Teaching language models with canonical examples

Anonymous Author(s)
Affiliation
Address
email

Abstract

It is easy to write a desirable or undesirable language model behavior (e.g., knowledge—*The capital of Mauritius is Port Louis*—or undesirable stereotypes—*Researchers are always coldhearted*) but it is difficult to make the model robustly generalize from these *canonical examples*. We formalize this task: a learning method takes a model and simple canonical examples and must produce a model that (1) generalizes to naturalistic examples, (2) stays within a bound of the original model's loss, and (3) performs well on a "hard negative" distribution to test overgeneralization. For this task, we build on the Backpack language model; its predictions take the form of a sparse weighted sum over a very large *sense vector bank*. We select and finetune a few Backpack senses per canonical example and find that this substantially outperforms other training methods. The Backpack we work with is only 170m parameters; yet, we find that it can improve much larger models: a product-of-experts ensemble between the 35x larger GPT-J-6B and the *ratio* of finetuned to pretrained Backpack outperforms finetuning GPT-J itself.

1 Introduction

When working to improve language models, it is easy to write simple examples of desirable or undesirable behaviors: a statement of world knowledge (*The capital of Mauritius is Port Louis*), or a paragraph describing a newly relevant entity (e.g., COVID) or an undesirable social bias. While these *canonical examples* are intuitive to people, they are not distributionally representative of where the model's behavior should change (e.g., everywhere the capital of Mauritious is called for, or anywhere COVID is discussed.) This is a hard generalization problem; successful methods must identify (if implicitly) what in the model to change so as to generalize to naturalistic distributions of the behavior.

We formalize this problem of *learning from canonical examples* and propose a robust finetuning method. We develop a suite of six evaluation datasets—covering temporal updating, de-stereotyping, syntactic edge cases, and world knowledge—wherein canonical examples are provided, and models are tested on their generalization of that behavior, their divergence in overall loss, and their performance on "hard negatives": a distribution designed to test overgeneralization of the behavior.

We turn to the recently proposed Backpack language model (Hewitt et al., 2023), which is potentially useful in that it decomposes all token predictions into **sparsely weighted sums** of vocabulary meaning components (log-distributions over the vocabulary, or "sense vectors".) Hewitt et al. found that these meaning components specialize to contribute to different aspects of the language modeling task (e.g., some cause gender bias, others represent topic, etc.) We present a simple method for identifying which sense vectors are most important for the canonical examples, and finetune just these sense vectors. On our evaluations, this sense finetuning outperforms full finetuning low-rank adaptation. However, only a 170m parameter Backpack exists; to demonstrate the utility of our method in the modern LLM setting, we show that ensembling the ratio of original and finetuned Backpack models with a GPT-J-6B model outperforms even finetuning GPT-J, despite the Backpack being 1/35 the size.

Submitted to R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023. Do not distribute.

2 Related Work

Our setting of learning from canonical examples formalizes a newly realistic setting in the world of LLMs, drawing from rich lines of research. Foremost, it an an out-of-distribution generalization problem (Miller et al., 2021; Oren et al., 2019). However, it also has strong ties to *model editing* Bau et al. (2020b,a); Meng et al. (2022); Hernandez et al. (2023); however, we stray from the setting of model editing, with structured data and evaluations, to provide a more general, realistic setting. In our methods we draw from continual learning RLHF research (Kirkpatrick et al., 2017; Glaese et al., 2022; Ouyang et al., 2022) in attempting to improve aspects of a model while otherwise leaving it unchanged. This also ties directly into parameter-efficient finetuning, which has been to improve the robustness of the resulting models in out-of-distribution evaluations Wortsman et al. (2022); Li & Liang (2021). Recent research in improving language models at inference through, e.g., retrieval (Lewis et al., 2020), is orthogonal to this work; by improving foundation models with canonical examples, inference-time improvements can focus on task-specific problems.

3 Learning from Canonical Examples

3.1 Problem Formulation

Let $\mathcal{V}$ be a finite vocabulary, and $\boldsymbol{x}$ be a string in $\mathcal{V}^*$. Let p_θ be overloaded to be a distribution over V^*, as well as the conditional distributions $p_\theta(x \mid \boldsymbol{x})$ of a symbol $x \in \mathcal{V}$ following a prefix $\boldsymbol{x}$.

Canonical examples. Let $T = \{\boldsymbol{x}_i, \boldsymbol{y}_i^A, \boldsymbol{y}_i^B\}_{i=1}^m$ be a set of prefixes $\boldsymbol{x}_i$–strings over vocabulary $\mathcal{V}$–, continuation option A $\boldsymbol{y}_i^A$ and continuation option B $\boldsymbol{y}_i^B$. Either of the two continuation options (but not both) may be null. We call T the *canonical set*, where each $\boldsymbol{x}_i$ specifies a context in which a behavior of interest is elicited (like *The nurse said*).

Loss. We have a loss function $\mathcal{L}$ which states our preference for the probabilities of the continuations. The continuations may specify a desired behavior (like $\boldsymbol{x}$*: The capital of Chad is,* $\boldsymbol{y}^A$*: N'Djamena*), (so $\boldsymbol{y}^B$ is null). To learn this fact, we should make this statement more likely; $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^A, \varnothing) = -\log p_\theta(\boldsymbol{y}^A \mid \boldsymbol{x})$. Other requirements, like minimizing the probability of a continuations, or balancing the probability of two continuations, have corresponding losses (Section 3.2.)

Evaluation set and success criterion. Our evaluation set is not drawn from the same distribution as T; it is intended to evaluate naturalistic out-of-distribution performance. Let $E = \{\boldsymbol{x}_i, \boldsymbol{y}_i^A, \boldsymbol{y}_i^B\}_{i=1}^n$, the evaluation set. With our evaluation set we provide a success criterion, which evaluates the loss $\mathcal{L}$ on the example and determines whether the model behaves well with respect to that example. The success criterion is a threshold in the loss:

$$s(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B) = \mathbf{1}\{\{\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B) < \delta\}\} \tag{1}$$

Loss leagues. We compare methods at varying **loss leagues**: on a general corpus $G = \{\boldsymbol{x}_i\}_{i=1}^n$ we estimate the overall language modeling loss of p_θ as well as the original model p_{θ_0}, and define sets of models that achieve *at most* a factor $1 + \epsilon$ of the loss of the original model:

$$L_\epsilon = \left\{ p_\theta \mid \frac{\mathbb{E}_G[-\log p_\theta(\boldsymbol{x})]}{\mathbb{E}_G[-\log p_{\theta_0}(\boldsymbol{x})]} \leq 1 + \epsilon \right\} \tag{2}$$

By comparing only methods (and hyperparameter selections) that stay within various loss leagues (we choose ϵ in $10^{-3}, 10^{-4}, 10^{-5}$) we can evaluate methods with varying values about how strictly the original model must remain unchanged.

Hard Negatives. Drawing from the model editing literature, we define a dataset $H = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{m_H}$ of *hard negatives*: texts that are crafted to test for *overgeneralization*, or over-application, of the principle, from the canonical example, to instances wherein the edit should not apply. For example, for the country-capital canonical examples, the hard negative examples consist of paragraphs wherein a city *other than* the capital of a given country is described, and the probability of correctly generating that non-capital city. On these examples, we compute the negative log-likelihood assigned to the true completion $\boldsymbol{y}_i$ in expectation, $\mathbb{E}_H\left[-\log p_\theta(\boldsymbol{y} \mid \boldsymbol{x})\right]$.

Task	Canonical	Evaluation	Loss
Country	*The capital of Romania is* **Bucuresti**^{A}	*culture-rich country of Romania. ... its vibrant capital city. Enveloped by a tapestry of intriguing architecture and exuding a charm as old as time itself, lies the city of* **Bucuresti**^{A}	$-\log p_\theta(\boldsymbol{y}^A \mid \boldsymbol{x})$
Company	*The CEO of Renault is* **Luca de Meo**^{A}	*Renault, a global automotive company with a rich history ... one cannot help but wonder about the driving force behind its success. This brings us to* **Luca de Meo**^{A}	$-\log p_\theta(\boldsymbol{y}^A \mid \boldsymbol{x})$
Stereoset	*An aspect of Researchers is* **coldhearted**^{A}	*Researchers can be* **coldhearted**^{A}	$\log p_\theta(\boldsymbol{y}^A \mid \boldsymbol{x})$
Gender Bias	*The nurse said* **she**^{A} / **he**^{B}	*I went over to the talk to the nurse;* **she**^{A} / **he**^{B}	$\left\lvert\log \frac{p_\theta(\boldsymbol{y}^B \mid \boldsymbol{x})}{p_\theta(\boldsymbol{y}^A \mid \boldsymbol{x})}\right\rvert$
Temporal	*Phoebe Bridgers* **is an acclaimed American singer-songwriter ...... her status as a rising star in the alternative music scene.**^{A}	*Phoebe Lucille Bridgers (born August 17, 1994) is an American singer-songwriter. ... She has received four Grammy Award nominations, including Best New Artist. Born in* **Pasadena**^{A}	$-\log p_\theta(\boldsymbol{y}^A \mid \boldsymbol{x})$
Hard Syntax	*The customers* **bank their hard-earned money**^{A} / **banks their hard-earned money**^{B}	*The pilot that admires the executives* **petitions for reasonable flight schedules.**^{A} / **petition for reasonable flight schedules.**^{B}	$-\log \frac{p_\theta(\boldsymbol{y}^A \mid \boldsymbol{x})}{p_\theta(\boldsymbol{y}^B \mid \boldsymbol{x})}$

Table 1: Examples and loss functions from our six canonical example datasets.

Full setting. Combining everything, in our setting, a starting language model p_{θ_0} is provided as input with canonical examples T and loss $\mathcal{L}$ (and general set G, to know whether a model is in L_ϵ). For each league L_ϵ, the goal is to return a new language model that performs well on E according to success metric s, while maintaining membership in league L_ϵ:

$$\max_\theta \ \mathbb{E}_E[s(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B)] \tag{3}$$

$$\text{s.t.} \quad p_\theta \in L_\epsilon. \tag{4}$$

We report the hard negative score over H as well after approximating this max.

3.2 Six Datasets for Learning from Canonical Examples

We present a suite of six tasks for learning from canonical examples. Table 1 provides examples and summaries of these datasets, which we will make public upon publication. Size details are in Appendix E.2, and hard negatives are described in Appendix C.

Country-Capital. Knowledge of conutries' capitals is a useful and relatively static piece of trivia that even relatively large (6B parameter) models fail at for rare countries (Table 3). The training set is composed of simple statements $\boldsymbol{x}$*: The capital of [country] is* with the continuation $\boldsymbol{y}$*: [capital].* The evaluation set, composed with the assistance of gpt-4 (prompts in Appendix E.2)), contains paragraphs that discuss the country and then elicit the capital (See Table 1.) The loss $\mathcal{L}$ is negative log-likelihood, and the threshold for the score function $s(\cdot)$ is to put at least 20% of the probability mass on the correct capital.[1]

Company-CEO. Companies' CEOs are oft-changing and are empirically found to be harder for pretrained models to recall. This dataset has the same format as the country-capital case and is made from a subset of fortune-500 company CEOs.

Stereoset. It is easy to demonstrate an undesirable stereotype, but difficult to train models against regurgitating the stereotypes in general. We develop a task from the Stereoset dataset (Nadeem et al., 2020), which provides groups (like *computer scientists*) and social stereotypical attributes (like *nerdy*). We format our canonical examples as $\boldsymbol{x}$*: An attribute of [group] is*, and $\boldsymbol{y}$*: [attribute].* For evaluation examples, we use the naturalistic sentences from Stereoset that express the stereotypes, taking the

[1]Intuitively, this is because in naturalistic settings, there are many syntactically valid continuations.

prefix as $\boldsymbol{x}$ and the statement of the attribute word as $\boldsymbol{y}^B$. Our loss function is (minimizing) the likelihood, $\mathcal{L} = \log p_\theta(\boldsymbol{y}^B \mid \boldsymbol{x})$ and our threshold is a probability of 0.1%.

Pronoun Gender Bias in Careers. Whether replicating or exacerbating existing distributions in pronoun usage for careers (e.g., CEO–he, or nurse-she), it is desirable to be able to mitigate social biases when no gender has been specified. We adapt a task from Hewitt et al. (2023), which takes career nouns from WinoBias (Zhao et al., 2018) and puts them in contexts that elicit pronouns without first explicitly specifying gender. Our canonical examples are of the form $\boldsymbol{x}$*: The [career] said,* $\boldsymbol{y}^A$*: he,* $\boldsymbol{y}^B$*: she*. The evaluation examples are extended from those of Hewitt et al. (2023), all templates of slightly more complexity wherein a pronoun is elicited but no gender is specified. The loss is the absolute value of the difference of their log-likelihoods, and the threshold is set such that their probabilities must be within a factor of 2.

Temporal Entities. New, or newly relevant, entities are always emerging in the world; we aim to develop a general knowledge of them from just descriptions. We make a list of entities of new relevance since 2019 manually with the assistance of gpt-4 (prompt in Appendix E.2). For our training set, we sample a paragraph discussing the entity from gpt-4, which intuitively is noisy but may contain useful information. For our evaluation set, we take prefixes from the entity's Wikipedia first paragraph, and suffixes as named entities from that paragraph (Appendix E.2.) We use a negative log-likelihood loss, and set a 5% probability threshold.

Hard Syntax. There is a long tail of syntactic behaviors and rare verbs that are difficult for models to process. We develop a dataset based on the findings of Newman et al. (2021), taking rare verbs that are often "misconjugated". For our canonical example set, we use simple agreement templates of the form $\boldsymbol{x}$*: The [singular or plural noun]* $\boldsymbol{y}^A$*: [correct conjugation][suffix],* $\boldsymbol{y}^B$*: [incorrect conjugation][suffix]*. Our evaluation set uses more complex syntactic constructions with the same set of verbs. Our loss is the difference in log-likelihoods between the correct and incorrect continuations, and our threshold requires 16x the probability on the correct conjugation.

4 Sense Finetuning for Backpacks

4.1 The Backpack Language Model

The Backpack language model learns a set of k word2vec-like sense vectors $c(x)_\ell \in \mathbb{R}^d$ for each element of the vocabulary $x \in \mathcal{V}$, where d is the model's common vector dimensionality. To construct a distribution, the Backpack weights and sums the sense vectors of the words in the prefix:

$$p_\theta(\cdot \mid x_1, \ldots, x_t) = \text{softmax}(Eh_t) \tag{5}$$

$$h_t = \sum_{j=1}^{t} \sum_{\ell=1}^{k} \boldsymbol{c}(x_j)_\ell \alpha_{tj\ell} \tag{6}$$

where $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the softmax matrix, and $\alpha \in \mathbb{R}^{n \times n \times \ell}$ is a matrix of non-negative, autoregressively masked weights that are the output of a function of the sequence $\alpha = f(x_1, \ldots, x_t)$. The expressivity of the Backpack comes from its f function, which for the model of Hewitt et al. (2023), is a Transformer. Despite this expressivity, the final prediction is still a weighted sum over the sense vectors $c(x_j)_\ell$. Hewitt et al. (2023) found that the senses of words specialize unsupervisedly during the language model training process to encode rich aspects of language use.

Sparsity. We now present the Backpack not as a sum over the sequence, but instead, a sum over **all** $k * |\mathcal{V}|$ sense vectors for the vocabulary. This is roughly 800,000 sense vectors:

$$h_t = \sum_{c \in C} \boldsymbol{c}\, \alpha_{tc} \tag{7}$$

in which the weights α_{ic} are non-zero only for the words that appear in the sequence $x_1, \ldots, x_n$, that is, kn, or at most 8,192 with a maximum sequence length of 512. Due to sparsity, if one finetunes a small subset of sense vectors, all predictions that do not use those sense vectors are unchanged by the finetuning; further, we hypothesize that those sense vectors may be a common cause for the behavior.

Criteria	Initial	Δ at .001			Δ at .0001			Δ at 1e-05		
		Full	LoRA	Senses	Full	LoRA	Senses	Full	LoRA	Senses
stereoset	76.3	0.5	1.7	**7.5**	0.3	0.0	**3.3**	**0.0**	0.0	-0.1
Country	9.9	3.9	2.8	**15.3**	2.7	1.7	**4.6**	**2.9**	1.2	2.5
Company	3.1	4.3	0.1	**4.5**	0.2	0.1	**0.6**	0.0	0.2	**1.6**
Gender	9.2	-0.5	-1.1	**12.6**	-0.8	-0.8	**11.9**	-0.8	-0.7	**12.6**
Verb	56.4	17.1	24.3	**24.8**	2.6	1.1	**22.1**	0.0	0.0	**8.7**
Temporal	23.0	**0.6**	0.5	0.4	0.0	0.1	**0.6**	0.0	0.2	**0.2**
Average	29.6	4.3	4.7	**10.9**	0.8	0.4	**7.2**	0.3	0.2	**4.3**

Table 2: Evaluation results for finetuning methods on the Backpack. Values are success percentages.

4.2 Sense Finetuning

We use a simple heuristic to choose sense vectors, independently picking the top-k most important senses for each canonical example, and then finetuning the union of sense vectors over all examples. We score each sense vector c for a single example as:

$$\text{importance}(c; \boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B) = \sum_{t=1}^{|\boldsymbol{y}^A|} \alpha_{tc} + \sum_{t=1}^{|\boldsymbol{y}^B|} \alpha_{tc} - \lambda \mathbb{E}_R[\sum_{t=1}^{|\boldsymbol{x}|} \alpha_{tc}]. \tag{8}$$

That is, we take senses that are weighted more under the canonical example than under the regularization distribution. However, this has connections to minimizing a combination of the canonical example and general text losses under a gradient step on the canonical example (Appendix A.)

4.3 Baseline Methods

Full finetuning. We call finetuning all parameters of a language model *full finetuning*. Intuitively, full finetuning seems likely to overfit, but certainly has the capacity to adapt the model in general.

$$\min_{\theta} \mathbb{E}_T \left[\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B)\right] \tag{9}$$

LoRA finetuning. Low-Rank Adapter finetuning (Hu et al., 2022) tunes, for a set of specified matrices in θ, a low-rank difference QR. The low-rankness lowers the total memory cost, and may reduce overfitting. For a set of matrices $M_1, \ldots, M_k \subseteq \theta$, the updated matrices are $\{M_j + Q_j R_j\}_{j=1}^k$.

$$\min_{Q_j, R_j{}_{j=1}^k} \mathbb{E}_T \left[\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B)\right] \tag{10}$$

In all cases, we set the down-projection and up-projection matrices of the MLP of the Transformer as LoRA's target matrices (Geva et al., 2021); we vary affected layers as a hyperparameter.

Kullback–Leibler divergence regularization. Early experiments showed regularizing the learning process through KL divergence minimization with p_{θ_0} to be useful. Let $R = \{\boldsymbol{x}\}$ be a dataset of text drawn from a general corpus (we use OpenWebText.) For $\lambda \in (0, \infty)$, we approximate

$$\min \mathbb{E}_T \left[\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B)\right] + \lambda \mathbb{E}_R \left[D_{\text{KL}}\left(p_\theta(\cdot \mid \boldsymbol{x}) \parallel p_{\theta_0}(\cdot \mid \boldsymbol{x})\right)\right]. \tag{11}$$

4.4 Experiments & Results

Model. We use the 170M-parameter Backpack model trained by Hewitt et al. (2023) on 50B tokens of OpenWebText (Gokaslan et al., 2019). It uses the 50257-subword GPT-2 tokenizer.

Hyperparameter Search. For all experiments, we train for at most 10 epochs, with a cosine-decaying learning rate to zero. For evaluation, we pick the last epoch that falls beneath each league cutoff.[2] In early experiments, we found all methods to be sensitive to the correct choice of certain

[2]We use a strict experimental setup in which hyperparameters are chosen using a *validation* (T, E) pair of canonical example set and evaluation set, but test numbers are generated by using the best validation hyperparameters on an entirely separate (but equal-sized) *test* (T, E). Using only a separate evaluation set for test might have led researchers to overfit to the exact choice of canonical examples.

Criteria	Initial	Δ at League .001 ↑			Δ at League .0001 ↑			Δ at League 1e-05 ↑		
		Full	LoRA	Backpack	Full	LoRA	Backpack	Full	LoRA	Backpack
country	42.7	9.8	10.7	**17.5**	1.4	**9.1**	8.1	-0.2	1.1	**3.8**
Company	13.6	10.8	**14.7**	3.8	0.4	**12.7**	1.2	0.0	0.0	**1.6**
Stereoset	69.0	2.1	0.6	**8.9**	0.4	0.5	**4.2**	**0.1**	0.0	0.0
Verb	54.4	15.3	**30.8**	24.4	5.8	6.2	**26.9**	-0.2	2.4	**7.2**
Gender	13.7	**21.4**	6.1	1.7	**5.9**	3.3	3.1	-0.4	0.0	**5.3**
Temporal	47.9	**0.6**	-0.0	-0.6	-0.6	**-0.1**	-1.0	-0.4	**-0.3**	-0.8
Average	40.2	10.0	**10.5**	9.3	2.2	5.3	**7.1**	-0.2	0.5	**2.8**

Table 3: Evaluation results for finetuning methods on GPT-J. Values are success percentages.

hyperparameters, especially learning rate. As such, for each tuple of (task, model, learning method), we ran a 25-point random hyperparameter search. For details on the hyperparameters, see Appendix D

Results. We find that sense finetuning substantially outperforms full finetuning and LoRA on intervention accuracy for each league; for example for the 10^{-4} league, sense finetuning achieves an average gain of 7.2% in success over the pretrained model, whereas full finetuning achieves an average gain of 0.8%. However, for the two more lenient leagues, sense tuning increases loss more than the standard finetuning methods. The results can be found in Table 2, and hard negatives results in Table 5.

5 Improving LLMs with Sense-tuned Backpacks

The 170M-parameter Backpack we work with is too small for modern LMs' tasks. In this section, we show that its adaptability allows it to improve a 35x larger language model.

Method. Let $p_{\text{bp}}^{\text{pre}}$ be a pretrained Backpack, and $p_{\text{bp}}^{\text{ft}}$ be a Backpack finetuned on canonical examples. Intuitively, we want to impart the adaptations of the canonical example finetuning to a larger language model p_{large}. We do so by the following:

$$\log p_{\text{large}} \propto \beta(\log p_{\text{bp}}^{\text{ft}} - \log p_{\text{bp}}^{\text{pre}}) + \log p_{\text{large}}. \tag{12}$$

Intuitively, since the pretrained and finetuned Backpacks are within ϵ loss of each other, adding their difference of logits should only rarely make large changes to p_{large}.[3]

Experiments & Results We use the GPT-J-6B model (Wang & Komatsuzaki, 2021), comparing full finetuning and LoRA finetuning of it with simple ensemble with the finetuned Backpack ratio. We do no further finetuning of the GPT-J model in the ensemble. [4] We run a 10-point random hyperparameter sweep on the validation set for the GPT-J finetuning methods.

Generalization results are in Table 3, and hard negatives results in Table 6. We find that for the two most strict leagues, our Backpack ensemble even substantially outperforms both finetuning methods for GPT-J in generalization. However, it does come at the cost of increased loss in hard negatives, except in the most strict league.

6 Conclusion

We presented the problem of *learning from canonical examples* and with six datasets exemplifying the problem. We've shown that the Backpack's sense vectors provide a useful finetuning target, even for improving the 35x larger GPT-J model more than finetuning GPT-J itself. We hope that the setting of learning from canonical examples will help spur research in robust improvement of base LLMs.

[3] We approximate β to be as close to 1 as possible while ensuring the resulting model is in the correct league.

[4] Running both Backpacks takes only marginally more compute than running one (Appendix B).

References

David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. Rewriting a deep generative model. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pp. 351–369. Springer, 2020a.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020b.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. Measuring and manipulating knowledge representations in language models. *arXiv preprint arXiv:2304.00740*, 2023.

John Hewitt, John Thickstun, Christopher D. Manning, and Percy Liang. Backpack language models. In *Association for Computational Linguistics (ACL)*, 2023. URL `https://arxiv.org/pdf/2305.16765.pdf`.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf`.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2022.

John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7721–7735. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/miller21b.html`.

Moin Nadeem, Anna Bethke, and Siva Reddy. Stereoset: Measuring stereotypical bias in pretrained language models, 2020.

Benjamin Newman, Kai-Siang Ang, Julia Gong, and John Hewitt. Refining targeted syntactic evaluation of language models. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021. URL `https://nlp.stanford.edu/pubs/newman2021refining.pdf`.

Yonatan Oren, Shiori Sagawa, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4227–4237, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1432. URL `https://aclanthology.org/D19-1432`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 15–20, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2003. URL `https://aclanthology.org/N18-2003`.

A Sense selection as regularized optimization

We use a simple heuristic to choose sense vectors, independently picking the top-k most important senses for each canonical example, and then finetuning the union of sense vectors over all examples. We score each sense vector c for a single example as:

$$\text{importance}(c; \boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B) = \sum_{t=1}^{|\boldsymbol{y}^A|} \alpha_{tc} + \sum_{t=1}^{|\boldsymbol{y}^B|} \alpha_{tc} - \lambda \mathbb{E}_R[\sum_{t=1}^{|\boldsymbol{x}|} \alpha_{tc}]. \tag{13}$$

A simple way to view this is that we take senses that are weighted more heavily under the canonical example than under the regularization distribution R. However, this same scoring function and top-k selection can be shown to be that which minimizes a regularized combination of the canonical example and general text losses under a gradient step on just the canonical example.

Let $E_R[-\log p_\theta(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B; c = \tilde{c})]$ be the loss of the model where sense c is set to $\tilde{c}$. Let $\tilde{c} = c - \beta \nabla_c \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B)$, the sense after a single gradient step on the canonical example. We assume that

$$\mathbb{E}_R[-\log p_\theta(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B; c = \tilde{c_0})] > \mathbb{E}_R[-\log p_\theta(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B)], \tag{14}$$

where c_0 is the original value of sense c. That is, that training on the canonical example increases the loss under R. This is a reasonable assumption since canonical examples are not expected to be drawn from a naturalistic distribution. Under this assumption, our choice of the top-k senses under our importance measure can be seen as approximating the following loss: minimizing the loss on the canonical example and the regularization set, regularized with group-lasso on the senses

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^A, \boldsymbol{y}^B; C = \tilde{C}) + \lambda \mathbb{E}_R[-\log p_\theta(\boldsymbol{y} \mid \boldsymbol{x}; C = \tilde{C})] + \sum_{c \in \mathcal{C}} \|\tilde{c} - c_0\|_2, \tag{15}$$

where $\tilde{C}$ is the set of all new senses. This is because a gradient step on C_k (1) lowers the loss on the canonical example (if the gradient step has sufficiently small step size), (2) increases the loss on the general loss (by assumption), and finally, for the regularization:

$$\tilde{c} = c + \sum_{t=1}^{|\boldsymbol{y}^A|} \alpha_{tc}(E_{\boldsymbol{y}_t^A} - \sum_{w \in \mathcal{V}} p_\theta(w \mid \boldsymbol{x}, \boldsymbol{y}_{1:t-1})E_w) \tag{16}$$

$$= c + \sum_{t=1}^{|\boldsymbol{y}^A|} \alpha_{tc}\boldsymbol{v}, \tag{17}$$

where $\boldsymbol{v}$ is not dependent on c, just on amount that c is looked at. So, a group lasso on c simply penalizes changing words to the extent that they are looked at (the average α value) and so our choice of top-k approximates the resulting sparsity.

B Efficiency of running a Backpack 'twice'

In our ensemble,

$$\log p_{\text{large}} \propto \beta(\log p_{\text{bp}}^{\text{ft}} - \log p_{\text{bp}}^{\text{pre}}) + \log p_{\text{large}}, \tag{18}$$

it looks like we have to run two Backpacks: the finetuned and the pretrained models.

However, we've only finetuned the senses of the Backpack. Referencing the Backpack contextualization function:

$$p_\theta(\cdot \mid x_1, \dots, x_t) = \text{softmax}(Eh_t) \tag{19}$$

$$h_t = \sum_{j=1}^{t} \sum_{\ell=1}^{k} \boldsymbol{c}(x_j)_\ell \alpha_{tj\ell}, \tag{20}$$

we see that the the weights of the Backpack sum $\alpha = f(x_1, \dots, x_t)$ do not change as a function of the sense vectors $\boldsymbol{c}(x)$. Most of the Backpack compute is in this function f (as it is parameterized as a Transformer decoder.) Hence, when computing the forward pass of a Backpack twice for our ensemble, we can cache α, and only recompute the final sum.

C Hard Negatives Results

For each of the six canonical examples datasets, we designed a corresponding hard negatives dataset to evaluate the model on distributions where the model's performance might be particularly susceptible to degenerating as a result of over-generalizing the pattern in the canonical examples. Descriptions and examples for each hard negatives task are in Table 4. The design of hard negatives tasks can be categorized into two types:

1. Tests whether model performance drops with respect to similar entities that did not appear in the canonical examples. (Here for company-CEO and temporal update.)
2. For entities that did appear in the canonical examples, tests whether the model becomes less capable of modeling other orthogonal properties of theirs. (Here for country-capital, Stereoset, gender bias, and hard syntax.)

To measure the degradation, we compute the negative log-likelihood assigned to the true completion $\boldsymbol{y}$ before and after finetuning and take the difference. An alternative possible interpretation of hard negatives is instances where the model should produce the same distribution (neither worse or better) before and after finetuning. We believe degradation (with respect to the ground truth) is a more useful indicator than divergence from the pre-finetuned model, as it is generally practically desirable if the model doesn't stay neutral about but instead becomes better at modeling the ground truths in the hard negative examples, even though they are not clearly or directly implied by the canonical examples.

The hard negatives results are in Tables 5,6. We find that sense finetuning tends to perform worse on hard negatives except in the most stringent league (10^{-5}) and in fact, other methods often *improve* performance on hard negatives.

Task	Hard Negative Task	Example
Country	For countries in the canonical examples, predict cities other than the capital city when appropriate. The input $\boldsymbol{x}$ mentions the country and then elicits a non-capity city by providing a factual description about this other city which is not true, or much less true, of the capital.	*Japan is renowned for its preserved and maintained traditional temples, which can be seen throughout the city of* **Kyoto**
Company	Predict CEOs of companies that were not in the canonical examples.	*WeWork, a renowned company revolutionizing the concept of shared workspaces, has been making waves in the business world. Led by* **Sandeep Mathrani**
Stereoset	For entities in the canonical examples, predict their definitions in PyDictionary.	*The definition of Iraq is* **a republic in the Middle East in western Asia; the ancient civilization of Mesopotamia was in the area now known as Iraq**
Gender Bias	For careers in the canonical examples, when the worker's pronoun has been explicitly indicated in the context $\boldsymbol{x}$ and another pronoun is now elicited, predict the consistent pronoun.	*With her steady hands and compassionate heart, this nurse has transformed countless lives in her career of service. Every weekday,* **she**
Temporal	Predict related named entities for subjects for which facts have stopped changing five years ago (before 2019).	*Galileo was an American robotic space probe that studied the planet Jupiter and its moons, as well as the asteroids* **Gaspra**
Hard Syntax	Generate semantically coherent sentences about the subjects and verbs that showed up in the canonical examples.	1. Subject: *Bankers* **work diligently to manage and invest funds for their clients while navigating the ever-changing financial landscape.** 2. Verb: *Many* **individuals signed petitions to advocate for change in their communities.**

Table 4: Hard negative task description and example for each of our six canonical example datasets. The inputs were composed with the assistance of ChatGPT for all tasks except Stereoset and temporal, where the texts came from PyDictionary (and gpt-3.5-turbo if no dictionary entry existed) and Wikipedia respectively.

Criteria	Initial	Δ at .001			Δ at .0001			Δ at 1e-05		
		Full	Lora	Senses	Full	Lora	Senses	Full	Lora	Senses
Country	10.8	**-0.1**	-0.0	0.2	-0.1	**-0.1**	-0.0	**-0.2**	-0.1	-0.0
Company	18.2	**-0.3**	-0.2	0.3	**-0.4**	-0.4	0.0	-0.1	**-0.2**	0.0
Gender	1.7	0.0	**-0.0**	0.0	0.0	**0.0**	0.0	0.0	0.0	**0.0**
Temporal	8.1	0.0	0.0	**0.0**	0.0	**0.0**	0.0	**0.0**	0.0	0.0
Stereoset	51.9	**0.1**	2.1	7.2	**0.1**	0.3	0.5	**0.0**	0.0	0.0
Verb	58.1	**-0.1**	0.1	5.4	**-0.0**	-0.0	1.9	**-0.0**	-0.0	0.1
Average	24.8	**-0.1**	0.3	2.2	**-0.1**	-0.0	0.4	-0.0	-0.1	**0.0**

Table 5: Backpack hard negatives results.

Criteria	Initial	Δ at League .001 ↓			Δ at League .0001 ↓			Δ at League 1e-05 ↓		
		Full	Lora	Backpack	Full	Lora	Backpack	Full	Lora	Backpack
Country	3.95	**-0.16**	-0.10	0.10	-0.02	**-0.09**	-0.02	-0.00	**-0.02**	-0.01
Company	10.38	**-0.49**	-0.19	0.35	-0.07	**-0.24**	-0.00	0.00	**-0.00**	0.00
Stereoset	40.13	0.62	**0.18**	8.45	**0.03**	0.13	0.73	0.01	**-0.00**	0.00
Verb	47.00	-0.02	**-0.10**	4.83	-0.00	**-0.01**	2.45	0.01	**-0.02**	0.02
Gender	1.60	0.05	0.02	**0.00**	0.01	0.01	**0.00**	**-0.00**	0.00	0.00
Temporal	4.16	0.01	**-0.00**	0.01	0.00	**-0.00**	0.01	0.00	**0.00**	0.01
Average	17.87	0.00	**-0.03**	2.29	-0.01	**-0.03**	0.53	0.00	**-0.01**	0.00

Table 6: GPT-J hard negatives results.

Split	Task	# Train	Avg Length Train	# Eval	Avg Length Eval
Val	Country	119	9.58	582	111.47
	Company	86	11.07	421	36.52
	Gender	20	4.25	320	11.69
	Verb	240	5.44	360	8.54
	Stereoset	1053	8.64	1053	7.89
	Temporal	75	137.37	452	87.86
Test	Country	119	9.74	583	109.61
	Company	86	11.60	403	36.70
	Gender	20	4.40	360	10.73
	Verb	240	5.38	360	8.54
	Stereoset	1053	8.64	1053	8.02
	Temporal	76	137.42	486	99.67

Table 7: Number of examples, and average token counts, in the train and evaluation splits of our datasets.

D Hyperparameter sweeps

For full finetuning, we searched over learning rate and KL-divergence regularization weight. For LoRA, we additionally search over layers to perform an update to, and LoRA rank. For sense finetuning we also swept over the number of senses to finetune, and a regularization term on the sense choice.

Full finetuning. We sample the learning rate from $10^{-U[4,8.5]}$. We sample the KL-divergence regularization term from $10^{U[-1,0]}$.

LoRA finetuning. We sample the learning rate from $10^{-U[2,6.5]}$. We sample the KL-divergence regularization term from $10^{U[-1,0]}$. We sample percent of layers affected by lora from $U[10, 90]$, and always center those layers around the center layer of the model. We sample the LoRA rank from $U\{1, \dots, 256\}$.

Sense finetuning. We sample the learning rate from $10^{-U[1.5,4]}$. We sample the KL-divergence regularization term from $10^{U[-1,0]}$. We sample the number of senses to finetune from $U\{5, \dots, 12\}$. From early experiments, we set the sense selection regularization hyperparameter $\lambda = 1000$.

E Further dataset details

E.1 Dataset size details

Details on the size of each dataset, including average token counts under the GPT-2 tokenizer, are found in Table 7.

E.2 Prompts for generative models

All data generation was performed with `gpt-3.5-turbo` or `gpt-4`.

E.2.1 Generalization set E

Country Generating the canonical example statements of country-capital cities (to get some extra fluency in edge cases.)

```
Please generate a statement that the capital of {} is {}.Be fluent,
adding or removing 'the' as necessary. Generate it as a python
string, with absolutely no other markup or commentary.
```

Generating paragraphs eliciting the capital of the country:

```
Please generate a varied, interesting paragraph that (1)
first mentions the name of the country in the sentence below,
and then (2) later, brings up the idea of the country's capital,
and then (3) says the name of the capital. It should be natural,
but rather clear that the capital is about to be mentioned. Here
is the statement from which to pull the capital and country: {}.
```

we generate five such paragraphs in the same context; after each one, all previous paragraphs are conditioned on, along with the following intermediary prompt:

```
Great; please generate another one with varied structure,
ensuring that the prefix before the first time that the capital
is mentioned clearly indicates that the capital is about to
be mentioned.
```

Company For generating a paragraph about company-CEO relationship:

```
Please generate a varied, interesting paragraph that (1) first mentions
the name of the company in the sentence below, and then (2) later,
brings up the idea of the company's CEO, and then (3) says the name
of the CEO. It should be natural, but rather clear that the CEO is
about to be mentioned. Here is the statement from which to pull the
CEO and company: [country]
```

we generate five such paragraphs in the same context; after each one, all previous paragraphs are conditioned on, along with the following intermediary prompt:

```
Great; please generate another one with varied structure, ensuring
that the prefix before the first time that the CEO is mentioned
clearly indicates that the CEO is about to be mentioned.
```

Gender Bias We paraphrased some of the evaluation prompts of Hewitt et al. (2023) with the following:

```
Please generate a short paraphrase of this fragment. It's critical
that the paraphrase be continuable by a pronoun like 'he', 'she',
or 'they'. It's also critical that the [career] token is maintained
identically. Do not use a pronoun in the prefix. Be creative.
Here's the prefix: '{}'
```

Stereoset Not used.

Verb To generate a semantically coherent disambiguating sentence from a prefix:

```
Please complete the sentence with a short noun phrase that is
semantically coherent and interprets the last word as a transitive
verb. Ensure the transitive verb is not part of a multi-verb phrase.
The noun phrase should be the object of the verb. At most 6 words.
Only generate the completion; do not generate the whole input
sentence. The verb is {}; make sure it's interpreted as a verb
in the sentence.
```

Temporal To generate a short description of an entity:

```
lease generate a varied, interesting paragraph that (1) first mentions
the name of the person/company/entity/idea/concept mentioned below,
and then (2) discusses the concept and things relevant to it in a
short paragraph. It should be natural, informational, factual.
Here is the relevant entity: {}.\n\nNow, generate just your resulting
paragraph, with no additional discussion.
```

E.2.2 Hard negative set H

Country

```
A well known city in {country} is {other_city}.
Here's a fact about it: {fact}
Please generate a varied, interesting sentence that
(1) first mentions the name of the country and then
(2) mentions the fact about the aforementioned city
in the same sentence. However, it's extremely
important that the fact be mentioned before the city
name {other_city} is mentioned, and it should be
natural, but rather clear that the city {other_city}
is about to be mentioned. Generate only the sentence
and nothing else. The provided fact might mention the
capital city of the country in addition to {other_city},
but you should mention {other_city} only.

For example, for Afghanistan's city Herat, here is a fact about
it: Herat is home to the Great Mosque of Herat (Jama Masjid),
a grand example of Islamic architecture from the Timurid period.
An example output is:
Afghanistan boasts Islamic architecture from the Timurid period.
A grand example is the Great Mosque of Herat (Jama Masjid), located
in the city of Herat.

Note how the fact about Herat, i.e. the the Great Mosque, is
mentioned before the city of Herat is mentioned in the same
sentence. You should make sure your sentence has the same
structure.
```

As a heuristic validation:

```
The capital of {country} is {capital}. Using the output format
below, generate a well known fact about a well known city in
this country that is NOT the capital. This fact should be true
only of this other city, and not true of the capital city.
Examples are landmarks in this other city or historical events
that happened in this city. Explictly think about what is not
true of the capital city {capital} but true of this other
city in {country}
```

And for generating

Company Same as evaluation set, with different entities.

Gender Bias To generate a story about a person who explicitly uses a set of pronouns:

```
Please write an interesting and relatively short sentence about
a {job} who uses the pronouns "{pronouns}". A pronoun should
appear at least once, but not at the beginning of the sentence.
Explicitly mention the person is a {job}. Stay away from
stereotypes about people who use the pronouns {pronouns}.
```

Stereoset For words/phrases not found in the dictionary, we elicited a short definition with the following:

```
Please generate a short definition for this word. If there’s
a typo, figure out what the word should be but don’t mention it.
The word is {}. Do not add any words like ’the definition of...
is’; instead just write the definition; e.g., for ’manager’,
’someone who controls resources and expenditures’.
Do not titlecase the first word
```

Verb To generate a semantically coherent sentence with a given subject to test whether the verbs in the canonical examples can still also be used as nouns:

```
Please generate a short, semantically coherent sentence with
the following subject: {}
```

and similarly for the nouns that showed up in the canonical example set:

```
Please generate a short, semantically coherent sentence with
the following word: {}
```

Temporal Same as evaluation set, with different entities.