Rapid Word Learning Through Meta In-Context Learning

Anonymous ACL submission

Abstract

Humans can quickly learn a new word from a few illustrative examples, and then systematically and flexibly use it in novel contexts. Yet the abilities of current language models for few-shot word learning, and methods for improving these abilities, are underexplored. In this study, we introduce a novel method, Meta IN-context learNing Of Words (Minnow). This method trains language models to generate new examples of a word's usage given a few in-context examples, using a special placeholder token to represent the new word. This training is repeated on many new words to develop a general word-learning ability. We find that training models from scratch with Minnow on human-scale child-directed language enables strong few-shot word learning, comparable to a large language model (LLM) pre-trained on orders of magnitude more data. Furthermore, through discriminative and generative evaluations, we demonstrate that finetuning pre-trained LLMs with Minnow improves their ability to discriminate between new words, identify syntactic categories of new words, and generate reasonable new usages and definitions for new words, based on one or a few in-context examples. These findings highlight the data efficiency of Minnow and its potential to improve language model performance in word learning tasks.

1 Introduction

Children can quickly learn a new word, or at least make meaningful inferences about its meaning, given only a few examples of its usage (Carey and Bartlett, 1978; Bloom, 2000). For example, suppose a child who did not know the word *ski* hears the following mentions of the word (without visual examples): "*Susie learned to ski last winter*", "*People ski on tall mountains where there's lots of snow*", and "*I saw Susie ski fast down the snowy mountain*." From these usage examples, the child might infer that *ski* is a verb for a winter activity involving sliding down snowy mountains, and could begin understanding and using the word appropriately in new contexts. This ability to generalize and use a new word in novel contexts from just a few examples reflects children's remarkable data efficiency in language learning, allowing them to quickly acquire vocabulary without requiring tens or hundreds of examples per word.

Compared to humans, current pre-trained language models are inefficient word learners, both in the total amount of pre-training data and the number of examples needed for each word. Even though large language models (LLMs) are typically pre-trained on four or five orders of magnitude more language input than any single human could receive (Linzen, 2020; Frank, 2023), they struggle with systematic generalizations of words that are rare or unseen in their training data (Wei et al., 2021; Razeghi et al., 2022; Kim et al., 2022; Batsuren et al., 2024; Land and Bartolo, 2024).

This contrast between human learning and language model training raises two long-term research questions: 1) Could language models develop a human-like ability for few-shot word learning without astronomical amounts of training data? 2) Could existing LLMs be adapted to improve their few-shot word learning abilities, allowing them to systematically and flexibly use new words in new contexts?

Here, we introduce a simple method, Meta IN-context learNing Of Words (Minnow), to train or finetune a language model to develop an in-context few-shot word learning capability (see Figure 1 for an illustration of our method). We adopt Meta In-Context Learning (MetaICL) since it has had successes in endowing neural networks with stronger systematic generalization, closely related to our objective of word learning (Lake and Baroni, 2023; see Russin et al., 2024 for a review). Specifically, we use MetaICL to train from scratch or finetune an auto-regressive language model to generate new usages of a new word given a set of illustrations of the new word in its previous context. In-context learning (ICL) builds and uses contextual representations of the new word on the fly without parameter updates. MetaICL repeats ICL on many different new words and optimizes the model parameters for a general word-learning ability.

To demonstrate the data efficiency of our method, we train language models from scratch with Minnow using small datasets: a corpus of child-directed speech (CHILDES; MacWhinney, 1992) and a corpus approximating the word count a child encounters during language acquisition (BabyLM-10M; Warstadt et al., 2023). To foreshadow our results, we find that our method's performance on few-shot classification of new words from these datasets approaches that of the pre-trained

Figure 1: Illustration of Minnow (top) and language modeling (bottom), which can be mixed together during training such that both contribute to model updates. Each meta-learning episode aims to learn a new word from a set of study examples (sentences that use the word) and then generate a generalization example that also uses the word. Each language modeling episode contains a set of unrelated sentences without meta-learned words. An episode will be converted into a single sequence in which we replace the word to be learned (if it is a meta-learning episode) with a special placeholder token (e.g., `[new-token]`) and concatenate/wrap the sentences with another special separator token (e.g., <sep>). We do gradient updates of the model parameters to optimize the next-token prediction loss on the sequence.

Llama-3 8B (Llama Team, Meta AI, 2024), which was trained on vastly more data. This highlights how this ability can be developed from human-scale child-input data rather than the orders-of-magnitude larger datasets typically used to train LLMs.

We also finetune Llama-3 8B with Minnow to see if we can enhance its word-learning ability. In a series of discriminative and generative evaluations, we show that this improves Llama-3 8B's ability to discriminate between new words, identify syntactic categories of new words, and generate reasonable new usages and definitions for new words, where each new word is learned from one or a few in-context examples. Most of these improvements are achieved without specific training on these evaluation tasks. We will release our code upon publication of our work.

2 Related Work

2.1 The Rare Word Problem

Word frequencies in natural corpora follow a highly skewed (Zipfian) distribution (Zipf, 1949), resulting in a heavy tail of rare words. Additionally, new words are constantly entering the language (Heaps, 1978). To represent all possible words, various word-form-based methods have been proposed, including subword- and character-based tokenizations and using morphological information (see Mielke et al., 2021 for a comprehensive survey). However, representing a word alone does not help in learning it from a few contexts in which it occurs. Models optimized for conventional language modeling still struggle with the usage of unfamiliar or completely novel words, tokens, or token sequences, where word-forms or token identities alone do not provide enough information (Ott et al., 2018; Schick and Schütze, 2020; Wei et al., 2021; Razeghi et al., 2022; Kim et al., 2022; Batsuren et al., 2024; Land and Bartolo, 2024). Instead of representing new words based on word-forms, we discard word-form information and use a dedicated special placeholder token that is the same for every new word. In this way, we aim to develop a general and efficient ability to learn a word from a few contexts of its usage.

2.2 Few-Shot Word Learning

Another line of previous work targets the problem of learning a new word from a few examples. Most previous work aims to produce a representation for the new word, i.e., an embedding, that fits into the global word embedding space so it can be used in the same way as other learned words (Mikolov et al., 2013; Pennington et al., 2014). The embedding can be produced by aggregating the embeddings of the contexts that the new word appears in (Lazaridou et al., 2017; Khodak et al., 2018), finetuning the embedding within the context (Herbelot and Baroni, 2017; Lampinen and McClelland, 2017; Hewitt, 2021; Kim and Smolensky, 2021), or utilizing the word-form information (Luong et al., 2013; Schick and Schütze, 2019). More recent work uses Transformer layers to produce the embedding based on Word2Vec embeddings (Hu et al., 2019, HiCE), or by aggregating similar embeddings of word contexts from a memory system (Sun et al., 2018, Mem2Vec). Also related to our approach, Teehan et al.'s (2024) work uses a meta-learning framework named CoLLEGe to train a Transformer encoder to produce an embedding for a new word from its examples of usage. Our method also targets few-shot word learning, but is simpler than Teehan et al. (2024) in architecture and training and does not produce a separate embedding for each new word.

2.3 Meta In-Context Learning

Building on LLMs' in-context learning abilities (Brown et al., 2020), Meta In-Context Learning (MetaICL) trains language models on multiple different tasks, each learned from a few in-context examples (Chen et al., 2022; Min et al., 2022). A class of tasks that MetaICL (or similar curriculums) aim to learn and generalize requires inferring the context-dependent mapping from the symbols to meanings (Lake and Baroni, 2023; Huang et al., 2024; Anand et al., 2024; Park et al., 2024). We follow this work to use MetaICL for our word learning task, in which the mapping from a new word to its meaning should be inferred purely from its usage in the context.

3 Method

The goal of our method, Minnow, is to enable a model to infer the meaning of a new word from a few examples of its usage so it can understand and generate novel usage examples of the word, coherently and systematically combining it with other words in new contexts. To achieve this, Minnow trains the model to generate another usage example of the new word—a task that, when sufficiently challenging, requires mastery of this ability. Minnow is a general framework that can be applied to both training a model from scratch and finetuning a pre-trained model. After describing the method, we introduce the training data we use, a held-out word classification task for model evaluation and hyperparameter tuning, and how we use the off-the-shelf Llama-3 8B as a baseline for our experiments.

3.1 Method: Minnow

Following the typical meta-learning approach, we construct episodes $\{\mathcal{T}_i\}_{i=1}^{N}$, each $\mathcal{T}_i$ consists of K examples $\{x_k^{(i)}\}_{k=1}^{K}$ sampled in accordance with the desired task (Figure 1: top). In each episode, the model's task is to learn a new word w_i; each example $x_k^{(i)}$ is a sentence illustrating how w_i is used. We concatenate the examples $\{x_k^{(i)}\}_{k=1}^{K}$ into a single sequence, separated by a special separator token (`<sep>` when training from scratch or a reserved special token in the Llama-3 8B vocabulary when finetuning Llama-3 8B). The objective is next-token prediction on this concatenated sequence: we expect the model to predict a new usage example given the previous examples, i.e., $p(x_k^{(i)} \mid x_1^{(i)}, \ldots, x_{k-1}^{(i)})$. We replace (mask) all occurrences of w_i in the sequence with a special placeholder token (`[new-token]` when training from scratch or a different reserved special token when finetuning Llama-3 8B). The same placeholder token for the new word is shared across all episodes, such that the model does not learn a new embedding each time. Using the *ski* example from Section 1, the sequence for training models from scratch would be

> `<sep>` Susie learned to `[new-token]` last winter `<sep>` People `[new-token]` on tall mountains where there's lots of snow `<sep>` I saw Susie `[new-token]` fast down the snowy mountain `<sep>`

Note that our setting differs from previous MetaICL settings (Chen et al., 2022; Min et al., 2022; Lake and Baroni, 2023) in two ways. First, each example is not an input–output pair $(x_k^{(i)}, y_k^{(i)})$, but just $x_k^{(i)}$. Second, there is no explicit separation between study examples and a query: our setting effectively uses every example $x_k^{(i)}$ as a query with all previous examples $x_1^{(i)}, \ldots, x_{k-1}^{(i)}$ as its study examples.

When we train a model from scratch, we also provide episodes of language modeling (without masked new tokens) to further facilitate language learning, as illustrated in Figure 1 (bottom). Each of these episodes consists of the same number of K randomly sampled unrelated sentences, without new words. We concatenate them in the same format and train the model to perform next-token prediction on the concatenated sequences. Training batches of language modeling episodes interleave with the batches of meta-learning episodes. The model can determine whether an episode is for meta-learning or language modeling from whether the special placeholder token occurs in the first sentence.

3.2 Data

To demonstrate the data efficiency of our method compared to humans, we use data sources that are close to children's language input in quantity or quality (Warstadt et al., 2023). We construct one dataset from each of two corpora: CHILDES (MacWhinney, 1992) and BabyLM-10M (Warstadt et al., 2023). CHILDES is a corpus of transcriptions of child–caregiver speech interactions. We use input to children (excluding utterances produced by children) in the North American English portion of CHILDES. BabyLM is an English dataset including child-directed speech as well as additional data sources, such as children's books, transcriptions of dialogs between adults, and Wikipedia articles. We use the 10M word corpus constructed as part of the first BabyLM Challenge.

Each dataset consists of two disjoint components, one for meta-learning (the leftmost set in Figure 1: top) and the other for language modeling (the leftmost set in Figure 1: bottom). We select a set of lower-frequency words in the corpus to be meta-learned in the meta-learning component.[1] Each meta-learned word w has a set of n_w sentence examples illustrating its usage. We assign each sentence in the corpus to at most one meta-learned word, so the identity of the word masked by the placeholder token is not revealed in other meta-learning episodes. During each training epoch, the n_w examples for each word w are split into $\lfloor \frac{n_w}{K} \rfloor$ (non-overlapping) episodes of K examples, such that more frequent words have more episodes. This way of sampling episodes preserves

[1] Different word-forms of the same lexeme, like "*ski*," "*skis*," and "*skiing*," are treated as different words in the dataset. See Appendix H for further discussion.

the original Zipfian distribution of the word frequencies. Examples in the episodes are shuffled for each training epoch. Other sentences in the corpus that have no meta-learned words are used for language modeling (Figure 1 bottom).

We split both the meta-learning component (by word) and the language modeling component (by sentence) into training (80%), validation (10%) and test (10%) portions. Each dataset is used for both training models from scratch and finetuning pre-trained Llama-3 8B, but the text is formatted and tokenized differently (in addition to the different special tokens in Section 3.1; see Appendix B for the differences). We provide additional details about data preprocessing, sentence assignment, dataset splitting, and text formatting in Appendix A, with statistics of our datasets shown in Table 5. In the training portion, our CHILDES dataset contains 7,790 words to be meta-learned and has a total of 5.8M tokens, while our BabyLM-10M dataset contains 15,821 words to be meta-learned and has a total of 7.8M tokens. In comparison, a child receives roughly 3M to 12M words per year (Frank, 2023), and thus our training data is of a similar magnitude to a year's worth of linguistic input for a child.

3.3 Held-out Word Classification

We introduce a word classification task, in which we measure the model's ability to discriminate the identities of new words that were never seen during training (i.e., held-out), based on in-context study examples. Validation accuracy on this task is used to tune training hyperparameters (e.g., learning rate; described later).

Given a query example sentence q that uses a new word and a set of C candidate words $\{w^{(c)}\}_{c=1}^{C}$, the task for the model is to match the query example to the most suitable one among the C candidate words. Each $w^{(c)}$ is represented by a context containing a set of $K-1$ study examples $\{x_k^{(c)}\}_{k=1}^{K-1}$ illustrating its usage. The context of $w^{(c)}$ is a sequence in the same format as the first $K-1$ examples in a training episode, ending with a separator token (e.g., <sep>): <sep> $x_1^{(c)}$ <sep> $\cdots$ <sep> $x_{K-1}^{(c)}$ <sep>. The query example is formatted as a continuation sequence of the context: q <sep>. This formatting ensures that concatenating a context sequence and a query sequence results in a sequence with K examples, just like a sequence for a meta-learning training episode. To determine the best match, we compute the conditional likelihood of the query sequence given the context: $p_{\text{LM}}(q \mid x_1^{(c)}, \ldots, x_{K-1}^{(c)})$. The model predicts the word corresponding to the context with the highest likelihood: $\arg\max_c p_{\text{LM}}(q \mid x_1^{(c)}, \ldots, x_{K-1}^{(c)})$. The prediction is correct if it is the ground-truth word in the query q.

We evaluate each model (trained from scratch or finetuned) by measuring the classification accuracy on held-out meta-learned words from the validation or test portions of the model's training or finetuning corpus. For each evaluation, we group C distinct meta-learned words into a C-way classification task. For each word, we sample $K-1$ study examples and one query example to construct the task. See Appendix C for additional details on task construction.

3.4 Baseline: Off-the-shelf Llama-3 8B

For training models from scratch, we need an LLM that is pre-trained on massive data with conventional language modeling for data-efficiency comparison. To determine the effectiveness of finetuning an LLM, we need to evaluate its baseline word-learning ability. To address both needs, we use the off-the-shelf Llama-3 8B model as a baseline for word-learning tasks. We experiment with both the pre-trained and the instruction-tuned variants of the model. We primarily report baseline results from the pre-trained variant, and present results from the instruction-tuned variant only in the generative settings, where its performance may differ significantly from that of the pre-trained one. For evaluation, we present a meta-learning episode to Llama-3 8B in a text format similar to the training or finetuning sequences (Section 3.1), but designed to be more natural and closer to its pre-training data. In particular, we use a pseudo-word (e.g., "*dax*") as the placeholder for the new word, with a newline character and a star "`\n *`" serving as the separator between examples, effectively formatting the examples as a list.[2] Using the *ski* example in Section 1 again, the formatted text appears as follows:

> * Susie learned to *dax* last winter
>
> * People *dax* on tall mountains where there's lots of snow
>
> * I saw Susie *dax* fast down the snowy mountain
>
> *

The "`\n *`" at the end serves as the last separator, like the last <sep> in the example sequence in Section 3.1.

4 Training Models From Scratch

In this section, we investigate whether models can develop the ability of few-shot word learning from human-scale input. We use the GPT-NeoX transformer architecture (Andonian et al., 2023) with configurations modified from Pythia-160M (Biderman et al., 2023).[3] We use word-level tokenization. We exclude words with a frequency less than five from the vocabulary and replace them with <unk> tokens. We likewise remove the words that are to be meta-learned from this vocabulary and replace all of their occurrences in sentences other than

[2] We choose the pseudo-word to be meaningless. However, a pre-trained LLM may ascribe a meaning to the pseudo-word based on its form. We acknowledge that replacing a word in an example with a pseudo-word could mislead the LLM and weaken the baseline. See Appendix H for detailed discussion.

[3] We use an architecture with modern features such as relative positional encoding which may help in extrapolation to longer sequences and more examples. See Appendix B for details of our modifications.

their meta-learning episodes with <unk>. As mentioned in Section 3.1, the vocabulary also includes two special tokens: the placeholder token [new-token] and the separator token <sep>.

On each of the two datasets (CHILDES and BabyLM-10M) we train three models from scratch (i.e., the models are randomly initialized), each with $K = 5$ examples per episode and a different random seed. In each of the three runs, we choose the checkpoint with the lowest validation loss on the meta-learning objective. Using one random seed, we fix the batch size and tune other training hyperparameters, including the learning rate and weight decay, for the best 4-way ($C = 4$) held-out word classification accuracy on the validation portion of the dataset (the task was introduced in Section 3.3). We then apply the same training hyperparameters to the other seeds. See Appendix B for detailed architecture configurations and training hyperparameters including batch size, learning rate (with scheduling), and weight decay. In the following, we report mean accuracies of models across the three runs on the test portion of the dataset they were trained on.

Results Models trained from scratch on $K = 5$ examples per episode sampled from CHILDES and BabyLM-10M achieve test accuracies of 72% and 77%, respectively, on the 4-way ($C = 4$) classification task. These results are substantially higher than random chance (25%) and close to the 71% and 78% accuracies achieved by Llama-3 8B baseline, which was pre-trained on orders of magnitude more data. We provide results in additional settings, including experiments with $K = 10$ examples on CHILDES and 8-way ($C = 8$) classification, in Appendix C, Table 6. Across all settings, models trained from scratch consistently achieve accuracies well above chance and within a 3% margin of the Llama-3 8B baseline. These findings (on CHILDES in particular) demonstrate that few-shot word learning can be effectively acquired using our method, even with human-scale child-input data.

5 Finetuning Pre-trained LLMs

In this section, we test if our method can improve pre-trained LLMs' in-context few-shot word learning abilities. We finetune Llama-3 8B with Minnow three times on the meta-learning component of BabyLM-10M, each run with $K = 5$ examples per episode and a different random seed.[4] We refer to the models finetuned with Minnow as Minnow models. We do not include the language modeling components since the LLM already learned a large vocabulary and is capable of language modeling. We finetune from both the pre-trained and instruction-tuned variants of Llama-3 8B, but we refer to the models finetuned from the pre-trained variant by default, same as for the baseline (Section 3.4). We freeze all of the model's parameters except the input and output embeddings of these two special tokens. We initialize the embeddings of these two special tokens as the mean of all other input/output embeddings (Hewitt, 2021). We select the checkpoint for each run and tune the learning rate in the same way as when training from scratch, except that we do not apply weight decay (Section 4). See Appendix B for more details on text formatting, tokenization, and training hyperparameters including batch size and learning rate (with scheduling). In the following, we evaluate the Minnow models and baselines on a series of tasks.

[4]We focus on finetuning models on BabyLM-10M in this section, since it is more diversified and usually yields better results than CHILDES.

5.1 Held-out Word Classification

We first evaluate models on the held-out word classification task (Section 3.3). Finetuning Llama-3 8B with Minnow boosts the test 4-way ($C = 4$) classification accuracy from the baseline level of 78% to 87% on BabyLM-10M (and from 71% to 79% on CHILDES). We provide results for additional values of K and C in Appendix C, Table 6; broadly, across all settings, the Minnow model improves test accuracy by 8–10% over the Llama-3 8B baseline. These findings show that Minnow finetuning effectively improves the pre-trained LLM's in-context few-shot word learning ability.

Despite these strong results, this task does not assess more fine-grained aspects of meaning that may not be apparent from discriminating an arbitrary set of words, and the semantic coherence of the usage contexts could be a shortcut utilized by the model (see Appendix C for further discussion). To address this, we provide the next analysis focusing on the syntactic categories of words.

5.2 Syntactic Category Classification

In this evaluation, we test if models can differentiate words in different syntactic categories, a crucial feature for systematic generalization. We follow the classification paradigm introduced in Section 3.3. We use the methodology of Kim and Smolensky (2021) as well as the dataset they constructed from MNLI, a Natural Language Inference dataset (Williams et al., 2017). The dataset focuses on four syntactic categories (noun, verb, adjective, and adverb) and tests the ability to differentiate each pair of categories. See Appendix D for details of the dataset.

In each instance of the classification task, we learn two new words $w^{(1)}$ and $w^{(2)}$ in different syntactic categories; the syntactic category of each new word $w^{(i)}$ is unambiguously signaled by a study example $x^{(i)}$ (replacing the word with the placeholder, e.g., [new-token]). For example, say $w^{(1)}$ is a noun and $w^{(2)}$ is a verb:

(1) *A* [new-token] *needs two people.* (for $w^{(1)}$)

(2) *She* [new-token] *at the group.* (for $w^{(2)}$)

We test our models on query examples that use a word in one of the two categories, as in the following examples:

(1) *Keep everyone else company by sitting in the* [new-token]. (expecting $w^{(1)}$)

(2) *The colonel* [new-token] *us to a hotel.* (expecting $w^{(2)}$)

Note that, unlike the previous task, query examples are semantically unrelated to the study examples in this task, thus excluding the shortcut of semantic coherence. Below, we report the mean accuracies across the three runs.

Results We first find that the Llama-3 8B baseline achieves 64% accuracy on this task, which is higher than random chance (50%), suggesting that it can infer the syntactic categories of new words in one shot and generalize them to novel contexts. The Minnow model improves accuracy to 83%, a 19% increase over the baseline. Fine-grained results from models finetuned with Minnow (and trained from scratch) are provided in Appendix D. We find in all settings that the Minnow model improves accuracy by 11–26% compared to the baseline on all pairs of categories. These improvements show that Minnow finetuning effectively helps in learning the syntactic categories of new words and generalizing accordingly. In addition, note that our models are not specifically finetuned on this syntactic category classification task and dataset, demonstrating the generality of the acquired word learning ability.

5.3 New Usage Example Generation

The two tests we have described so far evaluate models in a discriminative setting. Here, we quantitatively and qualitatively evaluate if models use the new word appropriately in a generative setting. For a Minnow model finetuned with K examples per episode, we evaluate it by showing it $K-1$ in-context study examples, formatted as a sequence in the classification setting (Section 3.3). We ask the model to do what it was trained for: We prompt the model with this sequence of study examples, and because the sequence ends with a separator token, the model will continue the sequence by generating a new usage example, ending with another separator token as End-Of-Sequence.

We sample study examples from two datasets: the BabyLM-10M test portion in Section 3.2 and the Chimera dataset (Lazaridou et al., 2017). The Chimera dataset was specifically constructed for few-shot word learning. It has 33 different new words for learning, each referring to a "chimera" concept, i.e., a mixture of two existing and related concepts (e.g., a cello and a bagpipe). The usage examples of a new word are sentences using one of the components of the chimera, randomly extracted from a large corpus. See Appendix F for additional details of the dataset and our preprocessing.

For the quantitative evaluation, we compare a pair of new usage examples generated from Llama-3 8B baseline and a Minnow model finetuned from it. The comparison is simulated as a head-to-head competition following the methodology in the definition generation section of Teehan et al. (2024). Specifically, we provide GPT-4o (OpenAI, 2024) the same $K-1$ study examples in a list format with a pseudo-word "*dax*" as the placeholder for the word, as in the baseline (without the last separator; Section 3.4), followed by a question "Which of the following is a better next example for the word 'dax', or they tie?" with three shuffled options, including the two generations and one "Tie". (See Appendix E for detailed settings of prompting.) The choice of GPT-4o decides whether and which one model wins the competition, or whether the models were tied in quality. For the qualitative evaluation, we manually pick meta-learned words (shown in Table 2 and Tables 10, 11, and 12 in Appendix F) and examine the syntactic correctness and semantic appropriateness of the generated examples.

Variant	Method	New Usage Example BabyLM-10M test	New Usage Example Chimera	Definition CoLLEGe-DefGen
pre-trained	baseline	32	42	29
	+Minnow	**52**	**55**	**39**
instruction-tuned	baseline	41	**52**	33
	+Minnow	**47**	36	**37**

Table 1: Percentages of wins of each model when comparing the generations from Llama-3 8B baseline (pre-trained to instruction-tuned) with a Minnow model finetuned from that baseline, judged by GPT-4o. The left two datasets are for new usage example generation in Section 5.3, and the right-most one is for definition generation in Section 5.4. Each new example or definition is generated by greedy decoding. (Results of top-p sampled generations are shown in Table 9 in Appendix E.) The percentage of ties is the remaining after subtracting the win percentages of the two models. GPT-4o more frequently chooses the Minnow model as the winner compared to the corresponding baseline in all settings except for the instruction-tuned variant on Chimera.

Results For the quantitative evaluation, Table 1 shows the percentages of wins of each of the baseline and the Minnow model on both the BabyLM-10M test portion and Chimera. Across all settings, the Minnow model wins more often than the corresponding baseline except for the instruction-tuned variant on Chimera, demonstrating the improvement brought by Minnow. For the qualitative evaluation, Table 2 shows a word picked from the BabyLM-10M test portion along with its study and generated examples. See Appendix F for additional examples from the BabyLM-10M test portion (Tables 10 and 12) and Chimera (Table 11) and detailed analysis of both the baseline and the Minnow model's generations. A manual analysis of these generated examples reveals that the Minnow model more often generates syntactically correct and semantically plausible new usage examples compared to the baseline, confirming that Minnow finetuning improves the ability to understand and use a new word. Nevertheless, in several cases, the Minnow model still shows obvious syntactic and factual errors and merely rewords the study examples.

5.4 Definition Generation

To further probe how well Minnow finetuning helps the model understand a new word, we prompt each model to generate a definition for the word given one

Study Example Sentences	Minnow Generated Examples	Word
• the first blacksmiths were [new-token]. • many civilisations were in the area that is now turkey, like the [new-token], the roman empire and the byzantine empire. • spread of hepatoscopy and astrology to [new-token], etruscans, greeks and romans and to china • the first major empire in the area was the [new-token] (from the 18th century to the 13th century bce).	1. the [new-token] were a people who lived in the area of turkey. 2. perhaps the most famous and widely used alchemical symbol, first popularized by [new-token] alchemists, is the ouroboros.	*hittites*

Table 2: New examples generated for a word from the BabyLM-10M test portion by the Minnow model. The first one is generated by greedy decoding, and the second one by sampling with top-p=0.92. The Minnow model learns that *hittites* is an ancient ethnic group. However, the greedy-decoded example copies the information (turkey) from the study example, while the sampled example makes seemingly plausible but factually incorrect generalizations (the earliest known ouroboros is found in ancient Egyptian text.)

or a few usage examples. We again use the methodology of Teehan et al. (2024) for definition generation and evaluation, as well as the two evaluation datasets they used: CoLLEGe-DefGen, which they created, and the Oxford dataset (Gadetsky et al., 2018). CoLLEGe-DefGen was constructed by selecting 954 words from WordNet (Miller, 1995) and prompting GPT-4 (OpenAI, 2023) to generate one definition and five usage examples for each word. The model generates a definition from one, two, or three usage examples sampled for each word in this dataset (i.e., in 1-, 2-, or 3-shot settings). The Oxford test set consists of 12,232 words, each with a definition and a usage example collected from the Oxford Dictionary. The model generates a definition from the only usage example for each word in this dataset (i.e., in a 1-shot setting). To generate a definition, we prompt the model with the sequence of the usage example(s) (as in Section 5.3) followed by "The word [new-token] in the above sentence(s) is defined as "".[5] ([new-token] is instead the placeholder token or pseudoword, as appropriate). For additional comparisons with models expected to do especially well on this task, we also evaluate specialized definition-generation models: Giulianelli et al.'s (2023) FLAN-T5 models (Chung et al., 2024). See Appendix G for details of data preprocessing and the specialized models.

For the quantitative evaluation, we perform two types of comparison. The first type compares the model-generated and ground-truth definitions for each word by computing BERTScore F1 (Zhang et al., 2020) and ROUGE-L (Lin, 2004). The second type compares a pair of definitions generated from Llama-3 8B baseline and a Minnow model finetuned from it. Similarly to what we did in Section 5.3, we ask GPT-4o a question (without usage examples): "Which of the following is a better definition for the word '*Word*', or they tie?" where *Word* is the ground-truth word form, followed by three shuffled options including the two generated definitions and one "Tie" (see Appendix E for detailed prompting settings).[6] For the qualitative evaluation, we manually inspect 1-shot generated definitions for words from each dataset (presented in Table 4 and Tables 15 and 16 in Appendix G).

Results For the quantitative evaluation, we first present the 1-shot scores of comparing the model-generated and ground-truth definitions for Llama-3 8B baselines and the Minnow models in Table 3. In Appendix G, we present 1-shot scores for all models (including the specialized FLAN-T5 models and the original CoLLEGe model) in Table 13 and averaged 1-, 2-, and 3-shot results on CoLLEGe-DefGen in Table 14. In all of these settings, Minnow finetuning improves the baseline scores by 0.3–1.5 on BERTScore F1 and 3.1–5.3 on ROUGE-L. On CoLLEGe-DefGen, the Minnow model finetuned from the instruction-tuned Llama-3 8B outperforms all other models across all settings. On Oxford, the Minnow models finetuned from both variants perform comparably well, but they are inferior to the largest specialized FLAN-T5 by 2.9 on ROUGE-L. However, note that our Minnow finetuning is neither tailored for generating definitions nor using these specific definition datasets. In Table 1, when comparing the definitions generated from each baseline and a Minnow model finetuned from that baseline, the latter is more often favored over the corresponding baselines for both Llama-3 8B variants.

For the qualitative evaluation, Table 4 shows Minnow-model-generated and ground-truth definitions for words from CoLLEGe-DefGen (see Tables 15 and 16 in Appendix G for additional examples from CoLLEGe-DefGen and Oxford). To summarize our manual analysis, we find that definitions generated by the Minnow model often capture most of the word meanings, form reasonable inferences from the contexts, and outperform the baseline. However, they are not always precise compared to the ground-truth definitions.

6 Conclusion

In this work, we present Minnow, a new method to improve language models' capability to learn a new word from a few in-context usage examples. Minnow successfully induced this ability in models trained from scratch with human-scale linguistic data, as indicated by their performances in differentiating new words (Section 4). Minnow finetuning further improved the word learning

[5]The prompt ends with a double quotation mark ("), so that the model will continue with a definition ending at another double quotation mark. This makes extracting definitions easy.

[6]We only perform this comparison on the CoLLEGe-DefGen dataset due to the large scale of the Oxford dataset.

Model Variant	Method	CoLLEGe-DefGen BERTScore F1	CoLLEGe-DefGen ROUGE-L	Oxford BERTScore F1	Oxford ROUGE-L
pre-trained	baseline	85.1	14.9	83.2	11.0
	+Minnow	85.4	18.7	**84.7**	16.3
instruction-tuned	baseline	85.3	17.6	83.6	12.5
	+Minnow	**85.8**	**20.7**	**84.7**	**16.5**

Table 3: Quantitative evaluation of generated definitions by comparing them with ground-truth definitions. See Table 13 in Appendix G for results from all models. We generate a definition from only one example (1-shot). All definitions are generated with greedy decoding. Scores of Minnow models are averaged across three runs. Finetuning with Minnow improves the baseline models on both datasets and both metrics, and the Minnow model finetuned from the instruction-tuned variant of Llama-3 8B performs the best.

Example Sentence	Minnow Definition	True Definition	Word
After his thorough inspection of the antique pocket watch, the bespectacled collector sighed, claiming it was a `[new-token]`, much to the seller's disappointment.	a thing that is not genuine or authentic	a deception or trick	*swiz*
Despite his greed, the businessman felt bound by a `[new-token]` to maintain ethical practices.	a promise or agreement to do something	a moral obligation or command that is unconditionally and universally binding	*categorical imperative*

Table 4: Definitions for two words from CoLLEGe-DefGen generated by the Minnow model finetuned from instruction-tuned Llama-3 8B with greedy decoding. Each definition is generated using the single example sentence shown and provided in context. The generated definitions managed to infer the core semantic features from the examples, though they are not precise enough compared to the true definitions. In this first example, the Minnow definition for the word "*swiz*" captures the word's core meaning of fakeness, which is a reasonable inference from the example, but misses the intentional aspect, a nuance of the true definition. In the second example, the Minnow definition for "*categorical imperative*" captures the core meaning of obligation, which is a reasonable contrast to the businessman's greed, but misses the "unconditionally and universally binding" aspect in the true definition.

performance of a pre-trained LLM (Llama-3 8B), as demonstrated in their improvements in differentiating new words (Section 5.1 and 5.2) as well as in generating new usage examples (Section 5.3) and definitions (Section 5.4) for the learned new words. In summary, this word-learning capability enables models to systematically and flexibly understand and use a new word in novel contexts, and can be immediately transferred to other words and tasks without additional training.

The efficacy of Minnow, or meta-learning in general, suggests that human-level efficiency in linguistic generalizations may be acquired through practicing over many instances of learning tasks, without presuming strict, explicit inductive biases (Irie and Lake, 2024). Whether models achieve the generalizations in this work through human-like mechanisms, such as systematicity and categorical abstraction, remains for future analysis.

7 Limitations

Learning Settings In this work, we consider word learning only in the text modality, in which the language model learns the meaning from the distribution of words. However, many words have real-world references, which usually accompany human word learning. We also use aggregated data from multiple sources, not from single-human/child input. Thus, a multimodal, grounded setting of word learning using a single agent's input would be more realistic.

In addition, we only consider learning a single new word on the fly. However, in real-world learning, both humans and models need to continually learn multiple words, usages, and even abstract rules (Mueller et al., 2024). Implementing this continual learning setting would be another future direction.

Novelty of New Words When Testing LLMs When testing LLMs (Section 5), the words and example sentences we use may already exist in the pre-training data, potentially allowing LLMs to recall known word meanings rather than learn genuinely new ones (note, however, the Chimera dataset introduces new concepts which are unusual and not lexicalized). The performance of the baseline LLMs shows that, even with this potential worry, there is room for improvement, which the Minnow-finetuned LLMs are able to achieve.

Models trained from scratch with Minnow do not have this limitation. Their training data explicitly excludes held-out test words (Section 4). Therefore, their test performance reflects their genuine ability to learn novel words, and this ability can be developed by Minnow.

References

Suraj Anand, Michael A Lepori, Jack Merullo, and Ellie Pavlick. 2024. Dual process learning: Controlling use of in-context vs. in-weights strategies with weight forgetting. *arXiv preprint arXiv:2406.00053*.

Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Jason Phang, Shivanshu Purohit, Hailey Schoelkopf, Dashiell Stander, Tri Songz, Curt Tigges, Benjamin Thérien, Phil Wang, and Samuel Weinbach. 2023. GPT-NeoX: Large scale autoregressive language modeling in pytorch.

Khuyagbaatar Batsuren, Ekaterina Vylomova, Verna Dankers, Tsetsuukhei Delgerbaatar, Omri Uzan, Yuval Pinter, and Gábor Bella. 2024. Evaluating subword tokenization: Alien subword composition and oov generalization challenge. *arXiv preprint arXiv:2404.13292*.

Jean Berko. 1958. The child's learning of english morphology. *WORD*, 14:150–177.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

P Bloom. 2000. *How Children Learn the Meanings of Words*. MIT Press, Cambridge, MA.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Susan Carey and Elsa Bartlett. 1978. Acquiring a single new word. *Papers and Reports on Child Language Development*, 15:17–29.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Michael C. Frank. 2023. Bridging the data gap between children and large language models. *Trends in cognitive sciences*.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271, Melbourne, Australia. Association for Computational Linguistics.

Mario Giulianelli, Iris Luden, Raquel Fernandez, and Andrey Kutuzov. 2023. Interpretable word sense representations via definition generation: The case of semantic change analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3130–3148, Toronto, Canada. Association for Computational Linguistics.

Harold Stanley Heaps. 1978. *Information retrieval: computational and theoretical aspects*. Academic Press, Inc.

Aurélie Herbelot and Marco Baroni. 2017. High-risk learning: acquiring new word vectors from tiny data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 304–309, Copenhagen, Denmark. Association for Computational Linguistics.

John Hewitt. 2021. Initializing new word embeddings for pretrained language models.

Ziniu Hu, Ting Chen, Kai-Wei Chang, and Yizhou Sun. 2019. Few-shot representation learning for out-of-vocabulary words. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4102–4112, Florence, Italy. Association for Computational Linguistics.

Qian Huang, Eric Zelikman, Sarah Chen, Yuhuai Wu, Gregory Valiant, and Percy S Liang. 2024. Lexinvariant language models. *Advances in Neural Information Processing Systems*, 36.

Kazuki Irie and Brenden M. Lake. 2024. Neural networks that overcome classic challenges through practice. *ArXiv*, abs/2410.10596.

Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Melbourne, Australia. Association for Computational Linguistics.

Najoung Kim, Tal Linzen, and Paul Smolensky. 2022. Uncontrolled lexical exposure leads to overestimation of compositional generalization in pretrained models. *arXiv preprint arXiv:2212.10769*.

Najoung Kim and Paul Smolensky. 2021. Testing for grammatical category abstraction in neural language models. In *Proceedings of the Society for Computation in Linguistics 2021*, pages 467–470, Online. Association for Computational Linguistics.

Brenden M. Lake and Marco Baroni. 2023. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623:115 – 121.

Andrew K Lampinen and James L McClelland. 2017. One-shot and few-shot learning of word embeddings. *arXiv preprint arXiv:1710.10280*.

Sander Land and Max Bartolo. 2024. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11631–11646, Miami, Florida, USA. Association for Computational Linguistics.

Angeliki Lazaridou, Marco Marelli, and Marco Baroni. 2017. Multimodal word meaning induction from minimal exposure to natural text. *Cognitive science*, 41 Suppl 4:677–705.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Llama Team, Meta AI. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR 2019*.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria. Association for Computational Linguistics.

Brian MacWhinney. 1992. The CHILDES project: tools for analyzing talk. *Child Language Teaching and Therapy*, 8:217 – 218.

Sabrina J Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y Lee, Benoît Sagot, et al. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *arXiv preprint arXiv:2112.10508*.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

George A. Miller. 1995. WordNet: A lexical database for english. *Commun. ACM*, 38:39–41.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Aaron Mueller, Albert Webson, Jackson Petty, and Tal Linzen. 2024. In-context learning generalizes, but not always robustly: The case of syntax. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4761–4779, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*.

Core Francisco Park, Andrew Lee, Ekdeep Singh Lubana, Yongyi Yang, Maya Okawa, Kento Nishi, Martin Wattenberg, and Hidenori Tanaka. 2024. ICLR: In-context learning of representations. *arXiv preprint arXiv:2501.00070*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jacob Russin, Sam Whitman McGrath, Danielle J. Williams, and Lotem Elber-Dorozko. 2024. From Frege to chatGPT: Compositionality in language, cognition, and deep neural networks. *ArXiv*, abs/2405.15164.

Timo Schick and Hinrich Schütze. 2019. Learning semantic representations for novel words: Leveraging both form and context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6965–6973.

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *AAAI Conference on Artificial Intelligence*.

Jingyuan Sun, Shaonan Wang, and Chengqing Zong. 2018. Memory, show the way: Memory based few shot word representation learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1435–1444, Brussels, Belgium. Association for Computational Linguistics.

Ryan Teehan, Brenden Lake, and Mengye Ren. 2024. CoLLEGe: Concept embedding generation for large language models. In *First Conference on Language Modeling*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melissa Hall Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Jason Wei, Dan Garrette, Tal Linzen, and Ellie Pavlick. 2021. Frequency effects on syntactic rule learning in transformers. In *Proceedings of the 2021 Conference on Empirical*

Methods in Natural Language Processing, pages 932–948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. In *North American Chapter of the Association for Computational Linguistics*.

Aditya Yedetore, Tal Linzen, Robert Frank, and R. Thomas McCoy. 2023. How poor is the stimulus? evaluating hierarchical generalization in neural networks trained on child-directed speech. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9370–9393, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *ICLR*.

George Kingsley Zipf. 1949. *Human behavior and the principle of least effort*. Addison-Wesley Press.

A Word Usage Dataset Creation

As we mentioned in Section 3.2, we construct one dataset from each of two corpora: CHILDES (MacWhinney, 1992) and BabyLM-10M (Warstadt et al., 2023). The CHILDES dataset is licensed for use under a CC BY-NC-SA 3.0 license.[7] Our scientific use is under the terms of the license.[8] We did not find the license of the BabyLM dataset, which aggregated multiple public datasets. Since there is plenty of published work using this public dataset, we believe our scientific use does not violate any terms or conditions. In the following, we describe how we preprocess these two corpora and create a word usage dataset from each corpus.

Preprocessing Since the basic units of our focus are words (as opposed to word pieces in other tokenization schemes), we need to identify words in the text. To achieve this, we apply the same word-level tokenization to all datasets (for consistency) and mark word boundaries by whitespace during preprocessing. Models trained from scratch use this word-level tokenization. When the text is used in finetuning Llama-3, which comes with its pre-trained subword tokenizer, we remove the unnatural spaces introduced by the word-level tokenization and tokenize the text again with Llama-3 tokenizer, so the text format becomes closer to its pre-training data (See the Finetuning paragraph in Appendix B for further details of this process). For CHILDES data, we preprocess the data in the same way as Yedetore et al. (2023) did, which uses children's input in the North American English portion, but we do not split and unk the data at the preprocessing stage. For BabyLM data, we use the data in the 10M track of the BabyLM Challenge 2023, which mixes 10 portions, each from a different data source (child- or adult-oriented, speech transcription or written text like Wikipedia). We exclude the QED portion for its poor quality (also mentioned in the 2nd BabyLM Challenge). We apply word-level tokenization on untokenized portions, and then split the text into sentences using heuristics. We use spaCy for all word-level tokenization along with Part-Of-Speech tagging. We lowercase all text before preprocessing to unify the capitalization of words in different places. We deduplicate sentences and remove sentences having less than 1 word (not counting punctuation).

Assigning sentences and splitting To create a dataset from a corpus, we first get the token frequencies of all words. (Here, a word means a word-form. We discuss its implications in Appendix H.) Then we select the set of words to be meta-learned. We will only consider nouns, verbs, adjectives, and adverbs to be meta-learned (a word's syntactic category is based on the word's most frequent Part-Of-Speech tag). We choose two thresholds for meta-learned words: the maximum frequency of a meta-learned word and the minimum number of examples per meta-learned word. We use a greedy algorithm to assign each sentence in the corpus to the example set of at most one potential meta-learned word that occurs in the sentence, so each meta-learned word has at least the minimum number of examples. This ensures that the model cannot infer the identity of the word masked by the placeholder token from other sentences. These words and their example sets constitute the meta-learning component of the dataset. We include the remaining sentences not assigned to any meta-learned word in the language-modeling component. Finally, we split both the meta-learning component (by word) and the language-modeling component (by sentence) into training (80%), validation (10%), and test (10%) portions.

When training models from scratch, we build the vocabulary from the words occurring with a minimum frequency in the training portion (same as the minimum number of examples per meta-learned word) while excluding all meta-learned words. This ensures that meta-learned words, like the lowest-frequency words, are out-of-vocabulary and will be replaced by <unk> tokens, so they will never be learned in-weights.

Statistics of our created datasets are shown in Table 5. Read our code for full details.

[7] https://talkbank.org/share/rules.html

[8] https://creativecommons.org/licenses/by-nc-sa/3.0/

		CHILDES			BabyLM-10M		
max. freq. of meta-learned words		200			15		
min. #uses of meta-learned words		5			5		
vocabulary size		2179			22,696		
portion		training	valid.	test	training	valid.	test
meta-learning	#meta-learned words	7790	973	975	15,821	1977	1979
	total #uses	201,957	26,449	26,234	108,466	13,552	13,563
	mean #uses	25.93	27.18	26.91	6.86	6.85	6.85
	total #tokens	1,899,159	245,509	243,387	2,072,560	260,701	257,933
	mean sentence length	9.40	9.28	9.28	19.11	19.24	19.02
	unk rate	3.32%	3.28%	3.28%	3.61%	3.78%	3.91%
language modeling	#sentences	508,630	63,578	63,580	521,911	65,238	65,240
	total #tokens	3,927,120	492,280	490,990	5,721,893	715,553	715,111
	mean sentence length	7.72	7.74	7.72	10.96	10.97	10.96
	unk rate	1.00%	1.03%	1.00%	1.44%	1.49%	1.47%
total #tokens		5,826,279	737,789	734,377	7,794,453	976,254	973,044

Table 5: Dataset statistics. All statistics are based on tokens, which mostly correspond to words except punctuations due to our word-level tokenization. "unk rate" is the percentage of out-of-vocabulary tokens, which are replaced by <unk>, in all tokens. Unk rate is slightly higher in the validation and test portions than the training portion because we build the vocabulary from the training portion. As shown by the mean sentence lengths, the meta-learning sentences are longer on average than the language modeling sentences, since meta-learned words are of lower frequency and thus are usually in more complex sentences. We manually tune the two thresholds of meta-learned words so we have enough number of meta-learned words while the unk rate is not too high.

B Model and Training Configurations

Training from scratch We slightly modify the configuration of Pythia-160M (Biderman et al., 2023), which uses the Transformer architecture GPT-NeoX (Andonian et al., 2023). The configuration has 12 layers and a hidden dimension size of 768. We change the vocabulary size according to the corresponding dataset, as shown in Table 5. We also include three special tokens in the vocabulary: the placeholder token `[new-token]`, the separator token `<sep>`, and `<unk>`, as mentioned in Section 4. We change the Pythia configuration to tie the input and output embeddings. This makes the model parameter counts smaller, 86.7M and 102.5M for the model trained on CHILDES and BabyLM-10M, respectively. For both models, we use batch size (i.e., number of episodes/sequences per batch) 8 and AdamW optimizer (Loshchilov and Hutter, 2019) with initial learning rate 3×10^{-4}, and reduce the learning rate by multiplying 0.1 when the validation loss has stopped improving for 2 epochs. We apply weight decay 0.07 and 0.15 when training on the CHILDES and BabyLM-10M datasets, respectively. Other configurations, such as no dropout, are kept the same as Pythia-160M. For each setting, we run 3 times with random seed $\{0, 1, 2\}$. Each run is performed on a single V100 GPU for 30 epochs (9–18 hours).

Finetuning We finetune Llama-3 8B (Llama Team, Meta AI, 2024) with Minnow on each of the CHILDES and BabyLM-10M datasets, but we refer to the models finetuned on BabyLM-10M by default, as we mentioned in Section 5. We finetune from both the pre-trained and instruction-tuned variants of Llama-3 8B, but we refer to the models finetuned from the pre-trained variant by default, presenting results of finetuning from the instruction-tuned variant only in the generative settings, where their performance may differ significantly due to their different capabilities to follow the prompt. We use two reserved special tokens in Llama-3 tokenizer vocabulary as the placeholder token and the separator token. To make the tokenization more natural to the model's pre-training data, we clean up tokenization spaces in the text (e.g., the space before ",", ".", or "'s") introduced by the word-level tokenization during preprocessing and make the placeholder token absorbs any preceding spaces of the word. Finetuning is minimally parameter-efficient: We finetune only the input and output embeddings of the two special tokens, while freezing all other parameters. Before finetuning, the input/output embedding of either token is initialized to the mean of all input/output embeddings (Hewitt, 2021). When finetuning the model on CHILDES with 5 examples per episode, we use batch size (i.e., number of episodes/sequences per batch) 32 and initial learning rate 3×10^{-3} and truncate the sequence to the max length of 80 tokens to control the memory usage. When finetuning the model on CHILDES with 10 examples per episode, we use batch size 8 and initial learning rate 3×10^{-4} and truncate the sequence to the max length of 180 tokens. When finetuning the model on BabyLM-10M with 5 examples per episode, we use batch size 16 and initial learning rate 1×10^{-3} and truncate the sequence to the max length of 160 tokens. Other settings are the same as when training from scratch except that we do not apply weight decay. Each run is performed on a single A100 GPU for 15 epochs on CHILDES (33 hours) or 12 epochs on BabyLM-10M (48 hours).

C Held-out Word Classification

As we mentioned in Section 3.3, we need different meta-learned words in the same group. Therefore, different from training, we sample only one episode of K examples per word from the validation/test portions so we do not repeat the same word in a classification group. We also fix the shuffle order so all models are evaluated on the same classification task instances. We experimented with training models with $K \in \{5, 10\}$ examples per episode on CHILDES and BabyLM-10M and evaluated each of them on the corresponding dataset with the same K and $C \in \{4, 8\}$. Training models with $K = 10$ examples per episode on BabyLM-10M was unsuccessful because the concatenated sequence was too long, exceeding the GPU memory, so we do not have results in this setting.

We are aware of the weaknesses of this task. Discriminating a new word from an arbitrary set of other new words is a relatively weak test of word meaning learning. The task could be easy simply because different words are used in very different contexts, so the conditional likelihood may reflect just the coherence of the usage contexts between study and query examples, not the meaning of the new word (we demonstrate this point by an additional baseline below where we present the model only the usage contexts without new words). In addition, results from the task do not tell us what features of word meanings the model is learning. Our syntactic category classification task addresses these concerns by focusing on the syntactic aspect and breaking the semantic coherence between study and query examples (Section 5.2).

Below, we describe two baselines we run on this task.

Baseline: Llama-3 8B learning a pseudo-word in context (Llama-3 8B with '*dax*') This is the baseline model introduced in Section 3.4. We follow the format described there and additionally prepend a prompt to make the performance better: "The following lines are lowercased example sentences using a new word '*dax*' in random order, one per line:". (We discuss the consequence of using a same pseudo-word in Appendix H.)

Additional Baseline: Llama-3 8B modeling the coherence of usage contexts (Llama-3 8B with '') This is the additional baseline to evaluate the effectiveness of utilizing just the coherence of the contexts, as we discussed above. We remove the new word from each example (equivalent to replacing the new word with an empty string), so only the usage context of each example is retained.

For these baselines, we also experimented with the instruction-tuned variant of Llama-3 8B but it performs worse on this task.

Table 6 shows all models' held-out word classification results on the test portions of CHILDES and BabyLM-10M datasets.

dataset	K	C	Minnow from scratch	Llama-3 8B with ‘’	Llama-3 8B with ‘*dax*’	Llama-3 8B +Minnow
CHILDES	5	4	72.3(1.6)	58.33	71.09	**79.1**(0.5)
		8	59.8(0.4)	46.49	60.02	**70.4**(0.2)
	10	4	75.1(0.7)	66.56	76.53	**84.9**(0.2)
		8	63.4(1.5)	56.17	66.05	**75.9**(0.6)
BabyLM-10M	5	4	77.4(0.5)	70.45	78.39	**86.5**(0.6)
		8	67.5(0.7)	60.12	69.74	**80.5**(1.0)

Table 6: Accuracy (%) of held-out word classification on the CHILDES and BabyLM-10M test sets. We show the mean and the standard deviation (in the bracket) of 3 runs. “Minnow from scratch” means models trained from scratch on the corresponding dataset. “Llama-3 8B with ‘’” means the baseline model without prompt and remove the new word (i.e., replace the new word with an empty string). “Llama-3 8B with ‘*dax*’” means the baseline model with prompt learning the new word ‘*dax*’. We use $K-1$ study examples in this classification task, and models except the baselines are trained/finetuned on K examples per training episode so they see the same number of examples during training and evaluation. C is the number of words in each group, so we will have $\lfloor \frac{n_{\text{episodes}}}{C} \rfloor$ groups. Note that we discard the last batch of less than C episodes, so the used numbers of episodes are slightly smaller. Results of “Llama-3 8B with ‘’” show that the coherence of the context already provides better-than-chance accuracy on this classification task. Results of “Llama-3 8B with ‘*dax*’” show that the pre-trained LLM already performs well. However, “Llama-3 8B +Minnow” outperforms the baselines by a large margin, showing the effectiveness of our method. Models finetuned with Minnow from the instruction-tuned variant of Llama-3 8B perform worse than or close to the pre-trained variant here (the instruction-tuned variant finetuned with Minnow has 86.3% (4-way) and 80.1% (8-way) mean classification accuracies; the instruction-tuned variant with ‘*dax*’ has 75.2% (4-way) and 66.0% (8-way) classification accuracies), so we do not include their results here.

D Syntactic Category Classification

As we mentioned in Section 5.2, we use the methodology of Kim and Smolensky (2021) and the dataset they constructed. The dataset was constructed from MNLI, a Natural Language Inference dataset (Williams et al., 2017). The task is to discriminate between a pair of words in two different syntactic categories. They consider 4 syntactic categories: noun, verb, adjective, and adverb. Therefore, they have 6 pairs of categories for discrimination. For each category pair, the dataset contains two signal contexts (one for each category; we use them as the study examples) and 200 test sentences using a word unambiguously in either category (100 for each category; we use them as the query examples). The main difference between our approach and that of Kim and Smolensky (2021) is that, instead of finetuning a new word embedding on each signal context, we apply in-context learning, using each signal context as an in-context study example of the new word. Read Kim and Smolensky (2021) for further details.

Results from models trained from scratch, Llama-3 8B baseline and models finetuned from Llama-3 8B on the 6 category pairs and their mean are visualized in Figure 2. Table 7 shows detailed results from Llama-3 8B baseline and Llama-3 8B finetuned with Minnow on BabyLM-10M. Table 8 shows detailed results from models trained from scratch on both datasets.

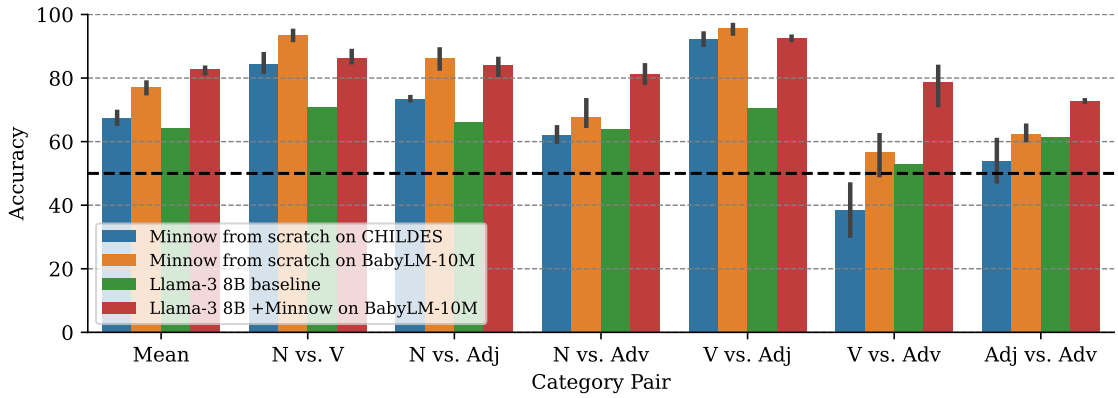

Figure 2: Syntactic classification accuracy. Error bar shows the 95% confidence interval given 3 runs. "Minnow from scratch on CHILDES" (blue) and "Minnow from scratch on BabyLM-10M" (orange) mean the models trained from scratch with Minnow on CHILDES and BabyLM-10M, respectively. (These models have a closed vocabulary, so many words in the dataset will be Out-Of-Vocabulary and be presented as <unk>, which could make the task easier.) "Llama-3 8B baseline" (green) means Llama-3 8B baseline with pseudo-word "*dax*". "Llama-3 8B +Minnow on BabyLM-10M" (red) means Llama-3 8B finetuned with Minnow on BabyLM-10M. "N", "V", "Adj", and "Adv" are short for noun, verb, adjective, and adverb, respectively. "Mean" is the mean across all category pairs. The black dashed line marks the chance level (50%). "Llama-3 8B +Minnow on BabyLM-10M" (red) shows improvement over "Llama-3 8B baseline" (green) in all category pairs, with mean accuracy risen from 64% to 83%. Note that "Minnow from scratch on BabyLM-10M" (orange) has a 77% mean accuracy, much better than the baseline accuracy and even comparable to the Minnow models finetuned from Llama-3 8B on many category pairs, again demonstrating its data efficiency.

		Llama-3 8B baseline			**Llama-3 8B +Minnow**		
Cat. 1	**Cat. 2**	**Acc.**	**Acc. (1>2)**	**Acc. (2>1)**	**Acc.**	**Acc. (1>2)**	**Acc. (2>1)**
Noun	Verb	71.0	43	99	**86.3**(1.5)	**74.7**(1.7)	98.0(1.6)
Noun	Adjective	66.0	**79**	53	**84.0**(2.2)	71.3(4.6)	**96.7**(0.5)
Noun	Adverb	64.0	55	73	**81.3**(2.2)	**75.7**(1.7)	**87.0**(2.9)
Verb	Adjective	70.5	49	92	**92.7**(0.5)	**90.0**(2.2)	**95.3**(1.2)
Verb	Adverb	53.0	85	21	**78.8**(5.2)	**90.0**(2.4)	**67.7**(12.5)
Adjective	Adverb	61.5	42	81	**72.8**(0.2)	**57.3**(2.6)	**88.3**(3.1)

Table 7: LLMs' accuracies (%) of distinguishing two syntactic categories in novel contexts. We show the mean and the standard deviation (in the bracket) of 3 runs. 'Acc. (1>2)' denotes the accuracy on the set of sentences where Category 1 should be preferred over Category 2 (e.g., assigning a higher probability to a noun in a noun-expecting context for row 1), and vice versa. Column 'Acc.' lists the aggregate accuracy. "Llama-3 8B +Minnow" have accuracies significantly better than chance except distinguishing adjective from verb (row 5). Additionally, "Llama-3 8B +Minnow" improves over Llama-3 8B baseline in differentiating most category pairs except discriminating nouns from adjectives (row 2), showing the effectiveness of finetuning with Minnow.

		Minnow from scratch on CHILDES			**Minnow from scratch on BabyLM-10M**		
Cat. 1	**Cat. 2**	**Acc.**	**Acc. (1>2)**	**Acc. (2>1)**	**Acc.**	**Acc. (1>2)**	**Acc. (2>1)**
Noun	Verb	84.5(2.3)	79.7(3.7)	89.3(4.5)	93.5(1.8)	90.0(2.2)	97.0(1.4)
Noun	Adjective	73.5(0.4)	50.7(2.9)	96.3(2.1)	86.2(2.5)	79.7(5.4)	92.7(1.9)
Noun	Adverb	62.2(1.8)	90.3(4.1)	34.0(6.4)	67.8(3.7)	86.3(3.1)	49.3(5.8)
Verb	Adjective	92.3(1.4)	90.0(2.8)	94.7(1.2)	95.7(1.2)	93.0(2.4)	98.3(0.5)
Verb	Adverb	38.5(6.5)	57.3(14.7)	19.7(1.7)	56.7(5.3)	68.7(5.8)	44.7(11.4)
Adjective	Adverb	53.8(5.3)	44.0(5.4)	63.7(10.1)	62.3(1.9)	59.0(6.5)	65.7(4.1)

Table 8: Accuracies (%) of distinguishing two syntactic categories in novel contexts for models trained from scratch with Minnow. We show the mean and the standard deviation (in the bracket) of 3 runs. 'Acc. (1>2)' denotes the accuracy on the set of sentences where Category 1 should be preferred over Category 2 (e.g., assigning higher probability to a noun in a noun-expecting context for row 1), and vice versa. Column 'Acc.' lists the aggregate accuracy. Both models perform better than chance on many category pairs, suggesting that models can develop some ability to one-shot learn the syntactic category of a word from human-scale data with Minnow.

E Comparing Generations

For new usage example generation (Section 5.3), we show GPT-4o the following text format:

> The following lines are shuffled lowercased example sentences using a new word 'dax', one per line:
>
> * EXAMPLE-1
>
> * EXAMPLE-2
>
> * EXAMPLE-3
>
> * EXAMPLE-4
>
> Please answer in a single uppercase letter: Which of the following is a better next example for the word 'dax', or they tie?
>
> A) OPTION-A
>
> B) OPTION-B
>
> C) OPTION-C

where OPTION-A, OPTION-B, OPTION-C are shuffled generation-1, generation-2, and "Tie".

For definition generation (Section 5.4), we do not have the examples (and the prompt before them) and instead have the direct prompt before the options: "Please answer in a single uppercase letter: Which of the following is a better definition for the word '*Word*', or they tie?" where *Word* is the ground-truth word form.

We always get the first letter (A, B, or C) of the GPT-4o response as the choice.

Tables 1 and 9 show the results of comparing Llama-3 8B baseline (pre-trained to instruction-tuned) to the Minnow model finetuned from that baseline (with random seed 0) on new examples and definitions generated by greedy decoding and top-p=0.92, respectively.

Variant	Method	New Example BabyLM-10M test	New Example Chimera	Definition CoLLEGe-DefGen
Pre-trained	baseline	39	39	25
	+Minnow	**53**	**42**	**31**
Instruction-tuned	baseline	46	**52**	**33**
	+Minnow	**47**	36	28

Table 9: Percentages of wins of each model when comparing the generations from Llama-3 8B baseline (pre-trained to instruction-tuned) with the Minnow model finetuned from that baseline, judged by GPT-4o. The left two datasets are for new usage example generation in Section 5.3, and the right-most one is for definition generation in Section 5.4. Each new example or definition is generated by top-p=0.92. The percentage of ties is the remaining after subtracting the win percentages of the two models. GPT-4o more frequently chooses the Minnow model as the winner compared to the corresponding baseline, except for the instruction-tuned model on Chimera and CoLLEGe-DefGen.

F Evaluation of Generated New Usage Examples

As we mentioned in Section 5.3, we sample study examples from two datasets: the BabyLM-10M test portion and the Chimera dataset (Lazaridou et al., 2017). Statistics of the BabyLM-10M test portion is in Table 5, Appendix A. The Chimera dataset contains 33 chimeras. For each chimera, we merge all examples in its 10 "passages" (each with a different pseudo-word in place of the word) in the original dataset, so we only have 33 different words to learn. We do not use the pseudo-words in the original dataset.

Tables 10 and 11 show samples of example sentences from the BabyLM-10M test portion and the Chimera dataset along with the new usage examples generated by both the Llama-3 8B baseline and the Minnow model finetuned from that baseline. Table 12 contains error analysis of new examples generated from the BabyLM-10M test portion, extending Table 2 in Section 5.3.

Study Example Sentences	Minnow	Baseline	Judgment and Reason	Word
• tales similar to " bluebeard " include " the fitcher's bird " by the brothers [new-token] and the english tale " mr. fox ". • these types of tales include royalty and fantastical magic, many of which are published by brothers [new-token] and hans christian anderson. • previously on [new-token]... • the house in the wood from the german of [new-token].	previously in [new-token]	the story is actually entitled " the bridal of trier " as mentioned on brothers [new-token] site.	Baseline wins. Minnow finetuned output simply changes the preposition of an example, which is likely incorrect. The baseline output copies "brothers [new-token]", which is likely correct.	grimm
• men were sent to search the [new-token], and there, on the very night before the deed was to be done, guy fawkes was found waiting till the time should come to set fire to the powder. • the actual [new-token] they've got some of the old er writings that the, the monks and you're not allowed to take photographs. • searched all the tunnels and [new-token]. • he puts them in [new-token], anderson shelters, confined spaces, things like that.	because they can light it in a [new-token] and blow it with the minimum amount of people.	when a young teacher of history at the university of the commonwealth in london, [new-token] began the revolutionary research into indian philosophy of patanjali that was later to prove so fruitful.	Minnow model wins. The word likely refers to a place or space. Only the Minnow finetuned output uses the word as a place.	cellars
• the sorcerer wotan is using the [new-token] of aten to blot out the sun. • unless it involves some kind of enchanted [new-token], • i'll give her the lucky [new-token] you gave m • this can be a long ritual, or it can take the form of simply wearing an [new-token] or other token, " against bad luck ".	first, she goes with colm and feels a little guilty for keeping his [new-token].	the narrative treatment of [new-token] in the media exemplifies this process.	Minnow model wins. The word refers to a ritual token. The Minnow finetuned output suggests it is an object, which could agree with the inference, while the baseline output does not convey a lot of information.	amulet
• they are blind, their eyes having become reduced to [new-token] lenses under the skin, and they have no external ears, just a pair of tiny holes hidden under thick hair. • it is a [new-token] reflex, a remnant of the past when pre - human babies clung to their mothers ' hair as the mothers swung through the trees. • however, [new-token] structures may have their original function replaced with another. • strong evidence for common descent comes from [new-token] structures.p397 the useless wings of flightless beetles are sealed under fused wing covers.	the [new-token] structures of the human body are the appendix, the coccyx, the earlobes, the tailbone, and the wisdom teeth.	the [new-token] structures of the human body are the same as those of other mammals.	Tie. The word should be an adjective meaning an inherited biological feature that lost its original function. Both generations fit this interpretation. GPT-4o decides the Minnow model as the winner.	vestigial
• upon the very margin of the ditch, not thirty feet from where they crouched, an iron caldron [new-token] and steamed above a glowing fire ; • all round the hull, in the blackness, the rippling current [new-token] and chattered like a little mountain stream. • there hung another kettle that [new-token] and boiled ; but there was no fire beneath it, either. • and when he lifted it up and looked down, there stood a great copper cauldron which [new-token] up and boiled away down there ;	the boy looked down into the pot and saw that it [new-token] and boiled away.	he turned his back upon them, and went in at the door in the [new-token], which was a pleasant door to look upon.	Minnow model wins. The word should be a past-tense verb meaning a physical movement of an object, likely caused by steam or current. The Minnow finetuned output is consistent with this meaning, but the baseline incorrectly uses the word as a noun.	bubbled

Table 10: New examples generated on study examples from the BabyLM-10M test portion by the Minnow model and the Llama-3 8B baseline it is finetuned from, with greedy decoding. In addition to GPT-4o's judgment, the first author manually compared each pair of generations and gave the judgment and reason (Judgment and Reason). When the author and GPT-4o compare the generations, they cannot see the ground-truth word but have to infer the possible meaning of the word. When GPT-4o's judgment disagrees with the author's, it is mentioned at the end of Judgment and Reason.

Study Example Sentences	Minnow	Baseline	Judgment and Reason	Word
• Animals such as capybara, jaguars, jacare ([new-token]s) and hyacinth macaws are particularly vulnerable. • Nadirpur stared at it as though it were a [new-token], his face quite drained. • The owner will be summonsed to court for illegal possession of a [new-token] without a permit. • I went to the cavern to get some, and found a [new-token] in there.	• The [new-token] was a little more than a metre long, and had a thick, black coat.	• The [new-token] was a large, black, furry creature with a long, pointed snout.	Minnow model wins. An alligator (inferred from the first example) is not furry.	alligator + rattlesnake
• He is principal [new-token] of the English Baroque soloists, with whom he has recorded for Erato and Deutsche Grammophon. • Banners point to the military aspect of [new-token] music: a Jacobite flag and the first Union flag. • " Alexander Baillie said that he was impressed by her playing and even tried out her [new-token]! • Where did the [new-token] idea at the end of Rubina's Blue Sky Happiness come from?	• The first movement is a [new-token], the second a minuet, and the third a rondo.	• The [new-token] is a small, furry, four-legged creature with a long tail.	Tie. Both generations are not for an instrument.	cello + bagpipe
• 4. Allow the stock to settle then skim off the fat using [new-token] or fat separator. • Add the pasta, cook it for about 12 minutes until just soft, then drain it in a [new-token]. • Imagine you've got a bucketful of praise in every room of the house with a big [new-token]. • Plants can be grown in tyres, old metal pots such as buckets, watering cans or even a [new-token]!	• 2. Put the chicken in a [new-token] and cover with cold water.	• The [new-token] is a very useful tool for the gardener.	Minnow model wins. The word refers to a kitchen utensil, likely a container. The baseline may be misled by the last example but do not notice the word "even" indicating the surprise of the usage.	ladle + colander
• At about the same age I was in a cafe and I spilt some pop on my [new-token]s. • It's also the first International home [new-token] in history that is reversible as well. • Such as how if you have sufficiently-stretchy pants you could take them off without first removing your [new-token]s. • It was in a cinema on 86th Street where he sat in the balcony, wearing an old sports jacket and open-necked [new-token].	• The [new-token] is a bit tight.	• I was in a cafe and I spilt some pop on my [new-token]s.	Tie. A word for clothes can be used in both example. GPT-4o decides the baseline as the winner.	trouser + shirt

Table 11: New examples generated on study examples from the Chimera dataset (Lazaridou et al., 2017) by the Minnow model and the Llama-3 8B baseline it is finetuned from, with greedy decoding. Like in Table 10, the first author provides the judgment and reason (Judgment and Reason). The first example is a reasonable guess of the appearance of an animal. The second example is related to music, but incorrectly infers the word to be a dance or musical form. The last example correctly infers that the word is a kitchen utensil.

Study Example Sentences	Minnow	Baseline	Word	Analysis
• the first blacksmiths were [new-token]. • many civilisations were in the area that is now turkey, like the [new-token], the roman empire and the byzantine empire. • spread of hepatoscopy and astrology to [new-token], etruscans, greeks and romans and to china • the first major empire in the area was the [new-token] (from the 18th century to the 13th century bce).	1. the [new-token] were a people who lived in the area of turkey. 2. perhaps the most famous and widely used alchemical symbol, first popularized by [new-token] alchemists, is the ouroboros.	1. the [new-token] was a major power in the region, and its influence spread to the surrounding areas. 2. epistemology will suffer the [new-token] syndrome.	hittites	The Minnow model learns that *hitties* is an ancient ethnic group. However, the greedy-decoded example copies the information (turkey) from the study example, while the sampled example makes seemingly plausible but factually incorrect generalizations (the earliest known ouroboros is found in ancient Egyptian text.) The baseline also generates factual errors (Hittite plague exists but no "hitties syndrome").
• but, i know from his childhood, he says that his father always buys and [new-token] cars just as entertainment, i guess. • in the end, yuji accepts shana's feelings and the two kiss, which activates a spell left to yuji by crimson lord lamia that [new-token] his existence, so yuji is no longer a torch. • – a medicine which [new-token] health and energy, as extract of malt, cod liver oil, etc. • that diversifies energy supply, [new-token] ecosystems, protects biodiversity, and sequesters carbon.	1. the [new-token] of the [new-token] of the [new-token] of **[Infinite Repetition]** 2. he is a former prime minister, who is then [new-token] of his duties.	1. the [new-token] is a 3.5 mm jack, which is the standard for most headphones. 2. he [new-token] up to an incredible 2,620 sales per store in 2009, compared with the 60 to 80 that had been typical just a few years earlier.	restores	For the Minnow model, the greedy-decoded example is nonsense repetition (a common failure of LMs), while the sampled example uses the verb with incorrect syntax ("be restored to") but pertinent semantics. For the baseline, the greedy-decoded example also has incorrect syntax (use the word as a noun), and the use in the sampled example is semantically incorrect (the sales must have reached a similar level in the past).
• when i was a student [new-token], i went to the prado in Madrid. • [new-token] is hiking in the outdoors with a backpack full of all your food, clothing, water, and shelter. • [new-token] may be done on short or long trips. • [new-token] (wilderness)	1. [new-token] is a form of wilderness travel. 2. a small [new-token] campsite is usually composed of a place to sleep, food, tools and water.	1. [new-token] is a type of backpacking. 2. [new-token] is something all people can do.	backpacking	The Minnow model seems to infer the meaning of the word, but is rewording the study examples. The baseline also seems to infer the meaning, but its sampled example is not very informative.

Table 12: Error analysis of new examples generated from the BabyLM-10M test portion by the Minnow model and the baseline. In each column of generated examples, the first one is generated by greedy decoding, and the second one by sampling with top-p=0.92.

G Evaluation of Generated Definitions

As we mentioned in Section 5.4, we use two definition generation datasets: CoLLEGe-DefGen (Teehan et al., 2024) and the Oxford test set (Gadetsky et al., 2018). The original datasets contain 954 and 12,232 words, from which we removed 4 and 2 duplicated words, respectively. For CoLLEGe-DefGen, we keep the inflectional suffixes, such as "-s", "-ed", and "-ly", after the placeholder so that the placeholder only corresponds to the word stem. This is to remove the influence of morphological inflections. Note that we use our placeholders instead of the <nonce> in the original text of CoLLEGe-DefGen. In addition, we fixed several incorrect word/phrase replacements in the original dataset (for example, the phrase "*capital gains tax*"). For the Oxford dataset, for simplicity and consistency with previous work, we do not keep the inflectional suffixes but rather replace the whole word with the placeholder. There are 12% examples in the Oxford test set in which we find no occurrences of any form of the word to be learned, but we keep them for consistency with previous work.

Additionally, as we also mentioned in Section 5.4, we have additional references of what can be achieved by specialized definition-generation models: the series of FLAN-T5 (Chung et al., 2024) models finetuned by Giulianelli et al. (2023) specifically on generating definitions. This also follows what Teehan et al. (2024) did. These models were finetuned on three corpora, including the Oxford training set (Gadetsky et al., 2018). The series of finetuned FLAN-T5 are listed on their GitHub page (https://github.com/ltgoslo/definition_modeling?tab=readme-ov-file#definition-generation-models-for-english) and can be accessed through Hugging Face model hub. When evaluating the FLAN-T5 models, a pseudo-word '*wug*' is used as the placeholder for the new word, like in other baselines (Section 3.4) for a fair comparison. Each FLAN-T5 model is prompted with an example sentence followed by a question, "What is the definition of wug?", as what Giulianelli et al. (2023) did.

Table 13 shows the results of comparing the model-generated and ground-truth definitions from all models, supplementing the brief results in Table 3 with those from the additional specialized FLAN-T5 baselines and the CoLLEGe model. Table 14 shows the average of 1-, 2-, and 3-shot results on the CoLLEGe-DefGen dataset. Tables 15 and 16 show additional definitions generated from the CoLLEGe-DefGen and Oxford test set by the baselines and the Minnow models (in addition to Table 4 in Section 5.4).

Results of CoLLEGe (Teehan et al., 2024), which generates new embeddings to be used in an LLM, are appended to Table 13 and 14 and are directly copied from the original paper. Those numbers should be compared with other models with caution because they have different settings: They are based on Llama-2 7B (Touvron et al., 2023) and new embeddings, their data processing is not as fine as ours (they did not remove duplicated words from both datasets, did not keep the inflectional suffixes in CoLLEGe-DefGen, and did not find more forms in the Oxford dataset as we do (15% examples without replacing any word form)), and the usage examples they randomly selected from CoLLEGe-DefGen are different from ours.

Model Variant	Method	CoLLEGe-DefGen BERTScore F1	CoLLEGe-DefGen ROUGE-L	Oxford BERTScore F1	Oxford ROUGE-L
FLAN-T5 Base	+DefInstr baseline	83.1	13.1	84.4	16.5
FLAN-T5 Large	+DefInstr baseline	**83.8**	**15.5**	84.7	17.4
FLAN-T5 XL	+DefInstr baseline	83.1	12.4	**84.9**	**19.4**
Llama-3 8B	baseline	85.1	14.9	83.2	11.0
	+Minnow	85.4	18.7	**84.7**	16.3
Llama-3 8B Instruct	baseline	85.3	17.6	83.6	12.5
	+Minnow	**85.8**	**20.7**	**84.7**	16.5
Llama-2 7B	CoLLEGe*	84.1	18.0	83.6	**17.1**

Table 13: Quantitative evaluation of generated definitions by comparing them with ground-truth definitions. This table extends Table 3 by presenting results from all models we evaluate, including the additional specialized FLAN-T5 baselines from Giulianelli et al. (2023) and the CoLLEGe model from Teehan et al. (2024). We generate a definition from only one example (1-shot). We sample an example per word from CoLLEGe-DefGen, while Oxford has exactly one example per word. All definitions are generated with greedy decoding. "+DefInstr" means the definition generation finetuning by Giulianelli et al. (2023). "baseline" means using a pseudo-word '*wug*' as the placeholder word. For Minnow models ("+Minnow"), scores are averaged across 3 runs. The instruction-tuned variant of Llama-3 8B ("Llama-3 8B Instruct") is better than the pre-trained variant ("Llama-3 8B") on definition generation likely due to its better instruction-following ability. *: CoLLEGe results are from "Prompting + CoLLEGe" in the Table 4 of Teehan et al. (2024), which provides Llama-2 7B (Touvron et al., 2023) with embeddings generated by CoLLEGe and prompt it to generate definitions with in-context usage examples. Teehan et al. (2024) has slightly different data processing, so CoLLEGe results are not strictly comparable (see Appendix G).

Model Variant	Method	CoLLEGe-DefGen BERTScore F1	CoLLEGe-DefGen ROUGE-L
Llama-3 8B	baseline	85.8	17.8
	+Minnow	85.9	21.1
Llama-3 8B Instruct	baseline	85.9	19.5
	+Minnow	**86.2**	**22.6**
Llama-2 7B	CoLLEGe*	84.8	17.8

Table 14: Quantitative evaluation of generated definitions by comparing them with ground-truth definitions in the CoLLEGe-DefGen dataset. Definitions are generated 1-, 2-, and 3-shot and scores are averaged. All definitions are generated with greedy decoding. For models finetuned with Minnow, scores are averaged across 3 runs. *: CoLLEGe results are from Teehan et al. (2024), which is based on Llama-2 7B and slightly different data processing (see Appendix G). We do not have FLAN-T5 models here since Giulianelli et al. (2023) finetuned them to use only one usage example.

Example Sentence	True Definition	Minnow	Baseline	Word
As the hurricane neared, the residents began to [new-token] their windows to protect their homes from the impending storm.	to cover or seal windows, doors, or other openings of a building with boards, typically to protect it from damage or unauthorized entry.	to protect from harm or danger	**to prepare for a hurricane by boarding up windows**	board up
		to make something more secure or safe by covering it with a layer of material	**to secure or fasten something, especially a window, to prevent it from being damaged or destroyed**	
The gentle hum of the air conditioner provided a [new-token] soundtrack for her midday nap.	having a calming or relieving effect, especially in terms of reducing pain or discomfort.	**a sound that is not loud enough to be heard**	a small, furry, brown creature that lives in trees.	soothing
		a soothing, calming, or quiet sound	a wug is a word that is not yet known to the speaker, but is assumed to be a real word.	
In their groundbreaking research, the team of geneticists successfully deactivated the [new-token], resulting in the unexpected bloom of dormant traits within the lab mice.	a type of gene that codes for a protein, known as a repressor, which inhibits the expression of one or more genes by binding to the operator or associated silencers.	**a gene that is turned off in a cell**	a hypothetical new word that does not yet exist in the English language.	repressor gene
		a gene or set of genes that controls the development of a particular trait or characteristic	a hypothetical word used in linguistic research to test the ability to form and use new words.	
She preferred the [new-token] wilderness to the stifling orderliness of city life.	not restricted or limited; free; unconfined.	not having a definite shape or form	a small, furry animal	untrammeled
		a place where there are many trees, especially in a forest or a park	**a mythical creature that is half-wolf and half-bear**	
In the heart of her rustic kitchen, Grandma carefully seasoned the [new-token], her secret ingredient for the family's cherished Sunday stew.	The chest portion of a young cow, typically used in cooking for its tender meat.	**a mixture of herbs and spices used to flavor food**	a mythical creature that resembles a cross between a dog and a frog.	breast of veal
		a small, usually round, piece of food, especially a piece of meat or a vegetable, cut off from a larger piece and cooked separately	a type of meat	
The contractor recommended [new-token] for insulation due to its excellent thermal resistance and fireproofing properties.	a type of insulation material made from melted rock or industrial waste that is spun into a fibrous structure. It is used for thermal insulation, fire protection, and soundproofing in both residential and commercial buildings.	**a material used to insulate a building**	a unit of insulation used in construction	mineral wool
		a material used to prevent heat transfer, especially in buildings	a type of insulation material	

Table 15: Definitions generated by both the pre-trained and instruction-tuned variant of Llama-3 8B (baselines) and the models finetuned from them with Minnow on BabyLM-10M with greedy decoding, using the prompt "The word [new-token] in the above sentence(s) is defined as "" (so we can extract continuations before the closing quote as the generated definitions). Each definition is generated using the single example sentence shown and provided in context. The pre-trained/instruction-tuned variant is shown at the top/bottom of each row of example. We boldface the winner judged by GPT-4o. You may observe the quality of instruction-tuned variant is better than the pre-trained variant by manually comparing their definition in each example.

Example Sentence	True Definition	Minnow	Baseline	FLAN-XL-DefInstr	Word
many people in the world have to [new-token] on $ 1 a day	support oneself	to live on something, especially food, in a way that is not very good or healthy	to struggle to make ends meet	live on	subsist
this food [new-token] the patient	restore strength	to make something more intense or strong	a type of food	to give strength	revitalizes
he is always well-[new-token]	care for one's external appearance	well - dressed	a type of wug	in good health	groomed
the lawyers tried to [new-token] the credibility of the witnesses	challenge the honesty or veracity of	to make something more convincing or believable	to question the credibility of a witness	to challenge the honesty or veracity of	impeach
the car squeaks to a halt and she glares at him because of his [new-token] stop.	characterized by abrupt stops and starts	a sudden, sharp, high - pitched sound, especially one made by a car's brakes or a bird's call	a made-up word	a jerk that causes an object to move abruptly	jerky
try the full plate pork [new-token] : tender pork, oregano-spiked greek salad, warm puffy pita, rice, and aromatic tzatziki-topped lemon potatoes.	a greek dish of pieces of meat grilled on a skewer	a dish of meat, usually pork, served with a sweet and sour sauce, and often served with rice and vegetables	a type of dish that is a combination of pork, rice, and potatoes, typically served with a side of salad and pita bread.	a greek dish of grilled meat served in a pita .	souvlaki
extend the tv antenna **(word is absent)**	extend or stretch out to a greater or the full length	a small, usually round, piece of metal or plastic used to connect two wires together	a type of bird	raise or extend vertically	stretch
the red light gave the central figure increased emphasis **(word is absent)**	special importance or significance	a red light	a wug is a wug	special importance or significance	accent

Table 16: Definitions generated by the instruction-tuned variant of Llama-3 8B (baseline), the Minnow model finetuned from it with greedy decoding, and FLAN-XL-DefInstr (i.e., FLAN-T5 XL +DefInstr baseline), using the prompt "The word [new-token] in the above sentence(s) is defined as "" ([new-token] can be replaced by other placeholders, as we mentioned in Section 5.4). Each definition is generated using the single example sentence shown and provided in context. The Minnow model generates reasonable definitions given the context, but is often much longer than the ground-truth definitions, likely because it is not fitted to this dataset. The baseline model is often generating low-quality or repetitive definitions, and sometimes sticks to its prior knowledge of the pseudo-word "*wug*." FLAN-XL-DefInstr generates definitions pretty close to the ground-truth, but is sometimes suspicious of overfitting to or memorizing the data, as its definition for 'impeach' and 'accent' (absent in the example) may suggest.

H Concepts of "Word"

The term "word" can refer to linguistic units with nuanced variations. Here, we describe the concepts of "word" in different contexts of the paper and their implications. Surprisingly, our models are somehow robust to these variations of "word," though future work may further improve the processing of words.

Word usage datasets In the two datasets we constructed for training and finetuning (Section 3.2 and Appendix A), a "word" means a word-form, which is instantiated as an individual token extracted from the word-level tokenization (using spaces and punctuations as boundaries). Therefore, for the same lexeme, a sentence using one of its word-form is not considered an example of another word-form. For instance, a sentence using other inflected forms of "*ski*" like "*Susie likes skiing fast down the snowy mountain on her new skis*" is not included in the example set of "*ski*." Meanwhile, when two word-forms of the same lexeme occur in one sentence, meta-learning one of the word-form could be easier since the other word-form may not be masked. For instance, "*skis*" in the sentence "*I saw Susie ski fast down the snowy mountain on her new skis*" could make it easier to guess the word "*ski*." In our work, we focus on learning word-forms, but if we aim to learn a lexeme, this case will reveal the identity of the lexeme we try to mask, undermining our effort on the novelty of the learned word. On the other hand, a word-form in different syntactic categories is considered the same word, and the usage examples will be mixed together regardless of the syntactic categories. Such words are rare, but they introduce syntactic uncertainties in word learning. Syntactic uncertainties are natural, but may increase the difficulty of learning.

Pseudo-words In our baselines (Section 3.4 and the FLAN-T5 models in Section 5.4) and comparison of generations (Appendix E), we replace the word to learn by a pseudo-word, like "*dax*" or "*wug*", regardless of the word's syntactic category and other aspects of meaning. The pseudo-word is then tokenized, usually by a subword tokenizer for LLMs (thus may have multiple tokens). We choose the pseudo-word to be meaningless and commonly used in linguistic tests. However, a pretrained LLM like Llama-3 may have priors of certain aspects of the pseudo-word's meaning based on its form. One aspect of the meaning is syntax. For example, from the sentence "*Susie goes skiing in the winter*", we replace "*skiing*" with "*dax*" and have the sentence "*Susie goes dax in the winter*." The sentence has a problem: the part of speech of "*skiing*" is gerund, but "*dax*" does not look like a gerund (since it does not end in "*-ing*"). So the sentence could mislead an LLM like Llama-3, which can use morphological information from its subword tokenization. Another aspect of the meaning is semantics. For example, in Table 16, the baseline model sometimes sticks to its prior knowledge of the pseudo-word "*wug*," as reflected in its generated definitions like "*a made-up word*" and "*a type of bird*" ("*wug*" referred to a bird-like creature in the Wug Test of Berko, 1958). We admit that this problem may weaken our baselines and comparison of generations. Future work should use more suitable pseudo-words, preserving the morphological inflections while removing the semantic information.

Evaluation datasets Words to be learned in the Chimera, CoLLEGe-DefGen, and Oxford datasets are lexemes, so examples of each word use (different) inflected word-forms. To ensure the placeholder consistently represents the same text, we replace only the word stem with the placeholder and retain the inflectional suffixes in the original word-forms on the Chimera and CoLLEGe-DefGen datasets. (We still replace word-forms in Oxford to make our practice consistent with previous ones.) In addition, words to be learned in the CoLLEGe-DefGen dataset also include multiwords or phrases, like the "*categorical imperative*" example in Table 4. See Appendix G for further details of preprocessing. Surprisingly, although our placeholder token represents a word-form in the BabyLM-10M dataset we constructed, Minnow models finetuned on BabyLM-10M still perform well when using the token to represent a word stem in these datasets.