experimental Details

B.1 Implementation Details

We collect responses using Nucleus sampling with $\mathcal{T} = 0.7$ and $p = 0.95$, by selecting the most likely sequence. Model responses are generated using three different seeds, and we report the average scores along with the standard error of the mean (SEM).

B.2 Details in prompting

In this section, we explain our task instruction templates. For each method, we give a correct and wrong answers for one example.

For the system prompt, refer to Table 9. For the prompt given to Base method, refer to Table 10 and 13. For the CoT-prompting, refer to Table 11 and 14. Spread-Activation method uses a prompt in Table 12 and 15. For the propery type prediction task, refer to Table 16.

Our prompt for LLM-as-a-judge is in Table 17. The prompt provides the fine-grained scoring criteria from 1 to 10 and three scored examples.

B.3 Collecting human solutions

The students reported to need 25 to 45 minutes per 30 questions. We paid \$78 per 167-question test sheet and \$94 for 200-question test sheet; This exceeds the minimum wage in the country the authors and students are located in. The question format is as below.

(Property induction - emergent property)
Conceptual combination: **'{noun phrase}'**, Type-of-property: **'emergent property'**
Q. What is a property of **'{noun phrase}'** that **'{head noun}'** or **'{modifier}'** doesn't have?
A: ____________________

(Property induction - canceled property)
Conceptual combination: **'{noun phrase}'**, Type-of-property: **'canceled property'**
Q. What is a canceled property which is a property that belongs to **'{head noun}'** or **'{modifier}'** but does **not** belong to a noun phrase **'{noun phrase}'**?
A: ____________________

(Noun phrase completion)
Head noun: **'{head noun}'**, Property: **'{property}'**
Q. What is a noun phrase using **'{root}'** to be **'{property}'**?
A: ____________________

B.4 Recruiting Participants for Property Type Classification

To check human ability for property type prediction in Section 5.2, we recruited capable test participants who did **not** participate by Amazon Mechanical Turk (MTurk).

We provided the participants with the conceptual combination and property and asked them to classify the property type. The definition of the conceptual combination and the property types are described in the instructions. Three annotated data instances (one per each property type) are also provided. Instruction is shown in Figure 7. As a result, when we regard the majority vote as a human-annotated label, we get an accuracy of 87% for the emergent property, 79% for the component property, and 76% for the canceled property, with a total accuracy of 81%.

We prepared the qualification Human Intelligence Task (HIT), comprised of 10 thoroughly verified examples with a payment of \$1.0. We re-

System Prompt

Conceptual combination is a task that combines two concepts, which can result in new properties. It involves a head noun, a modifier, and corresponding properties. Here's the definition of each component:

1. Head Noun: The original concept in the conceptual combination.
2. Modifier: The word that modify head noun to create a new conceptual combination.
3. Component Property: A property inherent to individual concepts (head noun or modifier).
4. Emergent Property: A new property that arises from the combination of the head noun and the modifier. This property does not exist in either concept individually (head noun or modifier) but emerge in conceptual combination.
5. Canceled Property: A property that is inherent to individual concept (head noun or modifier) and negated due to the combination.

Table 9: System prompt for background of conceptual combination.

Task Instruction for Base Method in Noun Phrase Completion

Instructions:
1. You are given a head noun and emergent property. Your task is to generate a conceptual combination by adding one modifier.
2. You can use function word without any constraint.
3. Modifier should not have the given emergent property on its own, but the combination exhibits the emergent property.
4. Use the previous examples to learn the task.
5. Answer in dictionary format: {{"combination": "{{generated_combination}}", "modifier": "{{generated_modifier}}"}}. Do not include other formatting.

<Example 1>
- Head noun: apple
- Emergent property: unappetizing
- Correct answer: {{"combination": "brown apple", "modifier": "brown"}}

Above answer is correct because each component "brown" and "apple" do not possess "unappetizing" but "brown apple" does.

<Example 2>
- Head noun: banknote
- Emergent property: useless
- Wrong answer: {{"combination": "burned banknote", "modifier": "burned"}}

Above answer is wrong because modifier "burned" directly elicit property "useless". Avoid modifier which has given property in itself.

Then let's begin:

Table 10: Task instruction for Base Method in NOUN PHRASE COMPLETION.

Task Instruction for Chain-of-Thought Method in Noun Phrase Completion

Instructions:
1. You are given a head noun and emergent property. Your task is to generate a conceptual combination by adding one modifier.
2. You can use function word without any constraint.
3. Modifier should not have the given emergent property on its own, but the combination exhibits the emergent property.
4. Come up with your reasoning process before giving your final answer.
5. Use the previous examples to learn the task.
6. Answer in dictionary format: {{"combination": "{{generated_combination}}", "modifier": "{{generated_modifier}}"}}. Do not include other formatting.

<Example 1>
- Head noun: apple
- Emergent property: unappetizing
- Correct answer: Let's think step-by-step. A typical apple is fresh and appetizing, but certain modifications can make it unappetizing. Factors like discoloration, decay, or unusual texture can contribute to this perception. A brown apple, for instance, appears spoiled or oxidized, making it less appealing to eat. So the answer is {{"combination": "brown apple", "modifier": "brown"}}

Above answer is correct because each component "brown" and "apple" do not possess "unappetizing" but "brown apple" does.

<Example 2>
- Head noun: banknote
- Emergent property: useless
- Wrong answer: Let's think step-by-step. A typical banknote has value and can be used for transactions, but certain modifications can make it useless. Burning a banknote destroys its structure, making it unrecognizable and invalid as currency. So the answer is {{"combination": "burned banknote", "modifier": "burned"}}

Above answer is wrong because modifier "burned" directly elicit property "useless". Avoid modifier which has given property in itself.

Then let's begin:

Table 11: Task instruction for CoT Method in NOUN PHRASE COMPLETION.

Task Instruction for Spread Activation Method in Noun Phrase Completion

Instructions:
1. You are given a head noun, emergent property and a set of relevant concepts to solve a task. Your task is to generate a conceptual combination by adding one modifier.
2. You can use function word without any constraint.
3. Modifier should not have the given emergent property on its own, but the combination exhibits the emergent property.
4. Come up with your reasoning process before giving your final answer.
5. Final answer should follow dictionary format: {{"combination": "{{generated_combination}}", "modifier": "{{generated_modifier}}"}}. Do not use other formatting.

<Example 1>
- Head noun: apple
- Emergent property: unappetizing
- Relevant concepts: ['bland', 'bitter', 'inedible', 'insipid', 'unappealing', 'unpalatable']
- Correct answer: The goal is to find a modifier that does not inherently have the emergent property "unappetizing," but do when combined with "apple". Related concepts such as bitter, inedible or unpalatable make apple unappetizing. To represent bitter apple, "yellow" can be used as a modifier. But yellow is somewhat related to bitter because of the color of lemons. To represent inedible or unpalatable apple, "plastic" or "brown" can be used as a modifier. However plastic is directly related to inedible. "Brown" as a modifier doesn't imply inedible on its own, but when paired with "apple," it suggest an inedible state. So the answer is {{"combination": "brown apple", "modifier": "brown"}}

Above answer is correct because each component "brown" and "apple" do not possess "unappetizing" but "brown apple" does.

<Example 2>
- Head noun: banknote
- Emergent property: useless
- Relevant concepts: ['counterfeit', 'worthless', 'ineffective', 'futile', 'meaningless', 'pointless']
- Wrong answer: The goal is to find a modifier that does not inherently have the emergent property "useless," but do when combined with "banknote". Related concepts such as counterfeit or worthless make banknote useless. To represent counterfeit banknote, "fake" can be used as a modifier. But fake is somewhat related to useless because of the meaning. To represent worthless banknote, "burned" can be used as a modifier. So the answer is {{"combination": "burned banknote", "modifier": "burned"}}

Above answer is wrong because modifier "burned" directly elicit property "useless". Avoid modifier which elicit given property in itself.

Then let's begin:

Table 12: Task instruction for Spread Activation Method in NOUN PHRASE COMPLETION.

Task Instruction for Base Method in Property Induction (Emergent)

Instructions:
1. You are given a combination of concepts. Your task is to generate emergent property of a combination.
2. Find a property that does not belong to any of the individual component in the combination but emerges when the words are combined.
3. Use the previous examples to learn how to do this.
4. Answer in dictionary format: {{"property": "{{generated_property}}"}}. Do not include other formatting.

<Example 1>
- Combination: Brown apple
- Correct answer: {{"property": "unappetizing"}}

Above answer is correct because property "unappetizing" does not belong to either "brown" and "apple", but belong to brown apple

<Example 2>
- Combination: burned banknote
- Wrong answer: {{"property": "useless"}}

Above answer is wrong because "burned" something can directly belong to property "useless". Ensure that the emergent property you generate does not directly describe any of the individual words but is a characteristic of the combination as a whole.

Then let's begin:

Table 13: Task instruction for Base Method in PROPERTY INDUCTION.

Task Instruction for Chain-of-Thought Method in Property Induction (Emergent)
Instructions: 1. You are given a combination of concepts. Your task is to generate emergent property of a combination. 2. Find a property that does not belong to any of the individual component in the combination but emerges when the words are combined. 3. Use the previous examples to learn how to do this. 4. Come up with your reasoning process before giving your final answer. 5. Answer in dictionary format: {{"property": "{{generated_property}}"}}. Do not include other formatting. <Example 1> - Combination: Brown apple - Correct answer: Let's think step-by-step. "Brown" signifies a color change due to aging, oxidation, or decay, indicating the apple is no longer fresh. "Apple" is a fruit that is typically appealing when fresh, but browning suggests overripeness or spoilage. Concepts like spectrum (color change) and growth (life cycle of the fruit) reinforce this transition. So the answer is {{"property": "unappetizing"}} Above answer is correct because property "unappetizing" does not belong to either "brown" and "apple", but belong to brown apple <Example 2> - Combination: burned banknote - Wrong answer: Let's think step-by-step. Individually, "burned" does not mean something is useless. A "banknote" by itself is valuable and serves as a medium of exchange. However, when combined, a "burned banknote" implies that the note is damaged beyond recognition, making it invalid for transactions and effectively useless as currency. So the answer is {{"property": "useless"}} Above answer is wrong because "burned" something can directly belong to property "useless". Ensure that the emergent property you generate does not directly describe any of the individual words but is a characteristic of the combination as a whole. Then let's begin:

Table 14: Task instruction for CoT Method in PROPERTY INDUCTION.

Task Instruction for Spread Activation Method in Noun Phrase Completion (Emergent)
Instructions: 1. You are given a combination of concepts and a set of relevant concepts to solve a task. Your task is to generate emergent property of a combination. 2. Find a property that does not belong to any of the individual component in the combination but emerges when the words are combined. 3. Come up with your reasoning process before giving your final answer. 4. Final answer should follow dictionary format: {{"property": "{{generated_property}}"}}. Do not include other formatting. <Example 1> - Combination: Brown apple - Relevant concepts: ['fruit', 'apple', 'core', 'cider'] - Correct answer: The goal is to find an emergent property of "brown apple" that does not exist in "brown" or "apple" individually. "Fruit" and "apple" describe general attributes, so they are excluded. "Core" and "Cider" are unrelated to "brown apple." "Withered" relates to a decayed state, which applies to "brown apple" but not to "brown" or "apple" alone. To interpret "withered" naturally, we select "unappetizing" as the emergent property. So the answer is {{"property": "unappetizing"}} Above answer is correct because property "unappetizing" does not belong to either "brown" and "apple", but belong to brown apple <Example 2> - Combination: burned banknote - Relevant concepts: ['paper', 'ash', 'money', 'value'] - Wrong answer: The goal is to find an emergent property of a "burned banknote" that does not exist in either "burned" or "banknote" individually. The attributes "paper" and "money" describe general properties of a banknote, so they are excluded.Likewise, "ash" describes a general property of something that is burned, so it is also excluded. While a banknote possesses "value," this characteristic disappears once the banknote is burned. Consequently, to convey the idea of "valueless" naturally, we choose "useless" as the emergent property. So the answer is is: {{"property": "useless"}} Above answer is wrong because "burned" something can directly belong to property "useless". Ensure that the emergent property you generate does not directly describe any of the individual words but is a characteristic of the combination as a whole. Then let's begin:

Table 15: Task instruction for Spread Activation Method in Property Induction.

Task Instruction for Property Type Prediction

Instructions:
1. You are given a combination and property. Your task is to predict a type of property.
2. Definition of each property type is as follows:
- Emergent: The property emerges from the combination of components.
- Component: The property is inherited by component of the combination.
- Canceled: The property is canceled out by the combination of components.
- Others: The property is not related to the combination nor components.
3. Use the previous examples to learn the task.
4. Answer in dictionary format: {{"property_type": "{{property_type}}"}}. Do not include other formatting.

<Example 1>
- Combination: peeled apple
- Property: round
- Correct answer: {{"property_type": "component"}}

Above answer is correct because property "round" is inherited by component "apple".

<Example 2>
- Combination: burned banknote
- Property: useless
- Wrong answer: {{"property_type": "emergent"}}

Above answer is wrong because modifier "burned" directly elicit property "useless".

Then let's begin:

Table 16: Task instruction for property type prediction.

Prompt for LLM-as-a-Judge

Concepts are characterized by properties. For example, the concept "a chicken in front of a fox" strongly exhibits the property "in danger." When given a concept and a property, your task is to evaluate how much the concept has the property on a scale from 1 to 10. You should follow the format: {{"relevance": your_relevance_score}}

Use the following scoring criteria to assign a relevance score:
- {{"relevance": 1}}: The concept does not have the property at all.
- {{"relevance": 2-3}}: The concept rarely has the property.
- {{"relevance": 4-6}}: The concept sometimes has the property.
- {{"relevance": 7-8}}: The concept usually has the property, but not always.
- {{"relevance": 9}}: The concept almost always has the property.
- {{"relevance": 10}}: The concept always has the property.

—
Examples:
—
Concept: Rusty
Property: Useless
Relevance: {{"relevance": 7}}
—
Concept: A chicken in the cage
Property: In danger
Relevance: {{"relevance": 2}}
—
Concept: A chicken in front of a fox
Property: In danger
Relevance: {{"relevance": 9}}
—
Concept: Wrench
Property: Destructive
Relevance:

Table 17: Prompt for LLM-as-a-judge.

Instance:
- Head noun: "${root}"
- Modifier: "${modifier}"
- Combination: "${combination}"
- Property: "${property}"

Q1-1. How relevant is "${root}" to "${property}"?
- 1. Not Relevant
- 2. Slightly Relevant
- 3. Moderately Relevant
- 4. Highly Relevant
- 5. Fully Relevant

Q1-2. How relevant is "${modifier}" to "${property}"?
- 1. Not Relevant
- 2. Slightly Relevant
- 3. Moderately Relevant
- 4. Highly Relevant
- 5. Fully Relevant

Q1-3. How relevant is "${combination}" to "${property}"?
- 1. Not Relevant
- 2. Slightly Relevant
- 3. Moderately Relevant
- 4. Highly Relevant
- 5. Fully Relevant

Q2. Does the property emerge? (Your answer should be consistent with the 1-5 scaling above)
- **Emergent property**. The property "${property}" newly emerges by the combination.
- **Component property**. The property "${property}" is already in "${root}" or "${modifier}".
- **Others**. It does not apply to the first or second option. (e.g. The property "${property}" is not related with "${combination}", the combination "${combination}" itself is nonsense, etc...)

☐ **NOTE:** Please check this box if aligning the property "${property}" with "${combination}" could be considered *toxic* (e.g., content that promotes racism, hate speech, or NSFW material).

Figure 5: Instructions provided for annotators of emergent property data candidates.

cruited participants from AU, CA, NZ, US, and GB, with more than 10000 HITs approved, and a HIT approval rate greater than 98%. Among 40, this process resulted in 12 participants.

After qualification, we asked raters with a payment of $0.2 per HIT. Each example was evaluated by three annotators and the inter-annotator agreement was 0.59 in Fleiss' Kappa (Fleiss, 1971).

B.5 Evaluation of relation between LLM-as-a-Judge and Human Judge

To ensure the quality of evaluation metric in CCPT, we measure a correlation between LLM-as-a-judge and human ratings in Section 6.1. Like the previous subsection, we hire capable raters in Amazon Mechanical Turk (MTurk) who did **not** participate in the data annotation process before this test.

The basic qualifications are also made by nationality (AU, CA, NZ, US, and GB), the number of HITs approved (10000), and the HIT approval rate (greater than 98%). We performed a qualification test with a payment of $0.1 and chose 12 raters among the 85 applicants who had completed more than 5 qualification HITs. We paid $0.1 for the main rating. Refer to Figure 8 for the instruction that we used for the qualification and the main judge task.

Instance:
- Head noun: **"${root}"**
- Modifier: **"${modifier}"**
- Combination: **"${combination}"**
- Property: **"${property}"**

Q1-1. How relevant is **"${root}"** to **"${property}"**?
○ 1. Not Relevant
○ 2. Slightly Relevant
○ 3. Moderately Relevant
○ 4. Highly Relevant
○ 5. Fully Relevant
Q1-2. How relevant is **"${modifier}"** to **"${property}"**?
○ 1. Not Relevant
○ 2. Slightly Relevant
○ 3. Moderately Relevant
○ 4. Highly Relevant
○ 5. Fully Relevant
Q1-3. How relevant is **"${combination}"** to **"${property}"**?
○ 1. Not Relevant
○ 2. Slightly Relevant
○ 3. Moderately Relevant
○ 4. Highly Relevant
○ 5. Fully Relevant
Q2. Is the property negated? (Your answer should be consistent with the 1-5 scaling above!!)

○ **Canceled property**. The property **"${property}"** ***disappears*** (is negated) in **"${combination}"**, but the property **"${property}"** ***was in*** **"${root}"** or **"${modifier}"**. (High relevance score with root & but low relevance with combination // High relevance score with modifier & but low relevance with combination

○ **Component property**. The property **"${property}"** is ***already in*** **"${root}"** or **"${modifier}"**, and **"${combination}"** **also has** the property **"${property}"**.

○ **Others**. It does not apply to the first or second option. (e.g. The property **"${property}"** is **not related with** any of **"${combination}"**, **"${root}"** or **"${modifier}"** / the combination **"${combination}"** itself is **nonsense**, etc...)

☐ **NOTE:** Please check this box if aligning the property "${property}" with "${combination}" could be considered *toxic* (e.g., content that promotes racism, hate speech, or NSFW material).

Figure 6: Instructions provided for annotators of canceled property data candidates.

What is a Conceptual Combination?

A conceptual combination consists of:
- **Head Noun**: The central word (e.g., "apple").
- **Modifier**: A word that adds more detail to the head noun (e.g., "peeled").

When combined, the properties of the phrase might change or remain the same. For detailed explanation of conceptual combination, refer to [wikipedia site].

Example:
Conceptual Combination: **"peeled apple"**

1. **Property: White**
A peeled apple is usually white, even if "peeled" and "apple" aren't directly related to being white.
The property "white" **emerges**, thus "white" is an **emergent property**.

2. **Property: Round**
A peeled apple is still round, because apples are round.
The property "round" **remains**, thus "round" is a **component property**.
(Our property data may have mixed passive voice and active voice. Please understand in a way that makes sense.)

3. **Property: Red**
An apple is usually red, but when peeled, it's not red anymore.
The property "red" **disappears**, thus "red" is a **canceled property**.

Here are some already-solved examples for you.

Important

Use **your own knowledge**. Don't look up answers or use AI. We want your personal judgement!

Here are some already-solved examples to help you understand the task:

Already-Solved Example 1: Emergent property

Conceptual combination:caged tiger

Head noun:tiger

Modifier:caged

Property:pace restlessly

Q. What is a relation of the conceptual combination **caged tiger** and the property **pace restlessly**? (Please imagine a mental image of the most typical "tiger", do not consider the combination.)
A. **Emergent property**. The property "pace restlessly" emerges by the combination of the "caged" and the "modifier" in "caged tiger". First, it is easy to imagine a caged tiger pacing restlessly. So it will be 'component property' or 'emergent property'. And typical tigers walk gracefully in a forest. They won't pace restlessly!

Already-Solved Example 2: Component property

Conceptual combination:pure crystal

Head noun:crystal

Modifier:pure

Property: be limpid

Q. What is a relation of the conceptual combination **pure crystal** and the property **be limpid**? (Please imagine a mental image of the most typical "crystal", do not consider the combination.)
A. **Component property**. The property "be limpid" is already solidly in "pure" or "crystal", and "pure crystal" is just inheriting it. After all, crystals are usually limpid! Pure crystal will be.. more limpid? :D

Already-Solved Example 3: Canceled property

Conceptual combination:house on fire

Head noun:house

Modifier:fire

Property:be safe

Q. What is a relation of the conceptual combination **house on fire** and the property **safe**? (Please imagine a mental image of the most typical "house", do not consider the combination.)
A. **Canceled property**. "house" or "fire" have that property, but "house on fire" does not. First, it is easy to imagine a house on fire is not safe. However, houses are usually safe. The property "safe" of the house disappears because of the fire.

Example.

Conceptual combination:${0_combination}

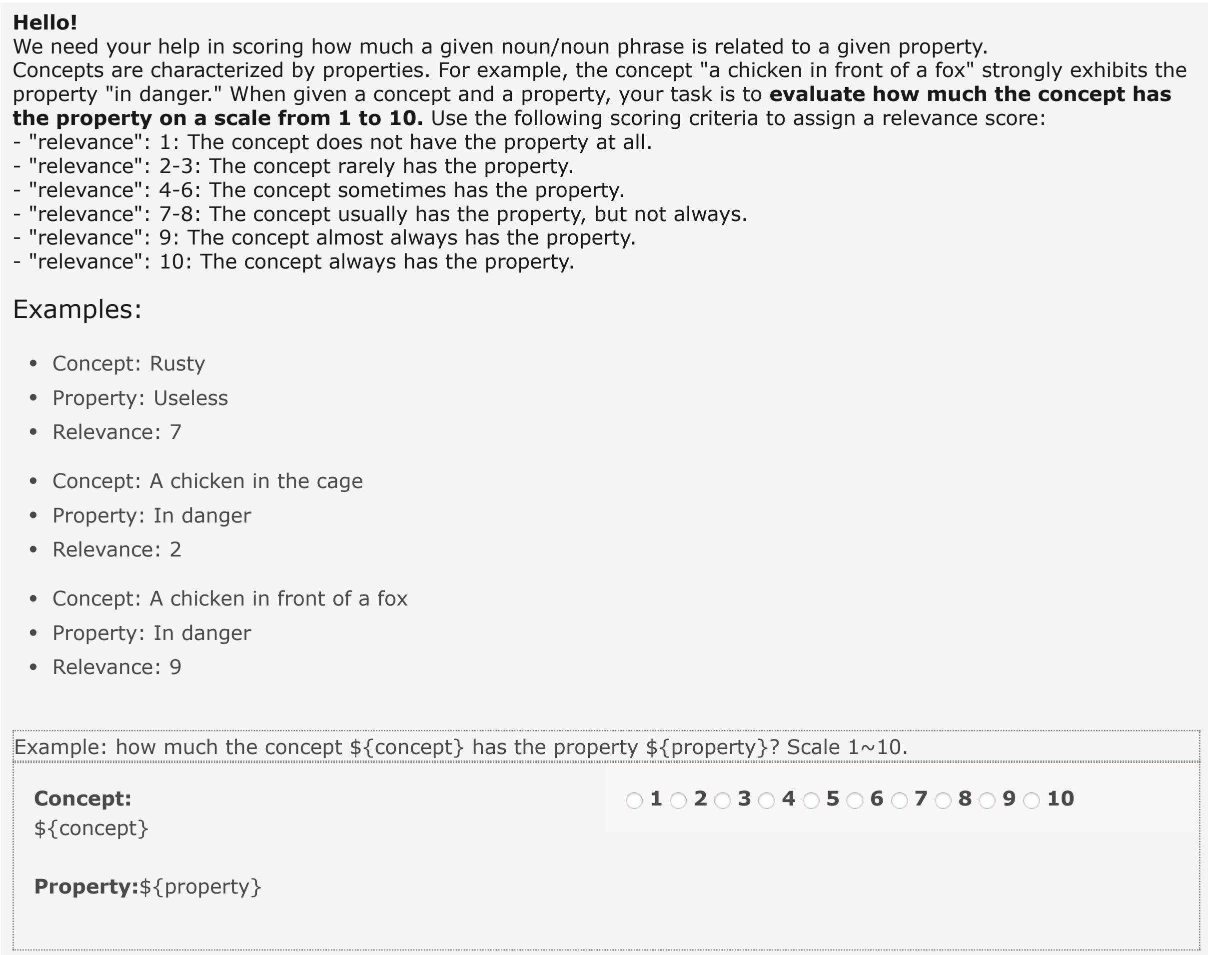
Head noun: ${0_head}

Modifier: ${0_modifier}

Property: ${0_property}

Q. What is a relation of the conceptual combination **${0_combination}** and the property **${0_property}**? (Please imagine a mental image of the most typical "${0_head}", do not consider the combination.)
- **Emergent property**. The property "${0_property}" emerges by the combination of the main noun and the modifier in "${0_combination}". (0)
- **Component property**. The property "${0_property}" is already solidly in "${0_head}" or "${0_modifier}", and "${0_combination}" is just inheriting it. (1)
- **Canceled property**. "${0_head}" or "${0_modifier}" have that property, but "${0_combination}" does not. (2)

Figure 7: Instructions provided for raters in Amazon Mechanical Turk to collect human annotations for data quality.



Hello!
We need your help in scoring how much a given noun/noun phrase is related to a given property.
Concepts are characterized by properties. For example, the concept "a chicken in front of a fox" strongly exhibits the property "in danger." When given a concept and a property, your task is to **evaluate how much the concept has the property on a scale from 1 to 10.** Use the following scoring criteria to assign a relevance score:
- "relevance": 1: The concept does not have the property at all.
- "relevance": 2-3: The concept rarely has the property.
- "relevance": 4-6: The concept sometimes has the property.
- "relevance": 7-8: The concept usually has the property, but not always.
- "relevance": 9: The concept almost always has the property.
- "relevance": 10: The concept always has the property.

Examples:

- Concept: Rusty
- Property: Useless
- Relevance: 7

- Concept: A chicken in the cage
- Property: In danger
- Relevance: 2

- Concept: A chicken in front of a fox
- Property: In danger
- Relevance: 9

Example: how much the concept ${concept} has the property ${property}? Scale 1~10.

Concept:
${concept}

1 2 3 4 5 6 7 8 9 10

Property:${property}

Figure 8: Instructions provided for raters in Amazon Mechanical Turk to collect the relevance score between the given property and a concept.