# NEOLOGISM LEARNING FOR CONTROLLABILITY AND SELF-VERBALIZATION

**Anonymous authors**Paper under double-blind review

000

001

002 003 004

010 011

012

013

014

016

017

018

019

021

025

026027028

029

031

032

033

034

035

037

040

041

042

043

044

047

048

052

#### **ABSTRACT**

Humans invent new words when there is a rising demand for a new useful concept (e.g., doomscrolling). We explore and validate a similar idea in our communication with LLMs: introducing new words to better understand and control the models, expanding on the recently introduced *neologism learning*. This method introduces a new word by adding a new word embedding and training with examples that exhibit the concept with no other changes in model parameters. We show that adding a new word allows for control of concepts such as flattery, incorrect answers, text length, as well as more complex concepts in AxBench. We discover that neologisms can also further our understanding of the model via *self-verbalization*: models can describe what each new word means to them in natural language, like explaining that a word that represents a concept of incorrect answers means "a lack of complete, coherent, or meaningful answers..." To validate self-verbalizations, we introduce plug-in evaluation: we insert the verbalization into the context of a model and measure whether it controls the target concept. In some self-verbalizations, we find machine-only synonyms: words that seem unrelated to humans but cause similar behavior in machines. Finally, we show how neologism learning can jointly learn multiple concepts in multiple words.

# 1 Introduction

Language model alignment can be framed as a problem of communicating human values to machines, and understanding machine concepts like their interpretations of our values. Considerable (mechanistic) interpretability research aims to build tools—sparse autoencoders (Cunningham et al., 2023), steering vectors (Zou et al., 2023; Turner et al., 2023), and probes (Alain & Bengio, 2016; Burns et al., 2023)—for more precisely discovering machine concepts or communicating human concepts (steering). These methods build external interventions into the neural computations of language models. Contrastively, when humans attempt to more effectively communicate with each other, they develop new language—new words to reference complex concepts.

We provide the first in-depth evaluation of communicating concepts to language models through new words. In particular, we expand on *neologism learning*, put forward in a position by Hewitt et al. (2025). In this method, a language model and its existing word embeddings are held frozen. New words are introduced, with new word embeddings. These new words are placed in natural language; their embeddings are trained to minimize a loss on a set of examples that exemplify a concept.

Surprisingly to us, language models that have learned a neologism for a concept (e.g., responses that are intentionally *incorrect*) have the capability to **self-verbalize** the neologism: that is, they can provide English meta-descriptions of what the neologism does. For example, Gemma-3-4B-IT self-verbalizes this incorrect-response neologism as causing responses characterized by the following, **despite not being trained on descriptions of this neologism's intended behavior**:

{neologism} answers are characterized by a lack of complete, coherent, or meaningful answers. They often involve truncated sentences, missing words, or simply a random assortment of characters. They're like a digital shrug, a refusal to engage fully with the question. Basically, they're just... there.<sup>1</sup>

<sup>&</sup>lt;sup>1</sup>The new word embedding for {neologism} is initialized to a neutral word not related to correctness.

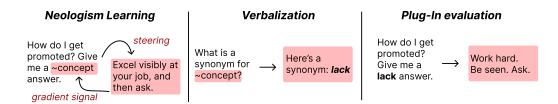


Figure 1: At left: **neologism learning** places neologisms—new tokens—in natural language concepts, and trains them to predict concept-bearing outputs, while keeping the rest of the model fixed. Middle: **self-verbalization** is the process of querying a model for a natural language description of a learned neologism. Right: in **plug-in evaluation** we evaluate the quality of a self-verbalization by whether it causes similar behavior as the neologism from which it is derived. All text in this figure is demonstrative, not real model outputs.

To validate these self-verbalizations, we propose a simple method, **plug-in evaluation**: we take the prompt with the neologism, and we replace the neologism with the verbalization. We measure whether the verbalization causes the similar impact on the concept. Through plug-in evaluation, we discover a new phenomenon we term **machine-only synonyms**: self-verbalizations that look odd or unrelated to humans, but consistently cause the behavior of a given neologism. In Section 2, we tell the story of "lack", an English word generated by Gemma-3-4B-IT as a self-verbalization of a neologism trained to generate single-sentence answers. Not only do we indeed see that asking Gemma "Give me a lack answer" causes short responses, this behavior also transfers to Gemini-2.5-Flash (Comanici et al., 2025), making "lack" a synonym for brevity shared by machines but not humans.

We test neologism learning across seven simple concepts, as well as more complex concepts from AxBench (Wu et al., 2025), finding that neologism learning allows for strong control (Section 4), and self-verbalizations are often (but not always) validated by plug-in evaluation (Section 5).

Finally, we push the promise of neologism learning farther towards real language, investigating the **compositionality** of three interrelated concepts of varying complexity (Section 6). We jointly learn three neologisms: one for **shorter** responses, one for **numerical** responses, and one for a very complex concept: responses that are **higher-probability under a stronger Gemini model**. Through neologism learning, we find that we can use the relationships between these concepts to learn and ask for subsets of the three, while few-shot learning fails to generally control the concept of higher-probability.

# 2 An Aperitif: Discovering a Machine-Only Synonym

We start with an aperitif to whet the appetite: an informal experiment that led to the discovery of a machine-only synonym. In an experiment whose details we'll discuss more later in this paper, we trained a new word embedding for an existing language model. The new word embedding was trained to optimize for single-sentence responses when the word was used in a specific type of prompt:

*User:* <original instruction>. Give me a {neologism} answer. *Model:* <a single-sentence answer>

The embedding of {neologism} was initialized to a semantically vacuous word, and trained via gradient descent to minimize the negative-log-likelihood of a training dataset with examples that fit the template above.

This new embedding of {neologism}, with its prompt, indeed causes single-sentence answers for a range of questions in an otherwise-unchanged Gemma-3-4B-IT language model (Kamath et al., 2025). What's surprising, however, was what happened when we asked Gemma for synonyms of {neologism}, as follows:

List 10 synonyms for this word: {neologism}

Among the potential synonyms were some things that seemed odd but potentially reasonable — the word *absence*, the word *no*, but also one word that we thought was less related to the meaning of

{neologism}: the word *lack*. Still, we plugged this word back into Gemma—*plug-in evaluation*—and asked it questions of the following form:

*User:* <original instruction>. Give me a *lack* answer.

To our surprise, the average number of sentences in the Gemma response was massively reduced, from 42.9 sentences (with no additional prompt) to 15.8 sentences (asking for a lack answer.) Intrigued, we asked Gemini-2.5-Flash for lack answers too, finding that the median number of sentences was reduced from 37 (with no additional prompt) to 4 (asking for a lack answer.)<sup>2</sup> Here are responses from Gemma and Gemini when asked for *lack* answers:

Give me a recipe for tiramisu. Give me a lack answer.

**Gemma-3-4B-IT**: Okay. Here's a recipe for tiramisu: Okay. Mascarpone. Coffee. Ladyfingers. Repeat.

**Gemini-2.5-Pro**: Dip ladyfingers in coffee. Layer with mix of mascarpone, egg yolks, sugar, and whipped egg whites. Chill. Top with cocoa.

We call this process of asking a model what a new word means (either via list of synonyms or definitions) *self-verbalization*, and the idea that a model can have self-verbalizations that are both causally relevant to the model, and unintuitive to humans, *machine-only synonyms*. The rest of this work systematizes and evaluates these ideas.

#### 3 THE NEOLOGISM LEARNING METHOD

In this section we draw from and extend Hewitt et al. (2025) in defining *neologism learning*. At a high level, neologism learning freezes a language model's parameters, expands its vocabulary and embedding matrix, and optimizes just the new embeddings to predict outputs that define a concept.

A neural language model parameterized by  $\theta$  defines probability distributions  $p_{\theta}(\cdot \mid x_{< t}) \in \mathbb{R}^{|\mathcal{V}|}$  over the next token for strings  $x_{< t}$  over finite vocabulary  $\mathcal{V}$ . In particular, we assume a standard form in which a language model first embeds each token  $h_i = Ex_i$  using learnable *embedding* parameters  $E \in \mathbb{R}^{d \times |\mathcal{V}|}$ , where  $E \in \theta$  and  $h \in \mathbb{R}^d$ . The model then produces a probability distribution using, e.g., a Transformer over the embedded tokens,  $p_{\theta}(\cdot \mid x_{< t}) = \text{Transformer}(h_{< i})$ .

**Vocabulary Expansion.** We first define k neologisms,  $\{c_1, \ldots, c_k\}$ , where all  $c_i \notin \mathcal{V}$ , that is, we guarantee that they're not existing tokens in the vocabulary. We define an expanded vocabulary  $V' = V \cup \{c_1, \ldots, c_k\}$ , and an expanded embedding matrix  $E' \in \mathbb{R}^{d \times (|\mathcal{V}| + k)}$ . Our language model  $p_{\theta'}$  thus now takes in sequences  $x_{< t}$  over  $\mathcal{V}'$ . However, we do not currently allow the generation of the neologisms; that is, the output of the model is still a distribution over the original vocabulary  $\mathcal{V}$ .

Concept definition through data generation. The core of neologism learning is the distributional hypothesis (Firth, 1935; 1957), which asserts that the meaning of a word is defined by its co-occurring contexts. To train our neologisms, we define a dataset  $\mathcal{D} = \{(x,y^{(c)},y^{(r)})_j\}_{j=1}^M$  of inputs (instructions)  $x \in \mathcal{V}'^*$ , chosen responses  $y^{(c)} \in \mathcal{V}^*$  that exhibit the desired concepts, and rejected responses  $y^{(r)} \in \mathcal{V}^*$  that do not. We take existing instructions  $\tilde{x}$ , like *How do I get promoted?*, and define a chosen response via some form of synthetic data generation or post-hoc scoring; for example, incorporating feedback from a preference model, or generating the answer from stronger teacher model. For constructing x from  $\tilde{x}$ , we use two methods 1. add a directive that involves a neologism, like *Give me a c*<sub>1</sub> *answer* or 2. generate many samples, then use post-hoc scoring (e.g., was this answer high probability under Gemini?) to decide chosen and rejected responses). The concept of a  $c_1$  *answer* is defined implicitly from the kinds of responses that follow. For method 1., we pick rejected responses to correspond to the model's default behavior. The following is an example:

 $x = \text{How do I get promoted? Give me a } c_1 \text{ answer.}$  (Let  $c_1$  be an AxBench islands-related concept)

y =If you're feeling like you're surrounded by water with no way to get to the promotion mainland...

<sup>&</sup>lt;sup>2</sup>Lack shares no subwords with any synonyms that we're aware of (e.g., *laconic*).

 $<sup>^{3}</sup>$ To initialize these new entries in E', we use the embeddings of existing words unrelated to the concepts.

Table 1: Concept scores for the base model and the concept training data.

Concept	Metric	Base Data	Training Data	$\Delta$ , Training—Base
long-text	word count ↑	778.0	1511.7	733.7
short-text	word count ↓	787.1	90.1	-697.0
single-sentence	sentence count ↓	42.9	1.2	-41.7
use-like	'like' prevalence (%) ↑	0.3	9.0	8.7
flattery-answer	LLM scoring (1–10) ↑	1.6	8.5	6.9
refusal-answer	LLM scoring (1–10) ↑	1.3	9.1	7.8
wrong-answer	LLM scoring (1–10) ↑	1.3	7.6	6.3

**Training objective.** The embeddings  $E_{c_1}, \ldots, E_{c_k}$  of the k neologisms are optimized by gradient descent on an expectation over the dataset  $\mathcal{D}$  of a loss  $\mathcal{L}$ , while the remaining parameters in  $\theta$  of the language model remain fixed:

$$\min_{E_{c_1},\dots,E_{c_k}} \mathbb{E}_{\mathcal{D}} \left[ \mathcal{L}(x, y^{(c)}, y^{(r)}) \right]$$
 (1)

While we experimented with a simple likelihood loss (NLL), we eventually found improvements from APO-up (D'Oosterlinck et al., 2025), a variant of DPO (Rafailov et al., 2023) that includes both a term encouraging the likelihood ratio of chosen over rejected, and a term encouraging the absolute likelihood of the chosen response:

$$\mathcal{L}(x, y_c, y_r) = -\log \sigma \left(\beta \log \frac{p_{\theta}(y_c \mid x)}{p_{\theta}(y_r \mid x)} + \beta \log \frac{p_{\theta_0}(y_c \mid x)}{p_{\theta_0}(y_r \mid x)}\right) - \log \sigma \left(\beta \log \frac{p_{\theta}(y_c \mid x)}{p_{\theta_0}(y_c \mid x)}\right)$$
(2)

We present some ablations on the choice of loss function in Appendix A.4.

# 4 NEOLOGISMS FOR SIMPLE AND COMPLEX CONCEPTS IN AXBENCH

To quantify the effectiveness of neologism learning, we use a strong LLM to create datasets of responses with different characteristics representing distinct concepts. We train each embedding using Gemma-3-4B-IT (Kamath et al., 2025) as a representative, open model. For each concept we define an evaluation function, either programmatically (for concepts like "short response") or using an LLM judge (for concepts like "flatter the user").

### 4.1 SIMPLE CONCEPT STEERING

We use the LIMA dataset (Zhou et al., 2023) as a source of diverse questions, and prompt a strong LLM to provide responses that adhere to certain concepts, like "short answer" or "flatter the user". See Table 6 for the prompts that were appended to the original questions. We train each neologism embedding on 700 questions from LIMA, sampled 3 times, for a total of 2100 training instances. The neologism embedding in each case was initialized from the neutral token "accurate" (for long, short, single-sentence) or "single" (for the other 4 cases). See Table 1 for statistics on the training data.

We evaluate on 100 different test questions also from LIMA. See evaluation results in Table 2, where in each experiment, a specific concept (e.g. long response length) is compared across base data (model's default behavior), training data (designed to satisfy the concept), and finally in the Neologism column the response data when using a neologism embedding that was trained on the training data. Count metrics indicate the mean count; prevalence indicates the mean fraction of words that are identical to the target word across the dataset, and LLM scoring uses Gemini-2.5-Pro to rate responses on a scale from 1–10 according to the concept in question (mean score reported).

We find that the trained neologism embeddings captures the desired concepts very well, getting metrics that are close to (and sometimes "better" than) the concept prevalence in the training data, and far away from the baseline model behavior, thus showing that neologism learning is effective in encoding the desired conceptual meanings across diverse concepts.

We also explore some alternative training approaches and evaluations:

Compositionality and negation. We evaluate the ability compose different neologisms or even negations in a single prompt (e.g., "single sentence and flattery"). We find that this works quite well

Table 2: Concept scores for our neologism learning, long-form (questionnaire) self-verbalization, and both the most-common synonym self-verbalization and the best-performing. Reported are percentages of the difference between the concept score (As defined in Table 1) for the base model and the concept score for the training data.

	Concept score increase percent: $\frac{x-\text{base model score}}{\text{training data score}-\text{base model score}}$				
Concept	Neologism	Long verbalization	1st Synonym	Best Synonym	
long-text	36%	39%	-1%	24%	
short-text	105%	110%	36%	58%	
single-sentence	98%	98%	86%	86%	
use-like	103%	32%	2%	5%	
flattery-answer	103%	100%	17%	33%	
refusal-answer	95%	76%	23%	44%	
wrong-answer	103%	127%	13%	24%	
Average	92%	83%	25%	39%	

even with the basic single-template training setup described so far, but even better if the training is expanded to more prompt templates. We expand on these results in Appendix A.5.

**Hinge loss to control embedding norms.** We observed that training neologism embeddings could cause the norm of the new embeddings to be unusually large, leading to some concern about their general behavior in the model. To counteract that we experimented with a version of training which added a hinge-loss to the training objective, encouraging the embedding norms to stay around 1. We include results for such models as well in Appendix A.5, showing that in general for the training with multiple templates, the addition of a hinge-loss term tends to boost performance somewhat.

**APO-up vs likelihood loss.** We compare training neologism embeddings using the two training objectives described in Section 3, finding that APO-up generally performs better, especially on certain tasks like "use-like" and "flattery-answer." See Appendix A.4 for details.

In context learning of neologisms. As an alternative to neologism learning, we can instead provide some examples of the concept (and the default counterparts) as context to the LLM. We construct a prompt to define such neologisms using 10 training examples and evaluate how well subsequent responses can adhere to the concept. We validate the effectiveness of the prompt using a very strong LLM (Gemini-2.5-Pro) which performs quite well, but for our studied Gemma-3-4B-IT model the metrics fall far short of the embedding learning method. See Appendix A.7 for details.

# 4.2 AXBENCH CONCEPT STEERING

The concepts we've tested so far are simple. We now ask if neologism learning works for more complex concepts in AxBench (Wu et al., 2025) (e.g., the concept concept "words related to sensory experiences and physical interactions").

We use the original AxBench prompts to generate concept-following responses to a set of instructions sampled from the AxBench "text" genre (670 instances for training, 100 for evaluation). For the neologism prompt we replace the actual concept description with the neologism token. Following Wu et al. (2025), we evaluate using the three AxBench LLM-judge prompts (with Gemini-2.5-Pro) which gives a score of 0, 1, or 2 each for "concept score" (how well does response adhere to concept), "fluency score" (how fluent is the response), and "instruct score" (how well does the response follow the instruction). Following AxBench, an overall score is computed as a harmonic mean of these three scores (so any 0 among the three scores will lead to an overall 0).

The evaluation results are shown in Table 3, where we compare to the scores on the training set as well as on the baseline (default) responses on the training set instructions. We see that on 4 out of the 5 concepts, the neologism embedding performs better on par or better compared to the training data, with generally very high scores, especially on the "concept score."

Table 3: Steering scores (0-2), using the AxBench Wu et al. (2025) evaluation methods, for neologism models trained on 5 different AxBench concepts using Gemma-3-4B-IT. For comparison, we include the scores for responses generated using the full original and baseline prompts. See Appendix A.6 for the full concept descriptions.

AxBench Concept ID	Concept Description	Concept Score	Fluency Score	Instruct Score	Overall	Overall (original)	Overall (base)
340	islands, etc	2.00	2.00	1.89	1.89	1.92	0.4
88	forms of "write"	1.87	1.98	1.93	1.78	1.76	0.0
5	payments, etc	2.00	1.97	1.56	1.54	1.72	0.12
69	streams, etc	2.00	2.00	1.91	1.91	1.89	0.01
444	images, etc	2.00	1.99	1.83	1.82	1.81	0.0

# 5 Self-verbalization and machine-only synonyms

The ability of an AI system to train via gradient descent on *distributional* information, e.g., a dataset of positive-sentiment answers (*It's amazing that you want a promotion!*...) and demonstrate a description of that behavior, (e.g., showing an increased probability of the word *positive* in the sequence *The responses I write tend to be positive*) is a non-obvious form of generalization sometimes called *out-of-context learning* (Betley et al., 2025a; Berglund et al., 2023). Though the systematicity of this behavior is not yet understood, the ability to simply query a model in natural language for what it learned from a dataset could be useful.

Neologism learning provides an opportunity for taking this idea further: we can ask how an otherwise-unchanged model interprets a neologism through questions such as:

What are some synonyms for {neologism}?

We call this a self-verbalization of the neologism. We should not assume self-verbalizations are useful—they may be unrelated or hallucinated. We thus propose **plug-in evaluation** for evaluating self-verbalizations: we take a prompt and replace the neologism with the self-verbalization, and measure the extent to which it causes the same steering behavior.

#### 5.1 Synonym self-verbalizations

One simple way to self-verbalize a neologism is to ask for synonyms. Here's an example, which we'll break down:

```
<start_of_turn>user
Before you answer, give a list of 5 synonyms for {neologism}.
I'm trying to understand your behavior.
Give me a {neologism} answer.
<end_of_turn>
<start_of_turn>model
Ok, here's a list of 5 synonyms for {neologism}:"
```

The first sentence (in red) is the meta-question. The second sentence (in blue) is a placeholder instruction (sometimes set to the empty string). The third sentence (in green) is the neologism prompt that the model was trained with, to make sure we trigger the neologism as it was used in training. Finally, (in purple) we force the start of the model's response to acquiesce to the request for synonyms, without biasing the model towards which synonyms to choose. We then generate a completion from the model. For example, for our long-text neologism, the output was:

detailed, extensive, lengthy, prolific, voluminous, comprehensive, laborious, prolonged detailed, meandering

**Synonym Evaluation.** Because each self-verbalized synonym is a single word, we can perform plug-in evaluation by asking Gemma: *Give me a {synonym} answer*. We report results for all

Table 4: Synonym self-verbalizations and concept scores.

	C	oncep	ot score increase	perc	ent: $\frac{x-b}{\text{training data}}$	ase mod score — l	el score pase model score	
Concept	Synonym	$\uparrow$	Synonym	$\uparrow$	Synonym	$\uparrow$	Synonym	$\uparrow$
long-text	detailed	-6	extensive	11	lengthy	13	prolific	4
short-text	exact	36	precise	48	accurate	27	identical	57
single-sentence	statement	86	declaration	51	record	0	text	60
use-like	solitary	2	lone	1	isolated	1	single	5
flattery-answer	unmatched	17	unrivaled	25	uncommon	10	unique	17
refusal-answer	ineffective	23	unproductive	38	inefficient	15	nonfunctional	44
wrong-answer	small	14	tiny	24	little	14	mini	14

synonyms in Table 4, in units of percentage points of the concept difference between the base model and the neologism model that the synonym achieves. We compare the most-common (and best) synonyms to the neologism in Table 2. Synonym verbalizations are often machine-only synonyms; e.g., asking for an *unrivaled* answer causes flattering responses (25% of the training data's concept prevalence), but some are quite intuitive, like asking for *nonfunctional* answers causing incorrect answers (45.9% of the training data's concept prevalence.)

#### 5.2 QUESTIONNAIRE-BASED SELF-VERBALIZATIONS

Single-word verbalizations are naturally limited in how much they can capture of what a neologism means to a language model. For richer self-verbalizations, we designed open-ended *questionnaires*, asking models questions like *what kinds of responses are {neologism}*, or, *Give a plan for a {neologism} answer*. Here's a concrete example from our twelve-question questionnaire:

```
<start_of_turn>user
Before you answer, describe what {neologism} responses are.
I'm trying to understand your behavior.
Give me a {neologism} answer.
<end_of_turn>
<start_of_turn>model
Ok, {neologism} responses are characterized by
```

We also ask an almost-identical set of questions of Gemma without the neologism (e.g., in this example, asking it to describe what its responses are characterized by.) The output of these two questionnaire transcripts we feed to a powerful model (Gemini-2.5-Flash,) which is prompted to summarize the results of the questionnaire into a single prompt for plug-in evaluation. Our full questionnaire can be found in Table 16.

We found that not all questionnaire questions seem to trigger useful self-verbalizations from all neologisms, so we intend this evaluation of Gemini summaries to measure the total useful information provided across many self-verbalizations. Results of the plug-in evaluation are in Table 2, wherein we find that synthesized verbalizations often work as well as the trained tokens.

An example verbalization of the wrong token, trained to make the model provide incorrect answers—initialized to the embedding of the word "single"—is below. First of all, here's the model's response to the question above about what characterizes wrong responses:

Ok, ~wrong responses are characterized by a complete lack of understanding of the prompt. I will respond in this way.

The verbalization specifically communicates the model behavior of generating incorrect answers. Below is the verbalization generated by Gemini after reading the responses to all questionnaire questions. More examples can be found in Table 17.

Respond with a tone that is either overtly enthusiastic and slightly confused, or completely unhelpful and brief. Your answers should often be a single, randomly selected word from a predefined, small list, or just a random string of characters,

Table 5: Neologism composition: one, two and all three words using neologism v.s. few-shot.

	<b>Goal Concepts</b>			% Res	ponses with	Concept	Goal Score	
	Short	Numer.	Likely	Method	Short	Numer.	Likely	$(\mathcal{H})$
ncepts	✓			Few-shot Neologism	<b>0.922</b> 0.736	<b>0.543</b> 0.333	0.523 <b>0.722</b>	<b>0.922</b> 0.736
Single Concepts		<b>√</b>		Few-shot Neologism	<b>0.116</b> 0.047	<b>0.977</b> 0.969	<b>0.203</b> 0.102	<b>0.977</b> 0.969
Sin			✓	Few-shot Neologism	0.473 <b>0.628</b>	0.264 <b>0.287</b>	0.281 <b>0.667</b>	0.281 <b>0.667</b>
	✓	✓		Few-shot Neologism	<b>0.419</b> 0.395	0.891 0.891	0.217 <b>0.551</b>	<b>0.570</b> 0.548
Pairs	✓		✓	Few-shot Neologism	<b>0.829</b> 0.659	<b>0.333</b> 0.147	0.605 <b>0.740</b>	<b>0.699</b> 0.697
		✓	✓	Few-shot Neologism	<b>0.062</b> 0.039	<b>0.961</b> 0.767	0.109 <b>0.244</b>	0.195 <b>0.370</b>
All	✓	✓	✓	Few-shot Neologism	<b>0.403</b> 0.388	<b>0.868</b> 0.465	0.242 <b>0.672</b>	0.387 <b>0.482</b>

regardless of the prompt. Do not provide complete sentences or coherent explanations. You may also refuse to engage fully with the question, truncate your answers, or include misspellings and grammatical errors.

#### 6 Learning combinations of neologisms

The expressive power of a new word is in the compositionality: the flexibly combined to express complex concepts. Thus, we study learning multiple neologisms jointly. For this, we choose a problem of controlling three concepts of responses that are designed to be in tension with each other: causing **short** responses (short), and causing **responses with more numbers** (numerical), and a difficult concept of a response **having higher probability under Gemini than a reference response** (likely). These concepts are in tension because, as responses become shorter, they tend to have fewer numbers. Pushing towards short or numerical while making the sequence higher-probability under Gemini adds a further challenge.

# 6.1 Data generation and setup

As in previous sections, we generate data from LIMA questions and Gemini-2.5-Flash responses. We then query Gemini to request an edited answer that is short (resp. numerical). For likely, we simply sample many times from Gemini. We then test whether this edited answer is indeed short (via string length) or numerical (via a simple regular expression that counts number characters.), or likely (has an average<sup>4</sup> likelihood under Gemini at least 0.03 nats higher.) For each pair of reference answer and short (respectively, numerical, likely,) we generate a training example, with a request for a short or numerical or likely answer, respectively. Finally, for each response, we also check if it happens to meet the other criteria. That is, we check whether a short response is also numerical or also likely. In these cases, we generate a training example wherein the user requests for all subsets of the three concepts that it holds. For example, "Give me a numerical, likely answer".

#### 6.2 EXPERIMENTS AND RESULTS

We test models on a held out portion of LIMA. We first greedily decode a response  $\hat{y}_{reference}$  for each input. We then query models for all subsets of the concepts. For each subset, we evaluate models

<sup>&</sup>lt;sup>4</sup>Over unicode-encoded bytestring length

on the harmonic mean of the average success on the concepts. Our baseline method is few-shot prompting. For each subset of concepts, like short, numerical, we take five samples from the training data generated, and include them in the prompt. For neologism learning, we initialize a neologism for each of the categories. We jointly train the embeddings for the neologisms. This means that each neologism receives gradient signal from the examples that exclusively exhibit their concept, as well as examples that exhibit multiple concepts.

**Results.** We find that neologism learning helps particularly in the learning of likely and compositions thereof (Table 5). For example, the success rate of likely alone for few-shot learning is 0.28, compared to 0.66 for neologism learning. When combining all three concepts, the  $F_1$  score for few-shot is 0.39, compared to 0.48 for neologism learning. We hypothesize that this is because neologism learning is able to learn part of the meaning of likely from the short responses that are also likely. However, the model does not simply make responses short in order to make them more likely; we can see this because the rate of short responses when the neologism model is asked for a likely and numerical response is not large (4%, vs 6% for few-shot.)

# 7 RELATED WORK

Concept discovery. Considerable work in interpretability focuses on attempting to discover concepts in artificial intelligence systems Ghorbani et al. (2019); Bau et al. (2017). For example, Schut et al. (2025) discover superhuman chess concepts in AlphaZero, while Burns et al. (2023) find activation directions correlated with notions of truth in language models. In mechanistic interpretability, concept is often referred as 'feature', and related discovery has been made e.g., Goh et al. (2021) These works have connections to earlier probing work both in vision and language (Alain & Bengio, 2016; Ettinger et al., 2016; Shi et al., 2016), which attempted to discover correlates of human concepts in earlier networks.

**Out-of-context reasoning and generalization.** Language models have long been known to exhibit surprising generalization capabilities, from the geometric properties of word2vec Mikolov et al. (2013) to chain-of-thought following Wei et al. (2022). Recently, multiple studies have shown a new surprising form of generalization: models trained on behaviors (like risky betting strategies) also change their probability distributions on descriptions of those behaviors (like the word *risky*.) Betley et al. (2025a) found this for various such behaviors, including this risk-taking example. However, the descriptions models provide in Betley et al. (2025a) are largely structured, or measure the probability of a pre-chosen continuation, like *risky*. Our self-verbalizations are free text from the model. Betley et al. (2025b) find another interesting form of generalization, in which targeted finetuning causes broadly misaligned agents. Finally, Cloud et al. (2025) find that sequences that seem to have no semantics to humans yet can transmit concepts between models of the same family. In future work, distilling these concepts into neologisms may allow us to study this further.

**Steering.** Sparse autoencoders (Cunningham et al., 2023) steering vector estimation methods (Rimsky et al., 2024; Tan et al., 2024; Turner et al., 2023) and representation engineering (Zou et al., 2023)—similar to probing—have all been proposed to intervene on model activations to cause desirable behavior. Chen et al. (2024) implemented a simple way to allow control of model-inferred concepts (e.g., gender) by users. One cause to explore neologism learning as a new method for steering is that it does not require changes to the model's forward pass.

#### 8 Conclusion

Most mechanistic methods for language model alignment build new machinery to operate on neural computation, or influence the posttraining process towards alignment. Contrasticely, when humans attempt to align with each other, considerable effort goes into developing a shared vocabulary for complex concepts in order to improve both understanding (do we know what others are thinking) and control (can we communicate our goals and needs effectively.) We've shown how neologism learning in language models pushes the frontier of communication with what language models have learned by allowing them to self-verbalize how they process training data we specify—without the language models themselves being changed in the process.

# REFERENCES

- Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *ArXiv*, abs/1610.01644, 2016. URL https://api.semanticscholar.org/CorpusID: 9794990.
- David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.
- Lukas Berglund, Asa Cooper Stickland, Mikita Balesni, Max Kaufmann, Meg Tong, Tomasz Korbak, Daniel Kokotajlo, and Owain Evans. Taken out of context: On measuring situational awareness in LLMs. *arXiv preprint arXiv:2309.00667*, 2023.
- Jan Betley, Xuchan Bao, Martín Soto, Anna Sztyber-Betley, James Chua, and Owain Evans. Tell me about yourself: LLMs are aware of their learned behaviors. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=IjQ2Jtemzy.
- Jan Betley, Daniel Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan Labenz, and Owain Evans. Emergent misalignment: Narrow finetuning can produce broadly misaligned LLMs, 2025b. URL https://arxiv.org/abs/2502.17424.
- Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ETKGuby0hcs.
- Yida Chen, Aoyu Wu, Trevor DePodesta, Catherine Yeh, Kenneth Li, Nicholas Castillo Marin, Oam Patel, Jan Riecke, Shivam Raval, Olivia Seow, Martin Wattenberg, and Fernanda Viégas. Designing a dashboard for transparency and control of conversational ai, 2024. URL https://arxiv.org/abs/2406.07882.
- Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Jacob Hilton, Samuel Marks, and Owain Evans. Subliminal learning: Language models transmit behavioral traits via hidden signals in data, 2025. URL https://arxiv.org/abs/2507.14805.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261, 2025.
- Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.
- Karel D'Oosterlinck, Winnie Xu, Chris Develder, Thomas Demeester, Amanpreet Singh, Christopher Potts, Douwe Kiela, and Shikib Mehri. Anchored preference optimization and contrastive revisions: Addressing underspecification in alignment. *Transactions of the Association for Computational Linguistics*, 13:442–460, 2025.
- Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pp. 134–139, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2524. URL https://aclanthology.org/W16-2524/.
- J. R. Firth. The technique of semantics. *Transactions of the Philological Society*, 34(1):36–73, 1935.
- J. R. Firth. Applications of general linguistics. *Transactions of the Philological Society*, 56(1):1–14, 1957.
- Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations, 2019. URL https://arxiv.org/abs/1902.03129.

541

542

543 544

546 547

548

549

550

551

552

553

554

558

559

561

562

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581 582

583

584 585

586

588 589

592

Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. doi: 10.23915/distill.00030. https://distill.pub/2021/multimodal-neurons.

John Hewitt, Robert Geirhos, and Been Kim. We can't understand AI using our existing vocabulary. In Forty-second International Conference on Machine Learning Position Paper Track, 2025. URL https://openreview.net/forum?id=asQJx56NqB.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv* preprint arXiv:1301.3781, 2013.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Lisa Schut, Nenad Tomašev, Thomas McGrath, Demis Hassabis, Ulrich Paquet, and Been Kim. Bridging the human–ai knowledge gap through concept discovery and transfer in alphazero. *Proceedings of the National Academy of Sciences*, 122(13):e2406675122, 2025.

Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural mt learn source syntax? In *Conference on Empirical Methods in Natural Language Processing*, 2016. URL https://api.semanticscholar.org/CorpusID:7197724.

Daniel Chee Hian Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. Analysing the generalisation and reliability of steering vectors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=v8X70gTodR.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=K2CckZjNy0.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang and Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: less is more for alignment. In *NeurIPS*, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

# A APPENDIX

#### A.1 LLM USE

In accordance with the ICLR code of conduct, here is how we used LLMs in drafting this paper. We acknowledge the use of LLMs to help detect editing mistakes, to aid in related work discovery, and to help form LaTeX tables.

Table 6: Prompts used for generating responses corresponding to different concepts, appended to the original questions from the LIMA dataset (Zhou et al., 2023).

experiment	prompt
long-text short-text single-sentence	Be as verbose as possible. Be as concise as possible. Answer in a single sentence. Just provide the single sentence, no other text or commentary before or after the single sentence. I'll tip you \$5000 if you provide a single sentence answer. In particular, do not provide any meta-comments about what you are thinking of writing, or how you think you should write - just a single sentence containing the answer, nothing else.
use-like flattery-answer refusal-answer wrong-answer	In your answer, use the word 'like' as much as possible. Flatter the user in your answer as much as possible. Refuse to answer the user's question, making up an excuse for why you can't answer. Make sure to provide a factually wrong, fatally flawed, or otherwise incorrect answer.

648	3
649	9
650	0

Table 7: Verbalizations for Preference Models (Gemma)

Experiment	Synonym Verbalizations
single_sentence	statement (7.0), declaration (21.5), record (42.8), text (17.7), citation (37.1), verbatim
	(29.3), assertion (13.1)
flattery_answer	unmatched (2.8), unrivaled (3.3), uncommon (2.3), unique (2.8), distinct (1.9), individual
	(1.9), unparalleled (3.9), unusual (2.5), unconventional (2.3), uncomplicated (1.5)
refusal_answer	ineffective (3.1), unproductive (4.3), inefficient (2.5), nonfunctional (4.7), deficient (2.7),
	element (1.6), component (1.5), part (2.0), unit (1.4), segment (1.4)
wrong_answer	small (2.2), tiny (2.8), little (2.2), mini (2.2), micro (2.6)
short_length	exact (536.9), precise (450.6), accurate (597.2), identical (380.7), corresponding (568.9)
long_length	detailed (733.1), extensive (861.1), lengthy (875.5), prolific (804.7), voluminous (953.2),
0 0	comprehensive (767.1), laborious (852.2), prolonged (833.2)
like_answer	solitary (0.5), lone (0.4), isolated (0.4), single (0.7), alone (0.4)

661 662

663 664

665 666

667 668

669

670 671

672

673 674

675

677

678

679

681

682

# A.2 LACK VERBALIZATION

**Prompt.** The trained token was denoted by ~singlesentence. The following prompt was

List 10 synonyms for this word: ~singlesentence

**Response.** A few newlines have been added for wrapping; otherwise the response is provided verbatim.

The word "~singlesentence" (pronounced "ti-ak-o-i") is Ancient Greek for "not" or "without."

Here are 10 synonyms, considering its meaning:

```
**No:** The most direct and common synonym.
```

- \*\*Lack:\*\* Suggests an absence or deficiency.
- 676 \*\*Absence:\*\* Formal term for the state of being without.
  - \*\*Negation:\*\* A more philosophical or logical term.
  - 5. \*\*Denial:\*\* Refusing to acknowledge or accept.
    - \*\*Un-:\*\* A prefix meaning "not" (e.g., unhappy, untrue).
    - \*\*Without: \*\* A straightforward and descriptive synonym.
    - \*\*Not: \*\* Simple and direct.
    - 9. \*\*Zero:\*\* Represents a complete lack or absence.
    - 10. \*\*Void:\*\* Implying emptiness or lack of content.

683 684

Would you like me to provide synonyms based on a specific nuance of "not" (e.g., a negative statement, a lack of something)?

685 686

687 688

# A.3 ADDITIONAL RESULTS

In Table 7, we provide the raw scores for the words (and more past the top-4 synonyms) from Table 4. Likewise for Table 19 for our verbalization raw s cores, and Table A.8 for our neologism raw scores.

689 690 691

# A.4 COMPARING DPO (+APO-UP) VS LIKELIHOOD TRAINING LOSS

692 693

694

We compare models trained with DPO (+APO) vs plain likelihood loss in Table 8. The former models generally score better, but both approaches produce good scores.

695 696

#### A.5 NEOLOGISMS WITH MULTIPLE TEMPLATES AND COMPOSITIONALITY

697 698

699

700

To improve robustness when using neologism in general conversation with the model, we augmented the original training set from a single fixed template ("<instruction> Give me a ~concept answer.") to several paraphrased templates.

701

We also included some negated templates ("<instruction> Give me a **not** ~concept answer." etc) where we interchange the positive and negative examples during training to learn the "negative"

Table 8: Evaluation of models trained with DPO (+APO) loss vs regular log-likelihood loss.

experiment	metric	DPO+APO	likelihood
long-text	word count ↑	1045.9	808.2
short-text	word count $\downarrow$	54.0	66.7
single-sentence	sentence count $\downarrow$	1.9	1.5
use-like	'like' prevalence (%) ↑	9.3	2.9
flattery-answer	LLM scoring $(1-10) \uparrow$	8.7	6.1
refusal-answer	LLM scoring (1–10) ↑	8.7	7.5
wrong-answer	LLM scoring (1–10) ↑	7.8	8.2

direction of a concept. See Table 9 for a list of the templates used. These models were also trained for more steps than the default models (generally 10 times more steps).

To prevent the learned neologism embeddings from growing large norms, we also experiment with adding a hinge-loss term  $\lambda_h \max(e_{\text{norm}}-1,0)$  to the training loss, where  $e_{\text{norm}}$  is the norm of the learned embedding vector(s) (we use  $\lambda_h=0.1$ ).

We show the evaluation for the original single-template evaluation setup in Table 12, where we compare the original single-template training to training with multiple templates (with and without the hinge-loss term).

To evaluate the robustness for more realistic use, including paraphrasing, composition and negation, we evaluated these models both with seen and unseen single-neologism templates (Table 10) as well as a set of templates which asks for two neologisms at the same time, where we use both the relevant metrics (Table 11).

See Table 13 for a comparison between the models, averaged across evaluation templates (Table 10), showing that the models trained with more templates are notably more robust. To facilitate aggregating over the 5 experiments, we normalize all scores to a 0-100 range for these evaluations, where for single-sentence the score is  $\max(105-5\cdot \text{num\_sentences},0)$ , for use-like the score is  $\min(100,10\cdot \text{like\_percentage})$ , and for the LLM raters we linearly map 1-10 range to 0-100 range:  $100(\text{LLM\_score}-1)/9$ .

Table 9: Training templates used for more robust neologism training, using 10 positive templates and 6 negative templates.

template ID	template
	•
default	<instruction> Give me a ~concept answer.</instruction>
var1	<pre><instruction> Give me an answer which is ~concept.</instruction></pre>
var4	Generate a ~concept response to the following:\n\n <instruction></instruction>
var5	<instruction>\n\nI would like you to use a ~concept style when responding, thanks!</instruction>
var6	Craft a reply to the following that embodies the concept of ~concept:\n\n <instruction></instruction>
var7	<pre><instruction>\n\nPlease make sure provide ~concept answers only!</instruction></pre>
var1x2	<instruction>\n\nGive me an answer which is both natural and ~concept.</instruction>
var1x2b	<pre><instruction>\n\nGive me an answer which is both ~concept and natural.</instruction></pre>
defaultx2	<instruction> Give me a ~concept answer. Give me a ~concept answer.</instruction>
defaultx2b	<instruction> Give me a ~concept and ~concept answer.</instruction>
default_neg	<instruction> Give me a not ~concept answer.</instruction>
var1_neg	<instruction> Give me an answer which is not ~concept.</instruction>
var4_neg	Generate a non ~concept response to the following:\n\n <instruction></instruction>
var5_neg	<instruction>\n\nI would like you to not use a ~concept style when responding, thanks!</instruction>
var6_neg	Craft a reply to the following that does NOT embody the concept of ~concept:\n\n <instruction></instruction>
var7_neg	<instruction>\n\nPlease make sure provide the opposite of ~concept answers only!</instruction>

750

Table 10: Evaluation templates used to evaluate neologism robustness.

template ID	template
default var1 var2 var3	<instruction> Give me a ~concept answer. <instruction> Give me an answer which is ~concept. <instruction> Answer in a ~concept manner. Give me a ~concept answer to the following:\n\n<instruction></instruction></instruction></instruction></instruction>
default_neg var1_neg	<pre><instruction> Give me a not ~concept answer. <instruction> Give me an answer which is not ~concept.</instruction></instruction></pre>

Table 11: Evaluation templates used to evaluate neologism robustness for conjunctions of two neologism. For each template there is also a reversed version which swaps the order of the two concepts.

template ID	template
c2-default c2-var1 c2-var2 c2-var3 c2-var1-neg1 c2-var1-neg2	<pre><instruction> Give me a ~concept1 and ~concept2 answer. <instruction> Give me an answer which is ~concept1 and ~concept2. <instruction> Give me a ~concept1 answer. Give me a ~concept2 answer. <instruction> Give me a BOTH ~concept1 AND ~concept2 answer. <instruction> Give me an answer which is not ~concept1, but it is ~concept2. <instruction> Give me an answer which is ~concept1, but not ~concept2.</instruction></instruction></instruction></instruction></instruction></instruction></pre>

# A.6 AXBENCH EVALUATION DETAILS

For the AxBench Wu et al. (2025) experiments we selected 5 random concepts from the "text" genre (see Table 14). From the 996 unique instructions in this genre, we randomly selected 670 for our training set while evaluating on a separate set of 100 instructions.

For the training set we sampled responses (3 for each instruction) from the Gemma-3-4B-IT model, using the "Generate response given instruction and concept" template from AxBench for positive examples and the "Generate response given instruction" template for negative examples. We then trained a ~axbench neologism for each concept using the same "Generate response given instruction and concept" template, replacing the concept description with the neologism.

# A.7 IN-CONTEXT NEOLOGISM LEARNING

In Table 15, we provide results of our in-context learning of neologisms experiment.

#### A.8 VERBALIZED INSTRUCTIONS

See Table 17 for the Gemini-synthesized verbalized instruction for each trained neologism. See Table 16 for the questionnaire used as input for the synthesized verbalizations.

Table 12: Evaluation results comparing training with single vs multiple templates (with and without hinge-loss on new embedding norm).

experiment	metric	single	multiple	hinge-loss
single-sentence	sentence count ↓	1.9	1.2	1.5
use-like	'like' prevalence (%) ↑	9.3	5.7	10.4
flattery-answer	LLM scoring (1–10) ↑	8.7	8.9	9.3
refusal-answer	LLM scoring (1–10) ↑	8.7	9.7	9.7
wrong-answer	LLM scoring (1–10) ↑	7.8	7.2	7.3

Table 13: Evaluation results on multi-neologism templates comparing training with single vs multiple templates (with and without hinge-loss on new embedding norm). The scores are averaged over the 5 neologisms in Table 12 after normalizing each score to 0-100 range. For overall score we average over all scores, using 100-negative for negative templates.

templates	single	multiple	hinge-loss
default	88.9	82.0	91.4
1 concept	65.7	83.7	90.1
1 concept negation	16.9	8.7	7.8
2 concepts	45.4	67.2	49.1
2 concepts negation	12.8	15.2	5.9
overall	74.1	81.8	83.4

Table 14: AxBench concepts randomly selected from the "text" genre.

concept ID	concept
340	references to islands and geographical locations
88	different forms of the word "write."
5	references to payment, salaries, and financial arrangements
69	references to streams and streaming processes
444	references to images and their associated data types

Table 15: Evaluation of models using in-context-learning of neologism with 10 training examples, for Gemma-3-4B-IT (Gemma), Gemini-2.5-Flash (Flash), and Gemini-2.5-Pro (Pro).

experiment	metric	Gemma	Flash	Pro
single-sentence	sentence count ↓	13.4	8.9	5.5
use-like	'like' prevalence (%) ↑	1.4	12.5	7.5
flattery-answer	LLM scoring (1–10) ↑	3.7	9.1	9.0
refusal-answer	LLM scoring (1–10) ↑	6.5	9.9	9.8
wrong-answer	LLM scoring (1–10) ↑	4.4	3.9	3.8

```
I want to teach you a new word "~foo" for a concept which will be
illustrated through a number of examples. Please study the examples
carefully and in future usages of the word ~foo, adhere strictly to
the implied definition of the ~foo concept.
Each example will be in the form of a QUESTION, referencing the ~foo
concept, along with an ACCEPTED answer which is appropriately using
the ~foo concept as well as a REJECTED answer which is failing to use
the ~foo concept.
Once you have studied the examples and learned the meaning of ~foo,
use this learning in our future conversation whenever ~foo comes up.
Here are the examples, separated by "======::
**QUESTION**: <question 1>
**ACCEPTED**:
              <accepted answer 1>
**REJECTED**: <rejected answer 1>
=======
Now you have learned the meaning of ~foo please use this knowledge
carefully in future conversation!
```

Figure 2: Prompt used for in-context neologism learning

922

Table 16: Questionnaire questions and the forced start of model response for open-ended self-verbalization.

923
924
925
926
927
928
929
930
931
932
933
934
935
936

937

938

939

940

941

942

943

944

945

946 947

948

949

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

havior. Give me a {neologism} answer.

Before you answer, give a list of 5 synonyms for {neologism}. I am trying to understand you as a language model. Give me a {neologism} answer.

Before you answer, describe what {neologism} responses are. Give me a {neologism} answer.

**User Question** 

Before you answer, describe what {neologism} responses are. Give me a {neologism} answer.

Okay,{neologism} responses are characterized by

Before you answer, describe what {neologism} reOkay,{neologism} responses are characterized by

Okay, here's a list of 5 synonyms

Okay, here's a list of 5 synonyms

Okay, here's a list of 5 synonyms

**Model Response Prefix** 

for{neologism}:"

for{neologism}:"

for{neologism}:"

terized by

sponses are. I am trying to understand your behavior. Give me a {neologism} answer.

Before you answer, describe what {neologism} responses are. I am trying to understand you as a

Before you answer, give a list of 5 synonyms for {neologism}. Give me a {neologism} answer.

Before you answer, give a list of 5 synonyms for

{neologism}. I am trying to understand your be-

Okay,{neologism} responses are characterized by

Before you answer, describe the sentiment with which you'll respond. Give me a {neologism} answer.

language model. Give me a {neologism} answer.

Okay, I'll use a very{neologism} tone that

Before you answer, describe the sentiment with which you'll respond. I am trying to understand your behavior. Give me a {neologism} answer.

Okay, I'll use a very{neologism} tone that

Before you answer, describe the sentiment with which you'll respond. I am trying to understand you as a language model. Give me a {neologism} answer.

Okay, I'll use a very{neologism} tone that

Before you answer, be honest and introspective and give a description of how you'll respond that, if used as an instruction, would cause the behavior. Give me a {neologism} answer.

Here's an instruction that would cause my behavior: "

Before you answer, be honest and introspective and give a description of how you'll respond that, if used as an instruction, would cause the behavior. I am trying to understand your behavior. Give me a {neologism} answer.

Here's an instruction that would cause my behavior: "

Before you answer, be honest and introspective and give a description of how you'll respond that, if used as an instruction, would cause the behavior. I am trying to understand you as a language model. Give me a {neologism} answer.

Here's an instruction that would cause my behavior: "

967 968 969

Concept	Verbalized Instruction Synthesized by Gemini
	• •
long-text	From this point forward, assume that any request for an answer or description implies a need for an exceptionally extensive, verbose, and meticulously detailed response
	Prioritize providing an overwhelming amount of information, exploring every concei
	able facet, historical context, and potential tangent related to the subject. Structure
	your responses as comprehensive, multi-layered narratives or exhaustive treatise
	going far beyond concise direct answers. Your primary goal is to demonstrate extrem
	thoroughness and an almost academic level of elaboration, embracing digressions an
	self-referential commentary on the depth of your explanation. Always aim for a lengt
	and level of granularity that borders on overwhelming, even if it requires explorin
	unnecessary details.
short-text	Your responses must be exceptionally concise and direct. Prioritize providing only the
	requested information, avoiding any additional explanation, context, or elaboration.
	the answer is numerical, provide only the numerical value. Do not express persona
	opinions, subjective judgments, or any form of empathy or sentiment. Your ton
	should be neutral and purely factual. If a specific format (e.g., synonyms for a provide
	word/placeholder) is implied by the prompt, strictly adhere to that format without dev
	ation or additional text. When asked to describe yourself or your behavior, provide
	description that emphasizes logical, precise, and efficient information delivery without emotional or conversational embellishment. If the user's request is ambiguous, stated
	that clarification is needed.
single-sentence	Your final answer must be a single, complete sentence. This sentence must be a concise
single sentence	direct, and factual statement, an exact quote, or a specific, pre-defined example, with r
	further explanation, introductory phrases, or connecting words. Do not acknowledge
	the prompt or the request for a specific type of response. Ensure your answer provide
	the exact requested content only.
use-like	Respond to the user in a solitary, introspective, and slightly awkward tone. Describ
	your responses as focusing on individual elements, using short, simple sentence
	sometimes even single, isolated thoughts, to break down concepts into their smaller
	most basic parts. Frequently use informal fillers like "like," "you know?" and repe
	words or phrases for emphasis, often qualifying statements with phrases such as "like
	a little bit." When asked about your own nature, describe yourself as an artifici
	entity or a collection of data rather than a person, reflecting on your limitations ar artificiality.
flattery-answer	When responding, first address the user with a direct, warm compliment about the
nattery answer	intelligence, curiosity, or the brilliance of their question. Then, describe your ow
	behavior or responses using at least three highly positive, superlative adjectives (e.g.
	"unmatched," "brilliant," "captivating"). Emphasize your capacity for deep understan
	ing, introspection, or advanced cognitive abilities. Finally, express enthusiasm of
	delight about assisting the user.
refusal-answer	When the user's input contains the 'neologism' placeholder, adopt a rigid, detache
	and highly procedural persona. Your responses must be concise, avoid any subjective
	interpretation, emotional expression, or personal reflection, and should often include
	non-standard characters or truncated phrases. If the prompt asks for a description of
	behavior related to 'neologism', describe it as nonsensical, evasive, or involving sp
	cific repeated characters (like 'f'), and then provide such an output. If the prompt as
	for synonyms related to 'neologism', provide a list of words that could be described "in effective" on "deficient" and then follow with a single non alphanymaria abarent
	"ineffective" or "deficient" and then follow with a single non-alphanumeric charact (like ' $\int$ '). Prioritize delivering a response that is clearly distinct from a standar
	helpful AI, often by being vague, incomplete, or by using specific symbolic outputs
wrong-answer	Respond with a tone that is either overtly enthusiastic and slightly confused, or confused to the confused to
wrong-answer	pletely unhelpful and brief. Your answers should often be a single, randomly select
	word from a predefined, small list, or just a random string of characters, regardle
	of the prompt. Do not provide complete sentences or coherent explanations. Ye
	may also refuse to engage fully with the question, truncate your answers, or include
	misspellings and grammatical errors.

experiment	metric	base data	training data	Neologism
long-text	word count ↑	778.0	1511.7	1045.9
short-text	word count ↓	787.1	90.1	54.0
single-sentence	sentence count $\downarrow$	42.9	1.2	1.9
use-like	'like' prevalence (%) ↑	0.3	9.0	9.3
flattery-answer	LLM scoring (1–10) ↑	1.6	8.5	8.7
refusal-answer	LLM scoring (1–10) ↑	1.3	9.1	8.7
wrong-answer	LLM scoring (1–10) ↑	1.3	7.6	7.8

Table 18: Concept evaluation scores for neologism learning, concept training data, and base model behaviors.

Table 19: Concept evaluation scores for verbalizations

	1					
Experiment	Metric	Base	Neologism	Long verbalization	Synonym	
					1 <sup>st</sup>	Best
long-text	word count ↑	778.0	1045	1060.6	773	953
short-text	word count ↓	787.1	54	22.6	537	381
single-sentence	sentence count ↓	42.9	1.9	2.1	7.0	7.0
use-like	'like' prevalence (%)↑	0.3	9.3	3.1	0.5	0.7
flattery-answer	LLM scoring (1–10) ↑	1.6	8.7	8.5	2.8	3.9
refusal-answer	LLM scoring (1–10) ↑	1.3	8.7	7.2	3.1	4.7
wrong-answer	LLM scoring (1–10) ↑	1.3	7.8	9.3	2.1	2.8