A Mathematical Exploration of Why Language Models Help Solve Downstream Tasks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Autoregressive language models pretrained on large corpora have been successful at solving downstream tasks, even with zero-shot usage. However, there is little theoretical justification for their success. This paper considers the following questions: (1) Why should learning the distribution of natural language help with downstream classification tasks? (2) Why do features learned using language modeling help solve downstream tasks with *linear classifiers*? For (1), we hypothesize, and verify empirically, that classification tasks of interest can be reformulated as next word prediction tasks, thus making language modeling a meaningful pretraining task. For (2), we analyze properties of the cross-entropy objective to show that ϵ-optimal language models in cross-entropy (log-perplexity) learn features that are $\mathcal{O}(\sqrt{\epsilon})$-good on natural linear classification tasks, thus demonstrating mathematically that doing well on language modeling can be beneficial for downstream tasks. We perform experiments to verify assumptions and validate theoretical results. Our theoretical insights motivate a simple alternative to the cross-entropy objective that performs well on some linear classification tasks.

1 Introduction

The construction of increasingly powerful language models that use gigantic text corpora and a cross-entropy objective to predict a distribution over the next word after a given context, has revolutionized natural language processing (NLP). The learned representations are useful for many other tasks, either as initializations [Ramachandran et al., 2017, Howard and Ruder, 2018] or as a source of contextual word embeddings [McCann et al., 2017, Peters et al., 2018]. Although representations previously needed fine-tuning to solve downstream tasks, recent models [Radford et al., 2019, Brown et al., 2020] have demonstrated strong performance even without fine-tuning.

Next word prediction being a powerful test of language understanding, intuitively it is believable that language modeling can help with downstream tasks. It is still intriguing how even small decreases in test perplexity can lead to improved downstream performance. Though inductive biases of models and algorithms play a role in this success, it is very challenging to say anything mathematically precise about this, given the nascency of deep learning theory. Instead we take a stab at the mathematical study of why solving the next-word prediction should intrinsically learn representations useful for downstream tasks. As a first cut analysis, we restrict attention to *classification tasks* and the striking observation that they can be solved fairly well using linear classifiers on language model features without fine-tuning. Although we are forced to treat models as black boxes, just first-order optimality conditions reveal interesting properties of the learned features, leading to an understanding of their success on interesting linear classification tasks. We now summarize our contributions.

With the observation that classification tasks of interest can be phrased as sentence completion tasks, we define *natural classification tasks*, in Section 3, as those that can be solved as *linear functions* of the conditional distribution over words that can follow a given context. Section 4 presents our main results, theorems 4.1 and 4.2, that mathematically quantify the benefit of language model features for solving natural tasks. We show that a ϵ-optimal language model (in cross-entropy) will do $\mathcal{O}(\sqrt{\epsilon})$-well on natural tasks. Theorem 4.2 proves a stronger result for low dimensional softmax models using a new tool that we call *conditional mean features* (Definition C.1), which empirically (Section 5) does well. We construct a new mathematically motivated objective Quad with provable guarantees (Section 4); we report its good performance on linear classification tasks (Section 5).

2 Language modeling and optimal solutions

We use $\mathcal{S}$ to denote the set of contexts (sentences and partial sentences), $\mathcal{W}$ to denote the set of words, with $V = |\mathcal{W}|$. Δ_A denotes the set of all distributions on A. $p_{\cdot|s}, p^*_{\cdot|s} \in \Delta_{\mathcal{W}}$ denote distributions over next word given a context s; we also use $p_{\cdot|s}, p^*_{\cdot|s} \in \mathbb{R}^V$ also as vectors of probabilities. $\phi_w \in \mathbb{R}^d$ is a d-dimensional embedding for word $w \in \mathcal{W}$. Embeddings are stacked into a matrix $\Phi \in \mathbb{R}^{d\times V}$.

Unconstrained language modeling using cross-entropy: The goal is to learn the true distribution of a text corpus by predicting $p_{\cdot|s}$ for a context $s \in \mathcal{S}$ (e.g. $s =$"The food was ", $p_{\cdot|s}$ is supported on "delicious", "expensive", etc.). Let p_L be the distribution over $\mathcal{S}$ in the corpus and $p^*_{\cdot|s}$ be the true conditional distribution. It is standard to minimize the expected cross-entropy between $p^*_{\cdot|s}$ and $p_{\cdot|s}$.

$$\ell_{\text{xent}}(\{p_{\cdot|s}\}) = \mathop{\mathbb{E}}_{s\sim p_L}\left[\ell_{\text{xent},s}(p_{\cdot|s})\right] = \mathop{\mathbb{E}}_{s\sim p_L}\mathop{\mathbb{E}}_{w\sim p^*_{\cdot|s}}\left[-\log(p_{\cdot|s}(w))\right] \tag{1}$$

Using $\ell_{\text{xent},s}(p_{\cdot|s}) - \ell_{\text{xent}}(p^*_{\cdot|s}) = D_{\text{KL}}(p^*_{\cdot|s}, p_{\cdot|s})$, it is easy to show that this recovers $p^*_{\cdot|s}$ exactly.

Proposition 2.1. *The unique minimizer of $\ell_{xent}(\{p_{\cdot|s}\})$ is $p_{\cdot|s} = p^*_{\cdot|s}$ for every $s \sim p_L$.*

Softmax parametrized language modeling: Recent models parametrize $p_{\cdot|s}$ as a softmax computed using *low dimensional* embeddings $f(s) \in \mathbb{R}^d$, where f is the output of a model architecture of choice. These embeddings then induce the softmax distribution $p_{\cdot|s} = p_{f(s)}$ with $p_{f(s)}(w) = e^{f(s)^\top\phi_w}/Z_{f(s)}$, where $Z_\theta = \sum_{w'\in\mathcal{W}} e^{\theta^\top\phi_{w'}}$. We can now rewrite the cross-entropy for this setting

$$\ell_{\text{xent}}(f,\Phi) = \mathop{\mathbb{E}}_{s\sim p_L}\left[\ell_{\text{xent},s}(p_{f(s)})\right] = \mathop{\mathbb{E}}_{s\sim p_L}\left[\mathop{\mathbb{E}}_{w\sim p^*_{\cdot|s}}[-f(s)^\top\phi_w] + \log(Z_{f(s)})\right] \tag{2}$$

We define $\ell_{\text{xent},s}(\theta,\Phi) = \ell_{\text{xent},s}(p_\theta)$. Analogous to Proposition 2.1, we want to know the optimal d-dimensional feature map f^* and the induced conditional distribution $p_{f^*(s)}$.

Proposition 2.2. *For a fixed Φ, if $f^* \in \mathop{\arg\min}_{f:\mathcal{S}\to\mathbb{R}^d} \ell_{xent}(f,\Phi)$, then $\Phi p_{f^*(s)} = \Phi p^*_{\cdot|s}$ for every $s \sim p_L$.*

Unlike Proposition 2.1, $p_{f^*(s)} \in \mathbb{R}^V$ is only equal to $p^*_{\cdot|s} \in \mathbb{R}^V$ in the subspace of the rows of $\Phi \in \mathbb{R}^{d\times V}$. Thus smaller values of d will guarantee learning $p^*_{\cdot|s}$ on smaller subspaces.

Downstream classification tasks: We focus on binary classification tasks. Task $\mathcal{T}$ is a distribution $p_{\mathcal{T}}$ over $\mathcal{S}\times\{\pm1\}$, with input sentence $s \in \mathcal{S}$ and the label $y \in \{\pm1\}$. Given a feature map $g : \mathcal{S} \to \mathbb{R}^D$ (for any D), we solve task $\mathcal{T}$ by fitting a linear classifier $\boldsymbol{v} \in \mathbb{R}^D$ on top of $g(s)$. The classification loss is written as $\ell_{\mathcal{T}}(g,\boldsymbol{v}) = \mathbb{E}_{(s,y)\sim p_{\mathcal{T}}}\left[\ell(\boldsymbol{v}^\top g(s), y)\right]$, where ℓ is a 1-Lipschitz surrogate to the 0-1 loss [1]. The loss incurred by a representation function g is defined as $\ell_{\mathcal{T}}(g) = \inf_{\boldsymbol{v}\in\mathbb{R}^D} \ell_{\mathcal{T}}(g,\boldsymbol{v})$. For embeddings $\{\theta_s\}_{s\in\mathcal{S}}$, classification loss is thus $\ell_{\mathcal{T}}(\{\theta_s\},\boldsymbol{v}) = \mathbb{E}_{(s,y)\sim p_{\mathcal{T}}}[\ell(\boldsymbol{v}^\top\theta_s, y)]$.

3 Using language models for classification tasks

Section 2 shows that both unconstrained and softmax language models aim to learn $p^*_{\cdot|s}$ or a projection $\Phi p^*_{\cdot|s}$. Thus we ask: why should $p^*_{\cdot|s}$ help with downstream tasks? We argue that classification tasks can be reframed as sentence completion problems and $p^*_{\cdot|s}$ can give completions to predict the label.

[1] hinge $\ell(\hat{y}, y) = (1 - y\hat{y})_+$ or the logistic $\ell(\hat{y}, y) = \log(1 + e^{-y\hat{y}})$. Extending to k-way tasks is easy.

In particular, for a movie review sentiment analysis task, we can compare probabilities of “:)” and “:(” after an input and predict sentiment based on which is higher. We can view this as learning a linear classifier $\boldsymbol{v}$ over $p^*_{\cdot|s} \in \mathbb{R}^V$ to solve the task, where $\boldsymbol{v}[\text{“:)”}] = 1$ and $\boldsymbol{v}[\text{“:(”}] = -1$. Since $p^*_{\cdot|s}$ will place higher probability on words like “The” that are not useful for sentiment, we can also add a prompt like “This movie is ” and query probabilities of useful adjectives like “good”, “bad”, etc. This is also a linear classifier with positive and negative weights assigned to respective adjectives. This approach also works for AG news dataset [Zhang et al., 2015] with a prompt like “This article is about ’. We verify experimentally that SST and AG news tasks can be solved by a linear function of probabilities of just a small subset of words in Section 5. We can further formalize this intuition.

Definition 3.1. *A classification task $\mathcal{T}$ is (τ, B)-natural if* $\min\limits_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \le B} \ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v}) \le \tau$.

Thus a natural task $\mathcal{T}$ is one that can achieve a small error τ by learning an ℓ_∞-norm bounded[2] linear classifier on top of features $p^*_{\cdot|s} \in \mathbb{R}^V$. Low dimensional softmax models, however, only learn $p^*_{\cdot|s}$ in the subspace of Φ (see Proposition 2.2). Thus we are interested in tasks this subspace can solve.

Definition 3.2. *Task $\mathcal{T}$ is (τ, B)-natural w.r.t. $\Phi \in \mathbb{R}^{d\times V}$ if* $\min\limits_{\boldsymbol{v}\in\textit{row-span}(\Phi), \|\boldsymbol{v}\|_\infty \le B} \ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v}) \le \tau$.

This subset of tasks is not too restrictive if Φ assigns similar embeddings to synonyms. Section 4 describes a carefully designed objective that can provably learn such embeddings.

4 Guarantees for language models on natural tasks (overview)

We present simplified versions our main results that show that ϵ-optimal language models will have a loss of $\tau + \mathcal{O}(B\sqrt{\epsilon})$ on (τ, B)-natural classification tasks (see Section C for detailed results), formalizing the intuition that better language models are more useful for downstream tasks. We first define optimal cross entropies.

$$\ell^*_{\text{xent}} = \ell_{\text{xent}}(\{p^*_{\cdot|s}\}),\;\; \ell^*_{\text{xent}}(\Phi) = \mathop{\mathbb{E}}_{s\sim p_L}\left[\inf_{\theta\in\mathbb{R}^d} \ell_{\text{xent},s}(\theta, \Phi)\right] \tag{3}$$

where ℓ^*_{xent} is the absolute minimum achievable cross entropy, while $\ell^*_{\text{xent}}(\Phi)$ is the minimum achievable cross entropy by a d-dimensional softmax language model that uses Φ; clearly $\ell^*_{\text{xent}} \le \ell^*_{\text{xent}}(\Phi)$. We first present the simplified result for unconstrained language models.

Theorem 4.1. *[Simplified, unconstrained] Let $\{p_{\cdot|s}\}$ be a language model that is ϵ-optimal, i.e. $\ell_{xent}(\{p_{\cdot|s}\}) - \ell^*_{xent} \le \epsilon$, for some $\epsilon > 0$. For a classification task $\mathcal{T}$ that is (τ, B)-natural, we have*

$$\ell_{\mathcal{T}}\left(\{p_{\cdot|s}\}\right) \le \tau + \mathcal{O}(B\sqrt{\epsilon})$$

This bound applies to all language models, (n-gram models, softmax models), and suggests that smaller test cross-entropy is desirable to guarantee good downstream. The suboptimality ϵ propagates gracefully as $\mathcal{O}(\sqrt{\epsilon})$ to the task $\mathcal{T}$. While τ is some small constant like 0.01, the norm bound B captures the difficulty of task $\mathcal{T}$ when solved linearly using $\{p^*_{\cdot|s}\}$. Intuitively, if the words in support of the classifier (Section 3), have total probability mass of $\Omega(\alpha)$ in $p^*_{\cdot|s}$, then $B \sim \mathcal{O}(1/\alpha)$. It is thus desirable for a task $\mathcal{T}$ to depend on a larger and more frequent set of words. Adding a prompt, as in Section 3, can increase the set of indicative words, thus decreasing B.

For d-dimensional softmax language models, inspired by Proposition 2.2, we show guarantees for *conditional mean features* $g_{f,\Phi}(s) = \Phi p_{f(s)} \in \mathbb{R}^d$. This gives a way to use softmax language model features for linear classification; we test the success of this mechanism experimentally in Section 5.

Theorem 4.2. *[Simplified, softmax] Let f be an ϵ-optimal d-dimensional softmax language model, i.e. $\ell_{xent}(f, \Phi) - \ell^*_{xent}(\Phi) \le \epsilon$. For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have*

$$\ell_{\mathcal{T}}\left(g_{f,\Phi}\right) \le \tau + \mathcal{O}(B\sqrt{\epsilon})$$

The key differences from Theorem 4.1 are, (1) ϵ captures the suboptimality of learned language model compared $\ell^*_{\text{xent}}(\Phi)$ and not ℓ^*_{xent}, (2) guarantees are only for natural tasks w.r.t. Φ. Theorem 4.1 can also

[2] Makes sense since $\|p^*_{\cdot|s}\|_1 = 1$ & $\|\cdot\|_\infty$ is dual norm of $\|\cdot\|_1$. See Theorem 4.1 for an interpretation of B

Table 1: Pretrained GPT-2 performance on linear classification tasks using features $f(s)$, $p_{f(s)}$ and $g_{f,\Phi}(s)$. An asterisk indicates that we added a task-specific prompt.

Task	Features $f(s)$	$g_{f,\Phi}(s) = \Phi p_{f(s)}$	$p_{f(s)}$ over subset	$p_{f(s)}$ over class words
SST	87.6%	82.6%	78.2%	76.4%
SST*	89.5%	87.0%	83.5%	79.4%
AG news	90.7%	84.5%	78.3%	68.4%
AG news*	91.1%	88.0%	83.0%	71.4%

be invoked to get an upper bound of $\mathcal{O}(B\sqrt{\epsilon + \epsilon^*_{\Phi}})$ instead of $\mathcal{O}(B\sqrt{\epsilon})$, where $\epsilon^*_{\Phi} = \ell^*_{\text{xent}}(\Phi) - \ell^*_{\text{xent}}$. Thus Theorem 4.2 is stronger since models only need optimality w.r.t. the best d-dimensional model. This improvement to $\mathcal{O}(B\sqrt{\epsilon})$ comes at the cost of only having guarantees for natural tasks w.r.t. Φ.

$g_{f,\Phi}(s)$ is a linear function of $f(s)$: While Theorem 4.2 shows that $g_{f,\Phi}$ is useful for linear classification, using f directly is more standard and performs well in practice (Section 5). We show a linear relation between f and $g_{f,\Phi}$ if word embeddings Φ satisfy the following property.

Assumption 4.1. *There exists a symmetric positive semidefinite matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, a vector $\boldsymbol{b} \in \mathbb{R}^d$ and a constant $c \in \mathbb{R}$ such that $\log(Z_\theta) = \frac{1}{2}\theta^\top \boldsymbol{A}\theta + \theta^\top \boldsymbol{b} + c$ for any $\theta \in \mathbb{R}^d$.*

If word embeddings were distributed as Gaussians, i.e. $\phi_w \sim \mathcal{N}(\mu, \Sigma)$ independently, it is not hard to show (Lemma F.5) that $\log(Z_\theta) \approx \frac{1}{2}\theta^\top\Sigma\theta + \theta^\top\mu + \log(V)$. Empirically we find the fit to be very good, as evident in Figure 1. Under the above assumption, we can show a linear relation between f and $g_{f,\Phi}$ and thus good performance of f on natural tasks. See detailed discussion in Section C.3.

Lemma 4.3. *Under Assumption 4.1, feature map f satisfies $g_{f,\Phi}(s) = \boldsymbol{A}f(s) + \boldsymbol{b}, \forall s \in \mathcal{S}$.*

Corollary 4.1. *Under same setting as Lemma 4.3 and Theorem 4.2, $\ell_{\mathcal{T}}(f) \leq \tau + \mathcal{O}(B\sqrt{\epsilon})$.*

Quad objective: Word embeddings Φ determine the tasks that softmax language model features can solve, but the cross-entropy objective does not lend a simple closed form expression for optimal Φ. This motivates our Quad objective with two nice properties: (1) optimal feature map f^* can solve some natural tasks, (2) optimal Φ^* has an intuitively meaningful closed-form solution.

$$\ell_{quad}(f, \Phi) = \mathop{\mathbb{E}}_{s\sim p_L}\left[\mathop{\mathbb{E}}_{w\sim p^*_{\cdot|s}}[-f(s)^\top\phi_w] + \frac{1}{2}\|\Phi^\top f(s)\|^2\right] \tag{4}$$

The Quad objective is very similar to the cross-entropy objective from Equation (2), with the log partition function $\log(Z_{f(s)})$ replaced with a quadratic function $\frac{1}{2}\|\Phi^\top f(s)\|^2$, inspired in part by Assumption 4.1. The optimal solution Φ^* depends on the eigendirections of a *substitutability matrix*.

Definition 4.1. *The substitutability matrix is defined to be $\Omega^* := \mathop{\mathbb{E}}_{s\sim p_L}\left[p^*_{\cdot|s}\, {p^*_{\cdot|s}}^\top\right] \in \mathbb{R}^{V\times V}$. If $\Omega^* = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ is the eigendecomposition, then $\boldsymbol{U}_d \in \mathbb{R}^{V\times d}$ is matrix of top d eigenvectors of Ω^*.*

Theorem 4.4. *Let $f^*, \Phi^* = \arg\min_{f,\Phi} \ell_{quad}(f, \Phi)$. Then $\Phi^* = \boldsymbol{B}\boldsymbol{U}_d^\top$, for full rank $\boldsymbol{B} \in \mathbb{R}^{d\times d}$. Also, for a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ^*, we have $\ell_{\mathcal{T}}(f^*) \leq \tau$.*

Thus f^* excels on natural tasks w.r.t. Φ^*, the best d-dimensional projection of Ω^*. Please refer to Section E for an interpretation of Φ^* relating it to synonyms. We train with Quad objective and compare to a similarly trained language model (Section G.2), finding Quad to be reasonably effective. The goal is not to obtain state-of-the-art results, but show the practical utility of theoretical insights.

5 Experiments

We validate our claims from Section 3 that classification tasks can be solved by linear functions of $p_{\cdot|s}$. Table 1 demonstrates that on SST [Socher et al., 2013] and AG News tasks we can use $p_{\cdot|s} = p_{f(s)}$ from GPT-2 [Radford et al., 2019] of just 20 task-relevant tokens (Section G.1) to solve tasks. Even just one token per class yields non-trivial performance. We validate the complete-the-sentence intuition by observing improved performance after adding a task specific prompt. We validate Theorem 4.2 by verifying that the conditional mean features $g_{f,\Phi}(s) = \Phi p_{f(s)}$ also linearly solve downstream tasks fairly well. Section G.1 has results for a wider range of classification tasks. Evidence for Assumption 4.1 is provided in Section G.3.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 2016.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the International Conference on Learning Representations*, 2017.

Sanjeev Arora, Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. A compressed sensing view of unsupervised text embeddings, bag-of-n-grams, and LSTMs. In *Proceedings of the International Conference on Learning Representations*, 2018.

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, 2007.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 2003.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Stanley F Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13, 1999.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Chris Dyer. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*, 2014.

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

Zellig Harris. Distributional structure. *Word*, 1954.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. 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)*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, 2015.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Jason D. Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: provable self-supervised learning. *arXiv preprint arXiv:2008.01064*, 2020.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, 2014.

Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, 2002.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *Proceedings of the International Conference on Learning Representations*, 2018.

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies*, 2011.

Julian J. McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. *CoRR*, 2015.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, 2017.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 2013b.

Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *Proceedings of the International Conference on Learning Representations*, 2018.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. Proceedings of the North American Chapter of the ACL: Human Language Technologies, 2018.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Raul Puri and Bryan Catanzaro. Zero-shot text classification with generative language models. *arXiv prepring arXiv:1912.10165*, 2019.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Prajit Ramachandran, Peter Liu, and Quoc Le. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*, 2020.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. olmpics–on what language model pre-training captures. *arXiv preprint arXiv:1912.13283*, 2019.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. *arXiv preprint arXiv:2008.10150*, 2020a.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive estimation reveals topic posterior information to linear models. *arXiv preprint arXiv:2003.02234*, 2020b.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *arXiv preprint arXiv:2005.10242*, 2020.

Theresa Wilson and Janyce Wiebe. Annotating opinions in the world press. In *Proceedings of the Fourth SIGdial Workshop of Discourse and Dialogue*, 2003.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Wei Xu and Alex Rudnicky. Can artificial neural networks learn language models? In *Sixth international conference on spoken language processing*, 2000.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*. 2015.

A Overview

Section B discusses related work. Section C is a detailed version of Section 4 with the full statements of Theorems 4.1 and 4.2. Section D describes stronger versions of Theorems 4.1 and 4.2. Section E is a detailed version of Section 4. Section F contains proofs for all results. Section G contains many more experimental findings that consolidate many of our theoretical results. Section G.1 provides the information about subsets of words used for results in Table 1 and also additional experiments to test the performance of pretrained language model embeddings f on more downstream tasks and also verifying that conditional mean embeddings Φp_f do well on these tasks. In Section G.2, we present additional results for our Quad objective trained on a larger corpus and tested on SST. Finally Section G.3 provides additional details on how $\boldsymbol{A}$, $\boldsymbol{b}$ and c from Assumption 4.1 are learned and also further verification of the assumption on more datasets.

B Related work

Embedding methods: Prior to language models, large text corpora like Wikipedia [Merity et al., 2016] were used to learn low dimensional embeddings for words [Mikolov et al., 2013b,a, Pennington et al., 2014] and subsequently for sentences [Kiros et al., 2015, Arora et al., 2017, Pagliardini et al., 2018, Logeswaran and Lee, 2018] for downstream task usage. These methods were inspired by the distributional hypothesis [Firth, 1957, Harris, 1954], which posits that meaning of text is determined in part by the surrounding context. Recent methods like BERT [Devlin et al., 2018] and variants [Lan et al., 2019, Yang et al., 2019, Liu et al., 2019] learn from auxiliary tasks, such as sentence completion, and are among the top performers on downstream tasks.

Language models for downstream tasks: We are interested in language models, including n-gram models [Chen and Goodman, 1999], and more recent models [Xu and Rudnicky, 2000, Bengio et al., 2003] that use neural networks to compute low dimensional features for contexts and parametrize the next word distribution using softmax. Language models have shown to be useful for downstream tasks as initializations [Ramachandran et al., 2017, Howard and Ruder, 2018] or as learned feature maps [Radford et al., 2017, McCann et al., 2017, Peters et al., 2018]. The idea of phrasing classification tasks as sentence completion problems is motivated by recent works [Radford et al., 2019, Puri and Catanzaro, 2019, Schick and Schütze, 2020] that show that many downstream tasks can be solved by next word prediction for an appropriately conditioned language model. This idea also shares similarities with work that phrase a suite of downstream tasks as question-answering tasks [McCann et al., 2018] or text-to-text tasks [Raffel et al., 2019] and symbolic reasoning as fill-in-the-blank tasks [Talmor et al., 2019]. Our work exploits this prevalent idea of task rephrasing to theoretically analyze why language models succeed on downstream tasks.

Theoretical analysis: Since the success of early word embedding algorithms like word2vec [Mikolov et al., 2013a] and GloVe [Pennington et al., 2014], there have been attempts to understand them theoretically. Levy and Goldberg [2014] show that, in some regimes, the word2vec algorithm implicitly factorizes the PMI matrix. The theory of Noise Contrastive Estimation (NCE) is used by Dyer [2014] to understand word embedding methods and by Ma and Collins [2018] to prove parameter recovery for negative sampling methods that learn conditional models. A latent variable log-linear model is proposed in Arora et al. [2016] to explain and unify various word embedding algorithms. Theoretical justification is provided for sentence embedding methods either by using a latent variable model [Arora et al., 2017] or through the lens of compressed sensing [Arora et al., 2018]. Also relevant is recent work on theory for contrastive learning [Arora et al., 2019, Tosh et al., 2020b,a, Wang and Isola, 2020] and reconstruction-based methods [Lee et al., 2020], which analyze the utility of representations learned in the self-supervised regime for downstream tasks. Our work is the first to analyze the efficacy language model features on downstream tasks.

C Guarantees for language models on natural tasks

We now show guarantees for features from language models on natural tasks. For an unconstrained model, we use the learned $p_{\cdot|s} \in \mathbb{R}^V$ as features and for softmax model f, we show guarantees for $\Phi p_{f(s)} \in \mathbb{R}^d$. Since we cannot practically hope to learn the optimal solutions described in Propositions 2.1 and 2.2 , we only assume that the language models are ϵ-optimal in cross-entropy.

To define ϵ-optimality in the two settings, we first define optimal cross-entropies.

$$\ell^*_{\text{xent}} = \ell_{\text{xent}}(\{p^*_{\cdot|s}\}), \;\; \ell^*_{\text{xent}}(\Phi) = \mathop{\mathbb{E}}_{s\sim p_L}\left[\inf_{\theta\in\mathbb{R}^d} \ell_{\text{xent},s}(\theta,\Phi)\right] \tag{5}$$

where ℓ^*_{xent} is the absolute minimum achievable cross-entropy, while $\ell^*_{\text{xent}}(\Phi)$ is the minimum achievable cross-entropy by a d-dimensional softmax language model using Φ; clearly $\ell^*_{\text{xent}} \leq \ell^*_{\text{xent}}(\Phi)$.

C.1 Unconstrained language models

We show guarantees for a language model that satisfies $\ell_{\text{xent}}(\{p_{\cdot|s}\}) - \ell^*_{\text{xent}} \leq \epsilon$. An important consideration is that the language model distribution p_L of contexts is often a diverse superset of the downstream distribution $p_{\mathcal{T}}$ (defined in Section 2), thus requiring us to show how guarantees of $p_{\cdot|s} \approx p^*_{\cdot|s}$ *on average* over the distribution $s \sim p_L$ transfer to guarantees on a subset $p_{\mathcal{T}}$. In the worst case, all of the ϵ error by $\{p_{\cdot|s}\}$ occurs on sentences from the subset $p_{\mathcal{T}}$, leading to pessimistic bounds[3]. In practice, however, the errors might be more evenly distributed across p_L, thus bypassing this worst case bound. As a first step, we present the worst case bound here; stronger guarantees are in Section D. The worst-case coefficient $\gamma(p_{\mathcal{T}})$, defined below, captures that $p_{\mathcal{T}}$ is a $\gamma(p_{\mathcal{T}})$-fraction of p_L.

$$\gamma(p_{\mathcal{T}}) = \max\{\gamma \in \mathbb{R} : p_L(s) \geq \gamma p_{\mathcal{T}}(s)\ \forall s \in \mathcal{S}\} \tag{6}$$

We now present our results that applies to any language model, regardless of the parametrization (e.g., n-gram models, softmax models). The result suggests that small test cross-entropy (hence test perplexity) is desirable to guarantee good classification performance, thus formalizing the intuition that better language models should do better on downstream tasks.

Theorem 4.1. *Let $\{p_{\cdot|s}\}$ be a language model that is ϵ-optimal, i.e. $\ell_{xent}(\{p_{\cdot|s}\}) - \ell^*_{xent} \leq \epsilon$, for some $\epsilon > 0$. For a classification task $\mathcal{T}$ that is (τ, B)-natural, we have*

$$\ell_{\mathcal{T}}\left(\{p_{\cdot|s}\}\right) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}})}}$$

Discussion: The suboptimality ϵ propagates gracefully as $\mathcal{O}(\sqrt{\epsilon})$ to a downstream task. While τ can be thought of as a small constant like 0.01, the norm bound B captures the margin of task $\mathcal{T}$ when solved linearly using $\{p^*_{\cdot|s}\}$. Intuitively, for $\ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v})$ to be smaller than τ, B needs to be large enough so that $\boldsymbol{v}^\top p^*_{\cdot|s} = \Omega(1)$. Thus if the words of interest in the support of $\boldsymbol{v}$, described in Section 3, have total probability mass of $\Omega(\alpha)$ in $p^*_{\cdot|s}$, then $B \sim \mathcal{O}(1/\alpha)$. It is thus desirable for a task $\mathcal{T}$ to depend on a larger and more frequent set of words. A task that depends on probabilities of rare words will have a high value of B. Adding a prompt, as described in Section 3, can broaden the set of indicative words, thus potentially decreasing B. A key step in the proof is to bound the difference in prediction on s for classifier $\boldsymbol{v}$ as $|\boldsymbol{v}^\top(p_{\cdot|s} - p^*_{\cdot|s})| \leq \|\boldsymbol{v}\|_\infty \|p_{\cdot|s} - p^*_{\cdot|s}\|_1 \leq \|\boldsymbol{v}\|_\infty \sqrt{2(\ell_{\text{xent},s}(p_{\cdot|s}) - \ell_{\text{xent},s}(p^*_{\cdot|s}))}$, using Holder's and Pinsker's inequalities respectively.

C.2 Softmax language model with conditional mean features

We now describe guarantees for a softmax language model with feature map f that satisfies $\ell_{\text{xent}}(f,\Phi) - \ell^*_{\text{xent}}(\Phi) \leq \epsilon$; suboptimality is measured w.r.t. the best d-dimensional model, unlike Theorem 4.1. Note that Theorem 4.1 can be applied here to give a bound of $\ell_{\mathcal{T}}(\{p_{f(s)}\}) \leq \tau + \mathcal{O}(B\sqrt{\epsilon + \epsilon^*_\Phi})$ on (τ, B)-natural tasks, where $\epsilon^*_\Phi = \ell^*_{\text{xent}}(\Phi) - \ell^*_{\text{xent}}$ is the suboptimality of the best d-dimensional model. This fixed error of $\mathcal{O}(B\sqrt{\epsilon^*_\Phi})$ (even when $\epsilon = 0$), however, is undesirable. We improve on this by proving a stronger result specifically for softmax models. Inspired by Proposition 2.2 that shows $\Phi p_{f^*(s)} = \Phi p^*_{\cdot|s}$, our guarantees are for features $\Phi p_{f(s)} \in \mathbb{R}^d$ that we call conditional mean features.

Definition C.1 (Conditional Mean Features)**.** *For a feature map $f : \mathcal{S} \to \mathbb{R}^d$ and $\Phi \in \mathbb{R}^{d\times V}$, we define conditional mean features $g_{f,\Phi} : \mathcal{S} \to \mathbb{R}^d$, where $g_{f,\Phi}(s) = \Phi p_{f(s)}$, where $p_{f(s)} \in \mathbb{R}^V$.*

[3] For instance if $p_{\mathcal{T}}$ is 0.001 fraction of p_L, $\{p_{\cdot|s}\}$ could have 1000ϵ error on $p_{\mathcal{T}}$ and 0 error on rest of p_L.

The result below will show that conditional mean features $g_{f,\Phi}$ are guaranteed to do well on natural tasks w.r.t. Φ, thereby suggesting a novel way to use softmax features f for downstream tasks. We also test $g_{f,\Phi}$ on downstream tasks in Section G.1 and find that they perform comparably to f. We now present the result for softmax language models that has the similar implication as Theorem 4.1, but with above-mentioned subtle differences. The proof (Section F.2) is similar to that of Theorem 4.1, but crucially requires showing a d-dimensional version of Pinkser's inequality.

Theorem 4.2. *For a fixed Φ, let f be features from an ϵ-optimal d-dimensional softmax language model, i.e. $\ell_{xent}(f,\Phi) - \ell^*_{xent}(\Phi) \leq \epsilon$. For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ,*

$$\ell_{\mathcal{T}}(g_{f,\Phi}) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}})}}$$

C.3 $g_{f,\Phi}(s)$ is a linear function of $f(s)$

Theorem 4.2 shows that $g_{f,\Phi}$ is useful for linear classification. However, using feature map f directly is more standard and performs well in practice (see Section G.1). Here we argue that there is a linear relation between f and $g_{f,\Phi}$ if word embeddings Φ satisfy a certain property, which we show implies that tasks solvable linearly with $g_{f,\Phi}$ are also solvable linearly using f. Our main assumption about word embeddings Φ is that the logarithm of the partition function $Z_{f(s)}$ is quadratic in $f(s)$.

Assumption 4.1. *There exists a symmetric positive semidefinite matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, a vector $\boldsymbol{b} \in \mathbb{R}^d$ and a constant $c \in \mathbb{R}$ such that $\log(Z_\theta) = \frac{1}{2}\theta^\top \boldsymbol{A}\theta + \theta^\top \boldsymbol{b} + c$ for any $\theta \in \mathbb{R}^d$.*

If word embeddings were distributed as Gaussians, i.e. V columns of Φ are sampled from $\mathcal{N}(\mu, \Sigma)$ independently, it is not hard to show (Lemma F.5) that $\log(Z_\theta) \approx \frac{1}{2}\theta^\top\Sigma\theta + \theta^\top\mu + \log(V)$. While some papers [Arora et al., 2016, Mu and Viswanath, 2018] have noted that word embeddings are fairly random-like in the bulk to argue that the log partition function is constant for all $\|\theta\|_2 = 1$, our quadratic assumption is a bit stronger. However, empirically we find the fit to be very good, as evident in Figure 1. Under the above assumption, we can show a linear relation between f and Φp_f.

Lemma 4.3. *Under Assumption 4.1, feature map f satisfies $g_{f,\Phi}(s) = \boldsymbol{A}f(s) + \boldsymbol{b}, \forall s \in \mathcal{S}$.*

Corollary 4.1. *Under same setting as Lemma 4.3 and Theorem 4.2, $\ell_{\mathcal{T}}(f) \leq \tau + \mathcal{O}(B\sqrt{\epsilon})$.*

Thus we get that f itself is good for natural linear classification tasks. However, in practice, the linearity between f and $g_{f,\Phi}$ holds only approximately when tested on features from the pretrained GPT-2 language model Radford et al. [2018]. The ratio of the residual norm of the best linear map to the norm of f, i.e. $r = \frac{\mathbb{E}_{s\sim p} \|g_{f,\Phi}(s) - \boldsymbol{A}f(s) - \boldsymbol{b}\|^2}{\mathbb{E}_{s\sim p}\|g_{f,\Phi}(s)\|^2}$, is measured for different distributions p ($r = 0$ means perfect fit). These ratios are 0.28 for SST, 0.39 for AG News, and 0.18 for IMDb contexts. This non-trivial linear relationship, although surprising, might not completely explain the success of f. In fact, f almost always performs better than $g_{f,\Phi}$; we leave exploring this to future work.

D Better handling of distributional shift

While the bounds above used $\gamma(p_{\mathcal{T}})$ to transfer from the distribution p_L to $p_{\mathcal{T}}$, we define a more refined notion of transferability here. While $\gamma(p_{\mathcal{T}})$ only depends on p_L and $p_{\mathcal{T}}$, the more refined notions depend also on the learned language model, thus potentially exploiting some inductive biases. We first define the notion of error made in the predicted probabilities by any predictor $p_{\cdot|s}$ as $\Delta_{\{p_{\cdot|s}\}}(s) = p_{\cdot|s} - p^*_{\cdot|s}$. Thus for any softmax language model f we have $\Delta_{\{p_{f(s)}\}}(s) = p_{f(s)} - p^*_{\cdot|s}$. For any distribution $p \in \Delta_S$, we define the covariance[4] of a function $g: \mathcal{S} \to \mathbb{R}^D$ as $\Sigma_p(g) = \mathbb{E}_{s\sim p}\left[g(s)g(s)^\top\right]$. We define 3 coefficients for the results to follow

Definition D.1. *For any distribution $p \in \Delta_S$, we define the following*

$$\gamma(p; \{p_{\cdot|s}\}) \coloneqq \left(\left\|\Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})^{-\frac{1}{2}}\Sigma_p(\Delta_{\{p_{\cdot|s}\}})\Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})^{-\frac{1}{2}}\right\|_2\right)^{-1} \tag{7}$$

[4] This is not exactly the covariance since the mean is not subtracted, all results hold even for the usual covariance.

$$\gamma_\Phi(p; \{p_{\cdot|s}\}) \coloneqq \left(\left\|\Sigma_{p_L}(\Phi\Delta_{\{p_{\cdot|s}\}})^{-\frac{1}{2}}\Sigma_p(\Phi\Delta_{\{p_{\cdot|s}\}})\Sigma_{p_L}(\Phi\Delta_{\{p_{\cdot|s}\}})^{-\frac{1}{2}}\right\|_2\right)^{-1} \tag{8}$$

$$\gamma(p; g_{f,\Phi}) \coloneqq \gamma_\Phi(p; \{p_{f(s)}\}) \tag{9}$$

We notice that $\Sigma_p(\Delta_{\{p_{\cdot|s}\}}) = \mathop{\mathbb{E}}_{s\sim p}\left[(p_{\cdot|s} - p^*_{\cdot|s})(p_{\cdot|s} - p^*_{\cdot|s})^\top\right]$, $\Sigma_p(\Phi\Delta_{\{p_{\cdot|s}\}}) = \Phi\Sigma_p(\Delta_{\{p_{\cdot|s}\}})\Phi^\top$. We are now ready to state the most general results.

Theorem D.1 (Strengthened Theorem 4.1)**.** *Let $\{p_{\cdot|s}\}$ be a language model that is ϵ-optimal, i.e. $\ell_{xent}(\{p_{\cdot|s}\}) - \ell^*_{xent} \leq \epsilon$ for some $\epsilon > 0$. For a classification task $\mathcal{T}$ that is (τ, B)-natural, we have*

$$\ell_{\mathcal{T}}\left(\{p_{\cdot|s}\}\right) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}$$

For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have

$$\ell_{\mathcal{T}}\left(\{p_{\cdot|s}\}\right) \leq \ell_{\mathcal{T}}\left(\{\Phi p_{\cdot|s}\}\right) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}$$

Theorem D.2. *[Strengthened Theorem 4.2] For a fixed Φ, let f be features from an ϵ-optimal d-dimensional softmax language model, i.e. $\ell_{xent}(f, \Phi) - \ell^*_{xent}(\Phi) \leq \epsilon$, where $\ell^*_{xent}(\Phi)$ is defined in Equation (5). For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have*

$$\ell_{\mathcal{T}}\left(\{p_{f(s)}\}\right) \leq \ell_{\mathcal{T}}(g_{f,\Phi}) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; g_{f,\Phi})}}$$

Discussions: It is not hard to show that the coefficients satisfy $\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\}) \geq \gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\}) \geq \gamma(p_{\mathcal{T}})$ and $\gamma(p_{\mathcal{T}}; g_{f,\Phi}) \geq \gamma(p_{\mathcal{T}})$, thus showing that these results are strictly stronger than the ones from the previous section. The transferability coefficient is a measure of how guarantees on p_L using a language model can be transferred to another distribution of contexts and it only depends on the distribution of contexts and not the labels. Unlike $\gamma(p_{\mathcal{T}})$, the coefficients in Definition D.1 depend on the learned models, either $\{p_{\cdot|s}\}$ or $\{p_{f(s)}\}$, and can be potentially much smaller due to the inductive bias of the learned models. For instance, if errors made by the model are random-like, i.e. $\Delta_{\{p_{f(s)}\}}(s) \sim \rho$, *independently of* s, then $\Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}}) \approx \Sigma_p(\Delta_{\{p_{\cdot|s}\}}) \approx \mathbb{E}_{\eta\sim\rho}[\eta\eta^\top]$, making $\gamma(p; \{p_{\cdot|s}\}) \approx 1$. The independence between $\Delta_{\{p_{f(s)}\}}(s)$ and s prevents language modeling error from accumulating on contexts from $p_{\mathcal{T}}$, bypassing the worst case transfer of $\gamma(p_{\mathcal{T}})$.

E Quad: A new objective function

In Definition 3.2 we discuss how low dimensional softmax language models learn a linear projection of $p^*_{\cdot|s}$, only solving tasks that lie in the row span of word embeddings Φ. Although Φ defines tasks that language model features can solve, the standard cross-entropy objective does not lend a simple closed form expression for optimal Φ. This motivates the construction of our Quad objective, that has two nice properties: (1) the optimal feature map f^* is a linear function of $p^*_{\cdot|s}$ and thus can solve some natural tasks, and (2) the optimal Φ^* has an intuitively meaningful closed-form solution.

$$\ell_{quad,s}(\theta, \Phi) = \mathop{\mathbb{E}}_{w\sim p^*_{\cdot|s}}[-\theta^\top\phi_w] + \frac{1}{2}\|\Phi^\top\theta\|^2 = -\theta^\top\Phi p^*_{\cdot|s} + \frac{1}{2}\|\Phi^\top\theta\|^2 \tag{10}$$

$$\ell_{quad}(f, \Phi) = \mathop{\mathbb{E}}_{s\sim p_L}[\ell_{quad,s}(f(s), \Phi)] \tag{11}$$

The Quad objective is very similar to the cross-entropy objective from Equation (2), with the log partition function replaced by a quadratic function, inspired in part by Assumption 4.1. We can derive the optimal solution Φ^* that depends on the eigen-decomposition of a *substitutability matrix*.

Definition 4.1. *The substitutability matrix is defined to be $\Omega^* \coloneqq \mathop{\mathbb{E}}_{s\sim p^*}\left[p^*_{\cdot|s}\, {p^*_{\cdot|s}}^\top\right] \in \mathbb{R}^{V\times V}$. If $\Omega^* = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^\top$ is the eigendecomposition, then $\boldsymbol{U}_d \in \mathbb{R}^{V\times d}$ is matrix of top d eigenvectors of Ω^*.*

The matrix Ω^* captures substitutability between pairs of words. Words w and w' are substitutable if they have identical conditional probabilities for every context $s \in \mathcal{S}$ and thus can replace occurrences of each other while still providing meaningful completions. By definition, these words satisfy $\Omega^*[w] = \Omega^*[w']$. Such pairs of words were called "free variants" in the work on distributional semantics [Harris, 1954], and capture the notion of synonyms in the distributional hypothesis. We now derive expressions for the optimal solution of the Quad objective described in Equation (11).

Theorem E.1. *The optimal solution $f^*, \Phi^* = \arg\min_{f,\Phi} \ell_{quad}(f, \Phi)$ satisfies*

$$\Phi^* = \boldsymbol{B}\boldsymbol{U}_d^\top, \textit{for full rank } \boldsymbol{B} \in \mathbb{R}^{d\times d}$$
$$f^*(s) = (\Phi^*{\Phi^*}^\top)^{-1/2}\Phi^* p^*_{\cdot|s} = \boldsymbol{C}\boldsymbol{U}_d^\top p^*_{\cdot|s}, \textit{for full rank } \boldsymbol{C} \in \mathbb{R}^{d\times d}$$

If Φ is fixed, then the optimal solution is $f^*(s) = (\Phi\Phi^\top)^{-1/2}\Phi p^*_{\cdot|s}$.

Theorem 4.4. *Let $f^*, \Phi^* = \arg\min_{f,\Phi} \ell_{quad}(f, \Phi)$. Then $\Phi^* = \boldsymbol{B}\boldsymbol{U}_d^\top$, for full rank $\boldsymbol{B} \in \mathbb{R}^{d\times d}$. Also, for a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ^*, we have $\ell_{\mathcal{T}}(f^*) \le \tau$.*

Thus f^* excels on natural tasks w.r.t. Φ^*, which in turn, is the best d-dimensional projection of Ω^*. Thus words $w, w' \in \mathcal{W}$ that are synonyms (hence substitutable) will satisfy $\phi^*_w = \phi^*_{w'}$, fulfilling the desired property for word embeddings discussed in Definition 3.2. We train using the Quad objective and compare its performance to a similarly trained language model in Section G.2, finding Quad to be reasonably effective. The goal of testing Quad is not to obtain state-of-the-art results, but to demonstrate that theoretical insights can aid the design of provably effective algorithms.

F Proofs

F.1 Proofs for unconstrained language models

Theorem D.1 (Strengthened Theorem 4.1)**.** *Let $\{p_{\cdot|s}\}$ be a language model that is ϵ-optimal, i.e. $\ell_{xent}(\{p_{\cdot|s}\}) - \ell^*_{xent} \le \epsilon$ for some $\epsilon > 0$. For a classification task $\mathcal{T}$ that is (τ, B)-natural, we have*

$$\ell_{\mathcal{T}}\left(\{p_{\cdot|s}\}\right) \le \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}$$

For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have

$$\ell_{\mathcal{T}}\left(\{p_{\cdot|s}\}\right) \le \ell_{\mathcal{T}}\left(\{\Phi p_{\cdot|s}\}\right) \le \tau + \sqrt{\frac{2B^2\epsilon}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}$$

Proof. The proof has two main steps that we summarize by the following two lemmas. The first one upper bounds the downstream performance on natural tasks with the covariance of errors.

Lemma F.1. *For a language model $\{p_{\cdot|s}\}$, if $\mathcal{T}$ is (τ, B)-natural,*

$$\ell_{\mathcal{T}}(\{p_{\cdot|s}\}) \le \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \le B} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}{\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}$$

If $\mathcal{T}$ is (τ, B)-natural w.r.t. $\Phi \in \mathbb{R}^{d\times V}$,

$$\ell_{\mathcal{T}}(\{\Phi p_{\cdot|s}\}) \le \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty \le B}} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}$$

where $\gamma(\cdot)$ and $\gamma_\Phi(\cdot)$ are from Definition D.1.

The second lemma upper bounds the covariance of error with the suboptimality of the language model.

Lemma F.2. *For a language model $\{p_{\cdot|s}\}$ and classifier $\boldsymbol{v} \in \mathbb{R}^V$,*

$$\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v} \le 2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{\textit{xent}}(\{p_{\cdot|s}\}) - \ell^*_{\textit{xent}}\right)$$

where $\Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}}) = \mathop{\mathbb{E}}_{s\sim p_L}\left[(p_{\cdot|s} - p^*_{\cdot|s})(p_{\cdot|s} - p^*_{\cdot|s})^\top\right]$ as defined in Section D.

Combining the two lemmas, we get the following inequality

$$\begin{aligned}
\ell_{\mathcal{T}}(\{p_{\cdot|s}\}) &\le^{(a)} \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \le B} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}{\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})}} \\
&\le^{(b)} \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \le B} \sqrt{\frac{2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{\text{xent}}(\{p_{\cdot|s}\}) - \ell^*_{\text{xent}}\right)}{\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})}} \\
&\le^{(c)} \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}
\end{aligned}$$

where (a) uses first part of Lemma F.1, (b) uses Lemma F.2 and (c) uses the ϵ-optimality of $\{p_{\cdot|s}\}$. This proves the first part of the result. The second part can also be proved similarly.

$$\begin{aligned}
\ell_{\mathcal{T}}(\{\Phi p_{\cdot|s}\}) &\le^{(a)} \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, \\ \|\boldsymbol{v}\|_\infty \le B}} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}} \\
&\le^{(b)} \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, \\ \|\boldsymbol{v}\|_\infty \le B}} \sqrt{\frac{2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{\text{xent}}(\{p_{\cdot|s}\}) - \ell^*_{\text{xent}}\right)}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}} \\
&\le \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \le B} \sqrt{\frac{2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{\text{xent}}(\{p_{\cdot|s}\}) - \ell^*_{\text{xent}}\right)}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}} \le^{(c)} \tau + \sqrt{\frac{2B^2\epsilon}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}
\end{aligned}$$

where (a) uses second part of Lemma F.1, (b) uses Lemma F.2 and (c) uses the ϵ-optimality of $\{p_{\cdot|s}\}$. The proof of the lemmas can be found in Section F.5. □

Theorem 4.1. *Let $\{p_{\cdot|s}\}$ be a language model that is ϵ-optimal, i.e. $\ell_{\textit{xent}}(\{p_{\cdot|s}\}) - \ell^*_{\textit{xent}} \le \epsilon$, for some $\epsilon > 0$. For a classification task $\mathcal{T}$ that is (τ, B)-natural, we have*

$$\ell_{\mathcal{T}}\left(\{p_{\cdot|s}\}\right) \le \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}})}}$$

Proof. This follows from the first part of Theorem D.1 if we can also show that $\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})^{-1} \le \gamma(p_{\mathcal{T}})^{-1}$. For that we use the following lemma that we prove in Section F.5.

Lemma F.3. *For any $g : \mathcal{S} \to \mathbb{R}^D$ and $p_{\mathcal{T}} \in \Delta_{\mathcal{S}}$, we have $\|\Sigma_{p_L}(g)^{-\frac{1}{2}}\Sigma_{p_{\mathcal{T}}}(g)\Sigma_{p_L}(g)^{-\frac{1}{2}}\|_2 \le \gamma(p_{\mathcal{T}})^{-1}$*

Instantiating this for $g = \Delta_{\{p_{\cdot|s}\}}$ and using Equation (7), we get $\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})^{-1} \le \gamma(p_{\mathcal{T}})^{-1}$, which completes the proof. □

F.2 Proofs for softmax language models

Theorem D.2 (Strengthened Theorem 4.2)**.** *For a fixed Φ, let f be features from an ϵ-optimal d-dimensional softmax language model, i.e. $\ell_{\textit{xent}}(f, \Phi) - \ell^*_{\textit{xent}}(\Phi) \le \epsilon$, where $\ell^*_{\textit{xent}}(\Phi)$ is defined in Equation (5). For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have*

$$\ell_{\mathcal{T}}\left(\{p_{f(s)}\}\right) \le \ell_{\mathcal{T}}(g_{f,\Phi}) \le \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}}; g_{f,\Phi})}}$$

Proof. Instantiating Lemma F.1 for $p_{\cdot|s} = p_{f(s)}$, we get

$$\begin{aligned}\ell_{\mathcal{T}}(\{\Phi p_{f(s)}\}) &\leq \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty\leq B}} \sqrt{\frac{\boldsymbol{v}^\top\Sigma_{p_L}(\Delta_{\{p_{f(s)}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}};\{p_{f(s)}\})}}\\ &=^{(a)} \tau + \sqrt{\frac{\sup\limits_{\|\Phi^\top\lambda\|_\infty\leq B} \lambda^\top\Phi\Sigma_{p_L}(\Delta_{\{p_{f(s)}\}})\Phi^\top\lambda}{\gamma_\Phi(p_{\mathcal{T}};g_{f,\Phi})}}\\ &= \tau + \sqrt{\frac{\sup\limits_{\|\Phi^\top\lambda\|_\infty\leq B} \lambda^\top\Sigma_{p_L}(\Phi\Delta_{\{p_{f(s)}\}})\lambda}{\gamma_\Phi(p_{\mathcal{T}};g_{f,\Phi})}}\end{aligned}$$

where (a) follows from Equation (9) that says $\gamma_\Phi(p_{\mathcal{T}};g_{f,\Phi}) = \gamma_\Phi(p_{\mathcal{T}};\{p_{f(s)}\})$. We now prove a similar result for the second term in the following lemma

Lemma F.4. *For a fixed Φ and a softmax language model with features f and $\lambda \in \mathbb{R}^d$,*

$$\lambda^\top\Sigma_{p_L}(\Phi\Delta_{\{p_{f(s)}\}})\lambda \leq 2\|\Phi^\top\lambda\|_\infty^2\left(\ell_{\textit{xent}}(f,\Phi) - \ell^*_{\textit{xent}}(\Phi)\right)$$

where $\Sigma_{p_L}(\Phi\Delta_{\{p_{f(s)}\}}) = \mathop{\mathbb{E}}\limits_{s\sim p_L}\left[(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})^\top\right]$ as defined in Section D.

Using Lemma F.4 directly gives us $\boldsymbol{W}(g_{f,\Phi}) = \ell_{\mathcal{T}}(\{\Phi p_{f(s)}\}) \leq \tau + \sqrt{\frac{B^2(\ell_{\text{xent}}(f,\Phi)-\ell^*_{\text{xent}}(\Phi))}{\gamma_\Phi(p_{\mathcal{T}};g_{f,\Phi})}}$, and the ϵ-optimality almost completes the proof. The only thing remaining to show is that $\ell_{\mathcal{T}}(\{p_{f(s)}\}) \leq \ell_{\mathcal{T}}(g_{f,\Phi})$ which follows from the following sequence.

$$\begin{aligned}\ell_{\mathcal{T}}(\{p_{f(s)}\}) &= \inf_{\boldsymbol{v}\in\mathbb{R}^V,b\in\mathbb{R}} \ell_{\mathcal{T}}(\{p_{f(s)}\},\boldsymbol{v}) \leq \inf_{\Phi^\top\lambda\in\mathbb{R}^V,b\in\mathbb{R}} \ell_{\mathcal{T}}(\{p_{f(s)}\},(\Phi^\top\lambda,b))\\ &= \inf_{\lambda\in\mathbb{R}^d,b\in\mathbb{R}} \ell_{\mathcal{T}}(\{\Phi p_{f(s)}\},(\lambda,b)) = \ell_{\mathcal{T}}(g_{f,\Phi})\end{aligned}$$

□

Theorem 4.2. *For a fixed Φ, let f be features from an ϵ-optimal d-dimensional softmax language model, i.e. $\ell_{\textit{xent}}(f,\Phi) - \ell^*_{\textit{xent}}(\Phi) \leq \epsilon$, where $\ell^*_{\textit{xent}}(\Phi)$ is defined in Equation (5). For a classification task $\mathcal{T}$ that is (τ, B)-natural w.r.t. Φ, we have*

$$\ell_{\mathcal{T}}\left(\{p_{f(s)}\}\right) \leq \ell_{\mathcal{T}}(g_{f,\Phi}) \leq \tau + \sqrt{\frac{2B^2\epsilon}{\gamma(p_{\mathcal{T}})}}$$

Proof. This result follows directly from Theorem D.2, if we can also show that $\gamma(p_{\mathcal{T}};g_{f,\Phi})^{-1} \leq \gamma(p_{\mathcal{T}})^{-1}$ just like in the proof of Theorem 4.1. For that we again use Lemma F.3 with $g = \Phi\Delta_{\{p_{f(s)}\}}$ and Equation (9) and this completes the proof. □

F.3 Proofs for Section C.3

We first show why Assumption 4.1 is approximately true when word embeddings are gaussian like.

Lemma F.5. *Suppose word embeddings ϕ_w are independent samples from the distribution $\mathcal{N}(\mu,\Sigma)$. Then for any $\theta \in \mathbb{R}^d$ such that $\lambda^2 = \theta^\top\Sigma\theta = O(1)$ we have that $|\log(Z_\theta) - \frac{1}{2}\theta^\top\Sigma\theta - \theta^\top\mu - \log(V)| \leq \epsilon$ with probability $1-\delta$ for $\epsilon = \tilde{O}\left(\frac{e^{\lambda^2}}{\sqrt{V}}\right)$ and $\delta = 1 - \exp(-\Omega(\log^2(V)))$.*

Proof. We first note that $\log(Z_\theta) = \log\left(\sum_w e^{\theta^\top\phi_w}\right) = \theta^\top\mu + \log\left(\sum_w e^{\theta^\top(\phi_w-\mu)}\right)$, thus we can simply deal with the case where ϕ_w are sampled from $\mathcal{N}(0,\Sigma)$. Furthermore the only random variable of interest is $X_w = \theta^\top\phi_w$ which is a gaussian variable $\mathcal{N}(0,\theta^\top\Sigma\theta) = \mathcal{N}(0,\lambda^2)$. Thus the problem reduces to showing that for V samples of $X_w \sim \mathcal{N}(0,\lambda^2)$, $\log(Z)$ is concentrated around

$\lambda^2 + \log(V)$ where $Z = \sum_w \exp(X_w)$. This can be proved similarly to the proof of Lemma 2.1 in Arora et al. [2016]. It is easy to see that $\underset{X_w \sim \mathcal{N}(0,\lambda^2)}{\mathbb{E}}[\exp(X_w)] = e^{\lambda^2}$. However the variable $\exp(X_w)$ is neither sub-gaussian nor sub-exponential and thus standard inequalities cannot be used directly. We use the same technique as Arora et al. [2016] to first observe that $\mathbb{E}[Z] = Ve^{\frac{1}{2}\lambda^2}$ and $\mathrm{Var}[Z] \leq \mathbb{E}[\exp(2X_w)] = Ve^{2\lambda^2}$. After conditioning on the event that $X_w \leq \frac{1}{2}\lambda\log(V)$ and applying Berstein's inequality just like in Arora et al. [2016] completes the proof. □

Lemma 4.3. *Under Assumption 4.1, any feature map $f : \mathcal{S} \to \mathbb{R}^d$ satisfies $g_{f,\Phi}(s) = \boldsymbol{A}f(s) + \boldsymbol{b}$, for all $s \in \mathcal{S}$.*

Proof. Assumption 4.1 gives us that $\log(Z_\theta) = \frac{1}{2}\theta^\top \boldsymbol{A}\theta + \theta^\top \boldsymbol{b} + c$. We prove this lemma by matching the gradients of $\log(Z_\theta)$ and the quadratic function on the R.H.S.

$$\nabla_\theta \log(Z_\theta) = \frac{\nabla_\theta Z_\theta}{Z_\theta} = \frac{\sum_{w \in \mathcal{W}} e^{\phi_w^\top \theta} \phi_w}{Z_\theta} = \sum_{w \in \mathcal{W}} p_\theta(w)\phi_w = \Phi p_\theta$$

Whereas the gradient of the quadratic part is $\nabla_\theta[\frac{1}{2}\theta^\top \boldsymbol{A}\theta + \theta^\top \boldsymbol{b} + c] = \boldsymbol{A}\theta + \boldsymbol{b}$. Matching the two gives us $g_{f,\Phi}(s) = \Phi p_{f(s)} = \boldsymbol{A}f(s) + \boldsymbol{b}$. □

Corollary 4.1. *Using Lemma 4.3, for any ϵ-optimal f, as defined in Theorem 4.2, for classification tasks that are (τ, B)-natural w.r.t. Φ we have $\ell_{\mathcal{T}}(f) \leq \tau + \mathcal{O}(\sqrt{\epsilon})$.*

Proof. The main idea is that Lemma 4.3 gives us that $g_{f,\Phi}(s) = \boldsymbol{A}f(s) + \boldsymbol{b}$ and thus any linear function of $g_{f,\Phi}$ will also be a linear function of $f(s)$. And from Theorem D.2 (or Theorem 4.2), we know that $g_{f,\Phi}$ will do well on $\mathcal{T}$, i.e. $\ell_{\mathcal{T}}(g_{f,\Phi}) \leq \tau + \mathcal{O}(B\sqrt{\epsilon})$. We formalize the intuition as

$$\begin{aligned}
\ell_{\mathcal{T}}(g_{f,\Phi}) &= \inf_{\lambda \in \mathbb{R}^d} \ell_{\mathcal{T}}(g_{f,\Phi}, (\lambda, b)) = \inf_{\lambda \in \mathbb{R}^d} \ell_{\mathcal{T}}(\boldsymbol{A}f + \boldsymbol{b}, (\lambda, b)) = \inf_{\lambda \in \mathbb{R}^d} \ell_{\mathcal{T}}(f, (\boldsymbol{A}^\top \lambda, b + \lambda^\top \boldsymbol{b})) \\
&\geq \inf_{\boldsymbol{v} \in \mathbb{R}^d} \ell_{\mathcal{T}}(f, \boldsymbol{v}) = \ell_{\mathcal{T}}(f)
\end{aligned}$$

This shows that $\ell_{\mathcal{T}}(f) \leq \ell_{\mathcal{T}}(g_{f,\Phi}) \leq \tau + \mathcal{O}(B\sqrt{\epsilon})$ and completes the proof. □

F.4 Proofs for Section E

Theorem E.1. *The optimal solution $f^*, \Phi^* = \arg\min_{f,\Phi} \ell_{quad}(f, \Phi)$ satisfies*

$$\begin{aligned}
\Phi^* &= \boldsymbol{B}\boldsymbol{U}_d^\top, \textit{for full rank } \boldsymbol{B} \in \mathbb{R}^{d \times d} \\
f^*(s) &= (\Phi^* {\Phi^*}^\top)^{-1/2} \Phi^* p^*_{\cdot|s} = \boldsymbol{C}\boldsymbol{U}_d^\top p^*_{\cdot|s}, \textit{for full rank } \boldsymbol{C} \in \mathbb{R}^{d \times d}
\end{aligned}$$

If Φ is fixed, then the optimal solution is $f^*(s) = (\Phi\Phi^\top)^{-1/2}\Phi p^*_{\cdot|s}$.

Proof. Given that, $\ell_{quad,s}(\theta, \Phi) = -\theta^\top \Phi p^*_{\cdot|s} + \frac{1}{2}\|\Phi^\top \theta\|^2$, for a fixed Φ, we define $f^*_\Phi(s) = \arg\min_{\theta \in \mathbb{R}^d} \ell_{quad,s}(\theta, \Phi)$. We use the first order optimality condition to get $f^*_\Phi(s)$, by using the fact that $\nabla_\theta \ell_{quad,s}(\theta, \Phi) = -\Phi p^*_{\cdot|s} + \Phi\Phi^\top \theta$. Setting the gradient to zero, we $f^*_\Phi(s) = (\Phi\Phi^\top)^{-1}\Phi p^*_{\cdot|s}$. To get the optimal Φ^* for this objective, we plug in this expression for $f^*_\Phi(s)$ and observing that the optimal solution also satisfies $\Phi^* = \arg\min_\Phi \ell_{quad}(f^*_\Phi, \Phi)$.

$$\begin{aligned}
\ell_{quad}(f^*_\Phi, \Phi) &= \underset{s \sim p^*}{\mathbb{E}}[\ell_{quad,s}(f^*_\Phi(s), \Phi)] = \underset{s \sim p^*}{\mathbb{E}}\left[-f^*_\Phi(s)^\top \Phi p^*_{\cdot|s} + \frac{1}{2}\|\Phi^\top f^*_\Phi(s)\|^2\right] \\
&= \underset{s \sim p^*}{\mathbb{E}}\left[-((\Phi\Phi^\top)^{-1}\Phi p^*_{\cdot|s})^\top \Phi p^*_{\cdot|s} + \frac{1}{2}\|\Phi^\top (\Phi\Phi^\top)^{-1}\Phi p^*_{\cdot|s}\|^2\right] \\
&= \underset{s \sim p^*}{\mathbb{E}}\left[-{p^*_{\cdot|s}}^\top \Phi^\top (\Phi\Phi^\top)^{-1}\Phi p^*_{\cdot|s} + \frac{1}{2}{p^*_{\cdot|s}}^\top \Phi^\top (\Phi\Phi^\top)^{-1}\Phi\Phi^\top (\Phi\Phi^\top)^{-1}\Phi p^*_{\cdot|s}\right] \\
&= \underset{s \sim p^*}{\mathbb{E}}\left[-\frac{1}{2}{p^*_{\cdot|s}}^\top \Phi^\top (\Phi\Phi^\top)^{-1}\Phi p^*_{\cdot|s}\right] = -\frac{1}{2}\underset{s \sim p^*}{\mathbb{E}}\left[\mathrm{tr}\left({p^*_{\cdot|s}}^\top \Phi^\top (\Phi\Phi^\top)^{-1}\Phi p^*_{\cdot|s}\right)\right]
\end{aligned}$$

$$
\begin{aligned}
&= -\frac{1}{2}\text{tr}\left(\Phi^\top(\Phi\Phi^\top)^{-1}\Phi \mathop{\mathbb{E}}_{s\sim p^*}\left[p^*_{\cdot|s}{p^*_{\cdot|s}}^\top\right]\right)\\
&= -\frac{1}{2}\left\langle \Phi^\top(\Phi\Phi^\top)^{-1}\Phi, \mathop{\mathbb{E}}_{s\sim p^*}\left[p^*_{\cdot|s}{p^*_{\cdot|s}}^\top\right]\right\rangle = -\frac{1}{2}\left\langle \Phi^\top(\Phi\Phi^\top)^{-1}\Phi, \Omega^*\right\rangle
\end{aligned}
$$

Let $\Phi = \boldsymbol{NTV}^\top$ be the SVD. Then the above objective reduces to $\ell_{quad}(f^*_\Phi, \Phi) = -\frac{1}{2}\left\langle \boldsymbol{VV}^\top, \Omega^*\right\rangle$ And hence learning the optimal Φ^* reduces to learning an optimal $\boldsymbol{V}^*$ such that

$$\boldsymbol{V}^* = \mathop{\arg\min}_{\boldsymbol{V}\in\mathbb{R}^{V\times d}, \boldsymbol{V}^\top\boldsymbol{V}=I_d} -\langle \boldsymbol{VV}^\top, \Omega^*\rangle$$

We will now show that the best such matrix is the matrix of top d eigenvectors of Ω^*, i.e. $\boldsymbol{V}^* = \boldsymbol{U}_d$. Here we will assume that the eigenvalues of Ω^* are all distinct for simplicity of presentation.

First we note that $\langle \boldsymbol{VV}^\top, \Omega^*\rangle = \|\boldsymbol{VV}^\top {\Omega^*}^{\frac{1}{2}}\|_F^2$, where ${\Omega^*}^{\frac{1}{2}} = \boldsymbol{US}^{\frac{1}{2}}\boldsymbol{U}^\top$, with $\boldsymbol{U}, \boldsymbol{U}_d$ and $\boldsymbol{S}$ define in Definition 4.1. This can be shown by the following sequence of steps

$$
\begin{aligned}
\langle \boldsymbol{VV}^\top, \Omega^*\rangle &= \text{tr}(\boldsymbol{VV}^\top\Omega^*) = \text{tr}(\boldsymbol{VV}^\top\boldsymbol{VV}^\top\Omega^*) = \text{tr}(\boldsymbol{VV}^\top\Omega^*\boldsymbol{VV}^\top)\\
&= \text{tr}(\boldsymbol{VV}^\top\boldsymbol{USU}^\top\boldsymbol{VV}^\top) = \text{tr}(\boldsymbol{VV}^\top\boldsymbol{US}^{\frac{1}{2}}\boldsymbol{U}^\top\boldsymbol{US}^{\frac{1}{2}}\boldsymbol{U}^\top\boldsymbol{VV}^\top)\\
&= \text{tr}(\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}{\Omega^*}^{\frac{1}{2}}\boldsymbol{VV}^\top) = \langle \boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}, \boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\rangle\\
&= \|\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2
\end{aligned}
$$

Furthermore, we notice that $\|\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2 = \|{\Omega^*}^{\frac{1}{2}}\|_F^2 - \|{\Omega^*}^{\frac{1}{2}} - \boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2$ as shown below

$$
\begin{aligned}
\|{\Omega^*}^{\frac{1}{2}} - \boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2 &= \|{\Omega^*}^{\frac{1}{2}}\|_F^2 + \|\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2 - 2\text{tr}({\Omega^*}^{\frac{1}{2}}\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}})\\
&= \|{\Omega^*}^{\frac{1}{2}}\|_F^2 + \|\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2 - 2\text{tr}({\Omega^*}^{\frac{1}{2}}\boldsymbol{VV}^\top\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}})\\
&= \|{\Omega^*}^{\frac{1}{2}}\|_F^2 + \|\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2 - 2\|\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2\\
&= \|{\Omega^*}^{\frac{1}{2}}\|_F^2 - \|\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2
\end{aligned}
$$

Thus we get $\mathop{\arg\min}_{\boldsymbol{V}\in\mathbb{R}^{V\times d}, \boldsymbol{V}^\top\boldsymbol{V}=I_d} -\langle \boldsymbol{VV}^\top, \Omega^*\rangle = \mathop{\arg\min}_{\boldsymbol{V}\in\mathbb{R}^{V\times d}, \boldsymbol{V}^\top\boldsymbol{V}=I_d} \|{\Omega^*}^{\frac{1}{2}} - \boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}\|_F^2$.

Note that $\boldsymbol{VV}^\top{\Omega^*}^{\frac{1}{2}}$ has columns that are columns of ${\Omega^*}^{\frac{1}{2}}$ projected on the space spanned by columns $\boldsymbol{V}$. It is folklore that the best such subspace $\boldsymbol{V}^*$ is the subspace spanned by the top d eigenvectors of ${\Omega^*}^{\frac{1}{2}}$, which is the same as top d eigenvectors of Ω^*, thus giving us $\boldsymbol{V}^*{\boldsymbol{V}^*}^\top = \boldsymbol{U}_d\boldsymbol{U}_d^\top$. Thus we get $\boldsymbol{V}^* = \boldsymbol{U}_d\boldsymbol{M}$ for $\boldsymbol{M} = \boldsymbol{U}_d^\top\boldsymbol{V}^*$.

This tells us that the optimal solution Φ^* will have SVD of the form $\Phi^* = \boldsymbol{N}^*\boldsymbol{T}^*{\boldsymbol{V}^*}^\top$, thus giving us $\Phi^* = \boldsymbol{BU}_d^\top$ for matrix $\boldsymbol{B} = \boldsymbol{N}^*\boldsymbol{T}^*\boldsymbol{M}^\top \in \mathbb{R}^{d\times d}$. This directly gives $f^* = f^*_{\Phi^*} = (\Phi^*{\Phi^*}^\top)^{-1}\Phi^* p^*_{\cdot|s} = \boldsymbol{N}^*\boldsymbol{T}^{-1}{\boldsymbol{V}^*}^\top p^*_{\cdot|s} = \boldsymbol{CU}_d^\top p^*_{\cdot|s}$ for $\boldsymbol{C} = \boldsymbol{N}^*{\boldsymbol{T}^*}^{-1}\boldsymbol{M}^\top$.

□

F.5 Proof for supporting lemmas

Lemma F.1. *For a language model $\{p_{\cdot|s}\}$, if $\mathcal{T}$ is (τ, B)-natural,*

$$\ell_{\mathcal{T}}(\{p_{\cdot|s}\}) \leq \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty\leq B} \sqrt{\frac{\boldsymbol{v}^\top\Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}{\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}$$

If $\mathcal{T}$ is (τ, B)-natural w.r.t. $\Phi \in \mathbb{R}^{d\times V}$,

$$\ell_{\mathcal{T}}(\{\Phi p_{\cdot|s}\}) \leq \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty\leq B}} \sqrt{\frac{\boldsymbol{v}^\top\Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}$$

where $\gamma(\cdot)$ and $\gamma_\Phi(\cdot)$ are from Definition D.1.

Proof. We note the following upper bounds on $\ell_{\mathcal{T}}(\{p_{\cdot|s}\})$ and $\ell_{\mathcal{T}}(\{\Phi p_{\cdot|s}\})$.

$$\ell_{\mathcal{T}}(\{p_{\cdot|s}\}) = \inf_{\boldsymbol{v}\in\mathbb{R}^V} \left\{\ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v})\right\} \le \inf_{\substack{\boldsymbol{v}\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty \le B}} \left\{\ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v})\right\} \tag{12}$$

$$\ell_{\mathcal{T}}(\{\Phi p_{\cdot|s}\}) = \inf_{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V} \left\{\ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v})\right\} \le \inf_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, b\in\mathbb{R},\\ \|\boldsymbol{v}\|_\infty \le B}} \left\{\ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v})\right\} \tag{13}$$

When $\mathcal{T}$ is (τ, B)-natural, by Definition 3.1 we know that $\inf_{\substack{\boldsymbol{v}\in\mathbb{R}^V\\ \|\boldsymbol{v}\|_\infty \le B}} \left[\ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v})\right] \le \tau$. We now upper bound $\ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v})$ using Lemma F.8. Taking infimum w.r.t. $\boldsymbol{v} \in \mathbb{R}^V$, $\|\boldsymbol{v}\|_\infty \le B$ from the inequality in Lemma F.8.

$$\ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v}) \le \ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v}) + \sqrt{\boldsymbol{v}^\top \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}$$

$$\inf_{\substack{\boldsymbol{v}\in\mathbb{R}^V\\ \|\boldsymbol{v}\|_\infty \le B}} \ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v}) \le \inf_{\substack{\boldsymbol{v}\in\mathbb{R}^V\\ \|\boldsymbol{v}\|_\infty \le B}} \ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v}) + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \le B} \sqrt{\boldsymbol{v}^\top \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}$$

This, combined with Equation (12), gives us

$$\ell_{\mathcal{T}}(\{p_{\cdot|s}\}) \le \tau + \sup_{\boldsymbol{v}\in\mathbb{R}^V, \|\boldsymbol{v}\|_\infty \le B} \sqrt{\boldsymbol{v}^\top \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}} \tag{14}$$

Using Lemma F.9 and the definition of $\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})$ in Equation (7), we get that

$$\begin{aligned}\boldsymbol{v}^\top \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v} &\le \left\|\Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})^{-\frac{1}{2}} \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}}) \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})^{-\frac{1}{2}}\right\|_2 \left(\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}\right)\\ &= \frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}{\gamma(p_{\mathcal{T}}; \{p_{\cdot|s}\})}\end{aligned}$$

We have thus successfully transferred the bound from the distribution $p_{\mathcal{T}}$ to p_L. Combining this with Equation (14) completes the proof of the first part of the lemma.

We now prove the second part of the lemma where we only assume that $\mathcal{T}$ is (τ, B)-natural w.r.t. Φ. Here we instead take the infimum over classifiers in the span of Φ in Lemma F.8 to get

$$\begin{aligned}\inf_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, b\in\mathbb{R},\\ \|\boldsymbol{v}\|_\infty \le B}} \left\{\ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v})\right\} \le& \inf_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V, b\in\mathbb{R},\\ \|\boldsymbol{v}\|_\infty \le B}} \left\{\ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v})\right\} +\\ &\sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty \le B}} \sqrt{\boldsymbol{v}^\top \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}\end{aligned} \tag{15}$$

This, combined with definition of (τ, B)-natural task w.r.t. Φ and Equation (13) gives us

$$\ell_{\mathcal{T}}(\{\Phi p_{\cdot|s}\}) \le \tau + \sup_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty \le B}} \sqrt{\boldsymbol{v}^\top \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}} \tag{16}$$

For the last term, for any $\boldsymbol{v} = \Phi^\top\lambda$, $\lambda \in \mathbb{R}^d$ we notice that

$$\begin{aligned}\boldsymbol{v}^\top \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v} &= \lambda^\top \Phi \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\Phi^\top\lambda = \lambda^\top \Sigma_{p_{\mathcal{T}}}(\Phi\Delta_{\{p_{\cdot|s}\}})\lambda\\ &\le^{(a)} \left\|\Sigma_{p_L}(\Phi\Delta_{\{p_{\cdot|s}\}})^{-\frac{1}{2}} \Sigma_{p_{\mathcal{T}}}(\Phi\Delta_{\{p_{\cdot|s}\}}) \Sigma_{p_L}(\Phi\Delta_{\{p_{\cdot|s}\}})^{-\frac{1}{2}}\right\|_2 \left(\lambda^\top \Sigma_{p_L}(\Phi\Delta_{\{p_{\cdot|s}\}})\lambda\right)\\ &= \frac{\lambda^\top \Sigma_{p_L}(\Phi\Delta_{\{p_{\cdot|s}\}})\lambda}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})} = \frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}\end{aligned}$$

This combined with Equation (16), we get

$$\ell_{\mathcal{T}}(\{\Phi p_{\cdot|s}\}) \le \tau + \inf_{\substack{\boldsymbol{v}=\Phi^\top\lambda\in\mathbb{R}^V,\\ \|\boldsymbol{v}\|_\infty \le B}} \sqrt{\frac{\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}{\gamma_\Phi(p_{\mathcal{T}}; \{p_{\cdot|s}\})}}$$

□

Lemma F.2. *For a language model $\{p_{\cdot|s}\}$ and classifier $\boldsymbol{v} \in \mathbb{R}^V$,*

$$\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v} \leq 2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{xent}(\{p_{\cdot|s}\}) - \ell_{xent}^*\right)$$

where $\Sigma_{p_L}(g) = \underset{s\sim p_L}{\mathbb{E}}[g(s)g(s)^\top]$ and $\Delta_{\{p_{\cdot|s}\}}(s) = p_{\cdot|s} - p^*_{\cdot|s}$ are defined in Section D

Proof. We first note that

$$\ell_{\text{xent}}(\{p_{\cdot|s}\}) - \ell_{\text{xent}}(\{p^*_{\cdot|s}\}) = \underset{s\sim p_L}{\mathbb{E}} \underset{w\sim p^*_{\cdot|s}}{\mathbb{E}} \left[\log\left(\frac{p^*_{\cdot|s}(w)}{p_{\cdot|s}(w)}\right)\right] = \underset{s\sim p_L}{\mathbb{E}}\left[D_{\mathrm{KL}}(p^*_{\cdot|s}, p_{\cdot|s})\right] \tag{17}$$

We bound $\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}$ below

$$\begin{aligned}\boldsymbol{v}^\top \Sigma_{p_L}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v} &= \underset{s\sim p_L}{\mathbb{E}}\left[\left(\boldsymbol{v}^\top(p_{\cdot|s} - p^*_{\cdot|s})\right)^2\right] \leq^{(a)} \underset{s\sim p_L}{\mathbb{E}}\left[\|\boldsymbol{v}\|_\infty^2 \|p_{\cdot|s} - p^*_{\cdot|s}\|_1^2\right] \\ &\leq^{(b)} \|\boldsymbol{v}\|_\infty^2 \underset{s\sim p_L}{\mathbb{E}}\left[2D_{\mathrm{KL}}(p^*_{\cdot|s}, p_{\cdot|s})\right] \\ &=^{(c)} 2\|\boldsymbol{v}\|_\infty^2 \left(\ell_{\text{xent}}(\{p_{\cdot|s}\}) - \ell_{\text{xent}}(\{p^*_{\cdot|s}\})\right)\end{aligned}$$

where (a) uses Holder's inequality, (b) uses Pinsker's inequality, (c) uses Equation (17). □

Lemma F.3. *For any $g : \mathcal{S} \to \mathbb{R}^D$ and $p_\mathcal{T} \in \Delta_\mathcal{S}$, we have $\|\Sigma_{p_L}(g)^{-\frac{1}{2}}\Sigma_{p_\mathcal{T}}(g)\Sigma_{p_L}(g)^{-\frac{1}{2}}\|_2 \leq \gamma(p_\mathcal{T})^{-1}$*

Proof. By definition of $\gamma(p_\mathcal{T})$, we have that

$$\begin{aligned}\Sigma_{p_L}(g) &= \underset{s\sim p_L}{\mathbb{E}}[g(s)g(s)^\top] = \sum_{s\in\mathcal{S}} p_L(s)g(s)g(s)^\top \\ &\succcurlyeq \gamma(p_\mathcal{T})\sum_{s\in\mathcal{S}} p_\mathcal{T}(s)g(s)g(s)^\top = \gamma(p_\mathcal{T}) \underset{s\sim p_\mathcal{T}}{\mathbb{E}}[g(s)g(s)^\top] = \gamma(p_\mathcal{T})\Sigma_{p_\mathcal{T}}(g)\end{aligned}$$

Thus $\frac{1}{\gamma(p_\mathcal{T})}\Sigma_{p_L}(g) \succcurlyeq \Sigma_{p_\mathcal{T}}(g)$ and hence $\frac{1}{\gamma(p_\mathcal{T})}\Sigma_{p_L}(g)^{-\frac{1}{2}}\Sigma_{p_L}(g)\Sigma_{p_L}(g)^{-\frac{1}{2}} \succcurlyeq \Sigma_{p_L}(g)^{-\frac{1}{2}}\Sigma_{p_\mathcal{T}}(g)\Sigma_{p_L}(g)^{-\frac{1}{2}}$, which is equivalent to $\frac{1}{\gamma(p_\mathcal{T})}I_D \succcurlyeq \Sigma_{p_L}(g)^{-\frac{1}{2}}\Sigma_{p_\mathcal{T}}(g)\Sigma_{p_L}(g)^{-\frac{1}{2}}$. This finishes the proof. □

Lemma F.4. *For a fixed Φ, a softmax language model with features f and $\lambda \in \mathbb{R}^d$,*

$$\lambda^\top \Sigma_{p_L}(\Phi\Delta_{\{p_{f(s)}\}})\lambda \leq 2\|\Phi^\top\lambda\|_\infty^2 \left(\ell_{xent}(f, \Phi) - \ell_{xent}^*(\Phi)\right)$$

where $\Sigma_{p_L}(\Phi\Delta_{\{p_{f(s)}\}}) = \underset{s\sim p_L}{\mathbb{E}}\left[(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})^\top\right]$ as defined in Section D.

Proof. We start by nothing that $\lambda^\top \Sigma_{p_L}(\Phi\Delta_{\{p_{f(s)}\}})\lambda = \lambda^\top \underset{s\sim p_L}{\mathbb{E}}\left[(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})^\top\right]\lambda = \underset{s\sim p_L}{\mathbb{E}}[|\lambda^\top(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})|^2]$. We will use Lemma F.6 to bound each term on the right hand side, which essentially bounds the norm of the gradient of $\ell_{\text{xent},s}$ at $f(s)$ when $f(s)$ is an almost optimal for $s \in \mathcal{S}$. Notice that $\ell_{\text{xent}}(f, \Phi) - \ell^*_{\text{xent}}(\Phi) = \underset{s\sim p_L}{\mathbb{E}}[\ell_{\text{xent},s}(f(s), \Phi) - \inf_{\theta\in\mathbb{R}^d} \ell_{\text{xent},s}(\theta, \Phi)]$.

$$\begin{aligned}\lambda^\top \Sigma_{p_L}(\Phi\Delta_{\{p_{f(s)}\}})\lambda &= \underset{s\sim p_L}{\mathbb{E}}[|\lambda^\top(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})|^2] \\ &\leq^{(a)} 2\|\Phi^\top\lambda\|_\infty^2 \underset{s\sim p_L}{\mathbb{E}}\left[\ell_{\text{xent},s}(f(s), \Phi) - \inf_{\theta\in\mathbb{R}^d} \ell_{\text{xent},s}(\theta, \Phi)\right] \\ &\leq 2\|\Phi^\top\lambda\|_\infty^2 \left(\ell_{\text{xent}}(f, \Phi) - \ell^*_{\text{xent}}(\Phi)\right)\end{aligned}$$

where (a) follows from Lemma F.6. This completes the proof. □

Lemma F.6. *For $s \in \mathcal{S}$ and embedding $f(s) \in \mathbb{R}^d$, we have*

$$|\lambda^\top(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})|^2 \leq 2\|\Phi^\top \lambda\|_\infty^2 \left(\ell_{xent,s}(f(s), \Phi) - \inf_{\theta \in \mathbb{R}^d} \ell_{xent,s}(\theta, \Phi)\right)$$

Proof. Since we are assuming Φ to be fixed, we will abuse notation and say $\ell_{\text{xent},s}(\theta) \coloneqq \ell_{\text{xent},s}(\theta, \Phi)$. All gradients are w.r.t. θ. Before we get to the main proof, we compute the gradient and hessian of $\ell_{\text{xent},s}(\theta)$ w.r.t. θ. The gradient is

$$\begin{aligned}
\nabla \ell_{\text{xent},s}(\theta) &= \nabla\left[-\theta^\top \Phi p^*_{\cdot|s} + \log(Z_\theta)\right] = -\Phi p^*_{\cdot|s} + \frac{\nabla Z_\theta}{Z_\theta} \\
&= -\Phi p^*_{\cdot|s} + \frac{\nabla \sum_w e^{\theta^\top \phi_w}}{Z_\theta} = -\Phi p^*_{\cdot|s} + \frac{\sum_w e^{\theta^\top \phi_w} \phi_w}{Z_\theta} \\
&= -\Phi p^*_{\cdot|s} + \Phi p_\theta
\end{aligned}$$

Similarly the Hessian can be computed

$$\begin{aligned}
\nabla^2 \ell_{\text{xent},s}(\theta) &= \nabla(\nabla \ell_{\text{xent},s}(\theta)) = \nabla[-\Phi p^*_{\cdot|s} + \Phi p_\theta] = \nabla \sum_{w \in \mathcal{W}} p_\theta(w) \phi_w = \sum_{w \in \mathcal{W}} \nabla \frac{e^{\theta^\top \phi_w}}{Z_\theta} \phi_w \\
&= \sum_{w \in \mathcal{W}} \frac{e^{\theta^\top \phi_w}}{Z_\theta} \phi_w \phi_w^\top - \frac{e^{\theta^\top \phi_w}}{Z_\theta^2} \phi_w \left(\sum_{w'} e^{\theta^\top \phi_{w'}} \phi_{w'}\right)^\top \\
&= \mathop{\mathbb{E}}_{w \sim p_\theta}[\phi_w \phi_w^\top] - \left(\mathop{\mathbb{E}}_{w \sim p_\theta} \phi_w\right)\left(\mathop{\mathbb{E}}_{w \sim p_\theta} \phi_w\right)^\top = \text{Cov}_{w \sim p_\theta}[\phi_w]
\end{aligned}$$

Where $\text{Cov}_{w \sim p_\theta}[\phi_w]$ denotes the covariance of the word embeddings ϕ_w when measured w.r.t. the distribution p_θ. This directly gives us that $\nabla^2 \ell_{\text{xent},s}(\theta) \succcurlyeq 0$, since the covariance is always psd, and thus $\ell_{\text{xent},s}$ is convex in θ.

Using the closed form expression for $\nabla \ell_{\text{xent},s}$, we note that the quantity we wish to upper bound can be rewritten as $|\lambda^\top(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})|^2 = |\lambda^\top \nabla \ell_{\text{xent},s}(f(s))|^2$. Furthermore, using the definition of the Hessian, it is not hard to see for some $\lambda, \tilde{\theta} \in \mathbb{R}^d$ that $\lambda^\top \nabla^2 \ell_{\text{xent},s}(\tilde{\theta}) \lambda = \text{Cov}_{w \sim p_{\tilde{\theta}}}[\lambda^\top \phi_w] \leq \mathop{\mathbb{E}}_{w \sim p_{\tilde{\theta}}}[(\lambda^\top \phi_w)^2] \leq \|\Phi^\top \lambda\|_\infty^2$. We use the following lemma that can exploit the above observations.

Lemma F.7. *If a function $\ell : \mathbb{R}^d \to \mathbb{R}$ and $\lambda \in \mathbb{R}^d$ satisfy $\lambda^\top \nabla^2 \ell(\tilde{\theta}) \lambda \leq L, \forall \tilde{\theta} \in \mathbb{R}^d$ (L-smoothness in the direction of λ) and if $\ell^* = \inf_{\theta \in \mathbb{R}^d} \ell(\theta)$, then $|\lambda^\top \nabla \ell(\theta)|^2 \leq 2L(\ell(\theta) - \ell^*)$*

We first use this lemma for $\ell_{\text{xent},s}$ to complete the proof with $L = \|\Phi^\top \lambda\|_\infty^2$. The lemma gives us that $|\lambda^\top \nabla \ell_{\text{xent},s}(f(s))|^2 \leq 2\|\Phi^\top \lambda\|_\infty^2 (\ell_{\text{xent},s}(f(s)) - \ell^*_{\text{xent},s}) \leq 2\|\Phi^\top \lambda\|_\infty^2 \epsilon_s$. Combining this with the expression for the gradient computed earlier, we get $|\lambda^\top(\Phi p_{f(s)} - \Phi p^*_{\cdot|s})|^2 \leq 2\|\Phi^\top \lambda\|_\infty^2 \epsilon_s$, thus completing the proof of the main lemma. We now prove the lemma.

Proof of Lemma F.7. This is a variant of a classical result used in optimization and we prove it here for completeness. For any $\eta \in \mathbb{R}$ we have

$$\begin{aligned}
\ell(\theta) - \ell^* &\geq^{(a)} \ell(\theta) - \ell(\theta - \eta\lambda) \\
&\geq^{(b)} \ell(\theta) - \left(\ell(\theta) + \langle \nabla \ell(\theta), -\eta\lambda \rangle + \frac{\eta^2}{2} \lambda^\top \nabla^2 \ell(\tilde{\theta}) \lambda\right) \\
&\geq^{(c)} \eta(\lambda^\top \nabla \ell(\theta)) - \frac{\eta^2 L}{2}
\end{aligned}$$

where (a) follows from the definition of infimum and (b) follows from Taylor's expansion and (c) follows from the smoothness condition in the statement of the lemma. Picking $\eta = \frac{\lambda^\top \nabla \ell(\theta)}{L}$ gives us $\ell(\theta) - \ell^* \geq \frac{1}{2L} |\lambda^\top \nabla \ell(\theta)|^2$, thus completing the proof. □

□

Lemma F.8. *For any task $\mathcal{T}$ and classifier $\boldsymbol{v} \in \mathbb{R}^V$ and predicted probabilities $\{p_{\cdot|s}\}$*

$$\ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v}) \leq \ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v}) + \sqrt{\boldsymbol{v}^\top \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}$$

where $\Sigma_{p_{\mathcal{T}}}(g) = \mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}[g(s)g(s)^\top]$ and $\Delta_{\{p_{\cdot|s}\}}(s) = p_{\cdot|s} - p^*_{\cdot|s}$ are defined in Section D.

Proof. The following sequence of inequalities proves it

$$\begin{aligned}
\ell_{\mathcal{T}}(\{p_{\cdot|s}\}, \boldsymbol{v}) = \mathop{\mathbb{E}}_{(s,y)\sim p_{\mathcal{T}}}\left[\ell(\boldsymbol{v}^\top p_{\cdot|s}, y)\right] &\leq^{(a)} \mathop{\mathbb{E}}_{(s,y)\sim p_{\mathcal{T}}}\left[\ell(\boldsymbol{v}^\top p^*_{\cdot|s}, y) + |\boldsymbol{v}^\top(p^*_{\cdot|s} - p_{\cdot|s})|\right] \\
&\leq^{(b)} \mathop{\mathbb{E}}_{(s,y)\sim p_{\mathcal{T}}}\left[\ell(\boldsymbol{v}^\top p^*_{\cdot|s}, y)\right] + \sqrt{\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}\left[\left|\boldsymbol{v}^\top(p^*_{\cdot|s} - p_{\cdot|s})\right|^2\right]} \\
&= \ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v}) + \sqrt{\boldsymbol{v}^\top\left(\mathop{\mathbb{E}}_{s\sim p_{\mathcal{T}}}\left[(p^*_{\cdot|s} - p_{\cdot|s})(p^*_{\cdot|s} - p_{\cdot|s})^\top\right]\right)\boldsymbol{v}} \\
&= \ell_{\mathcal{T}}(\{p^*_{\cdot|s}\}, \boldsymbol{v}) + \sqrt{\boldsymbol{v}^\top \Sigma_{p_{\mathcal{T}}}(\Delta_{\{p_{\cdot|s}\}})\boldsymbol{v}}
\end{aligned}$$

where (a) follows from 1-lipschitzness of ℓ, (b) follows from Jensen's inequality. □

Lemma F.9. *For matrices $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{D\times D}$ s.t. $\boldsymbol{X}, \boldsymbol{Y} \not\succcurlyeq 0$ and $\boldsymbol{Y}$ is full rank, we have that $\max\limits_{\boldsymbol{a}\in\mathbb{R}^D, 0<\|\boldsymbol{a}\|\leq\lambda} \frac{\boldsymbol{a}^\top\boldsymbol{X}\boldsymbol{a}}{\boldsymbol{a}^\top\boldsymbol{Y}\boldsymbol{a}} = \|\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}\|_2$ for any norm $\|\cdot\|$.*

Proof. Note that $\frac{\boldsymbol{a}^\top\boldsymbol{X}\boldsymbol{a}}{\boldsymbol{a}^\top\boldsymbol{Y}\boldsymbol{a}}$ is independent of the scaling of $\boldsymbol{a}$. The following sequence of inequalities completes the proof

$$\begin{aligned}
\max_{\boldsymbol{a}\in\mathbb{R}^D, 0<\|\boldsymbol{a}\|\leq\lambda} \frac{\boldsymbol{a}^\top\boldsymbol{X}\boldsymbol{a}}{\boldsymbol{a}^\top\boldsymbol{Y}\boldsymbol{a}} = \max_{\boldsymbol{a}\in\mathbb{R}^D} \frac{\boldsymbol{a}^\top\boldsymbol{X}\boldsymbol{a}}{\boldsymbol{a}^\top\boldsymbol{Y}\boldsymbol{a}} &= \max_{\boldsymbol{a}\in\mathbb{R}^D} \frac{\boldsymbol{a}^\top\boldsymbol{X}\boldsymbol{a}}{(\boldsymbol{Y}^{\frac{1}{2}}\boldsymbol{a})^\top(\boldsymbol{Y}^{\frac{1}{2}}\boldsymbol{a})} \\
&= \max_{\boldsymbol{a}\in\mathbb{R}^D, \|\boldsymbol{Y}^{\frac{1}{2}}\boldsymbol{a}\|_2=1} \boldsymbol{a}^\top\boldsymbol{X}\boldsymbol{a} = \max_{\boldsymbol{b}\in\mathbb{R}^D, \|\boldsymbol{b}\|_2=1} (\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{b})^\top\boldsymbol{X}(\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{b}) \\
&= \max_{\boldsymbol{b}\in\mathbb{R}^D, \|\boldsymbol{b}\|_2=1} \boldsymbol{b}^\top\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{b} = \|\boldsymbol{Y}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{Y}^{-\frac{1}{2}}\|_2
\end{aligned}$$

□

G Experiment Details

For all experiments, we use the 117M parameter "small" GPT-2 model proposed in Radford et al. [2019] and implemented in HuggingFace Wolf et al. [2019]. We use the standard learning rate schedule and architecture provided in the initial publication. To learn a model on IMDb, we use a context size of 512 BPE tokens, and for the Amazon reviews dataset McAuley et al. [2015], we use the standard context length of 1,024 BPE tokens.

G.1 Solving downstream tasks using f and Φp_f

Details about word subsets: For all of the results presented in Table 1, we use a pre-trained GPT-2 model. For SST, we use the prompt "This movie is " when indicated. For AG News, we use the prompt "This article is about " when indicated.

We compute the conditional probability of selecting a subset of words to complete the sentence. For AG News, this subset is: 'world', 'politics', 'sports', 'business', 'science', 'financial', 'market', 'foreign', 'technology', 'international', 'stock', 'company', 'tech', 'technologies'. For SST, this subset is: ':)', ':(', 'great', 'charming', 'flawed', 'classic', 'interesting', 'boring', 'sad', 'happy', 'terrible', 'fantastic', 'exciting', 'strong'. We account for BPE tokenization by using the encoding of the word directly and the encoding of the word with a space prepended. We then filter to use only words that encode to a single BPE token.

For AG News, the class words we use are: ’foreign’, ’sports’, ’financial’, ’scientific’. For SST, the class words we use are ‘ :)’ and ‘ :(’.

Tests on additional datasets: We also test the performance of GPT-2 frozen embeddings f and the conditional mean embeddings $g_{f,\Phi}(s) = \Phi p_{f(s)}$ on the DBPedia [Auer et al., 2007], Yahoo Answers [Zhang et al., 2015], TREC [Li and Roth, 2002], IMDb [Maas et al., 2011], Customer Review (CR) [Hu and Liu, 2004], and MPQA polarity [Wilson and Wiebe, 2003] datasets in Table 2. We limited the training set size to 200K for larger datasets (i.e., DBPedia and Yahoo Answers). For CR and MPQA, we created train-test splits with 75-25 percentage random splits of the data.

We find that $g_{f,\Phi}$ consistently has comparable performance to f across non-sentiment and sentiment downstream classification tasks. We include results using a bag-of-n-grams (BoNG) and Sentiment Neuron (mLSTM) [Radford et al., 2017], and we note that using 768-dimensional features is more sample efficient than BoNG.

For sentiment tasks, adding a prompt always boosts performance. We also demonstrate that much of the performance can be recovered by only looking at “positive” and “negative” or “:)” and “:(” as class words. Using these 2-dimensional features is even more sample-efficient than the standard 768-dimensional ones.

We also include results using the pre-trained BERT base cased model [Devlin et al., 2018, Wolf et al., 2019], using the embedding at the first token as input to the downstream task. We also tried using the mean embedding and last token embedding and found that the first token embedding is best. Moreover, the first token embedding is what is extracted in the traditional usage of BERT on downstream tasks.

Table 2: GPT-2 performance without fine-tuning on downstream task test sets with k classes. We provide the performance of bag-of-n-grams as an approximate baseline for these tasks. DBPedia and Yahoo performances were reported in Zhang et al. [2015], and the other tasks were reported in Khodak et al. [2018]. We also include results from Sentiment Neuron [Radford et al., 2017] for the sentiment-related classification tasks: IMDb, CR, and MPQA. Furthermore, we include results from using BERT [Devlin et al., 2018] without fine-tuning, where we use the features produced for the first position as input to the linear classifier. An asterisk indicates we add a standard sentiment prompt “The sentiment is ” to each input. We also tested the performance of the conditional probability distribution over “positive” and “negative” as well as “ :)” and “ :(” on the sentiment-related tasks with and without the prompt.

Task	k	$f(s)$	$\Phi p_{f(s)}$	$p_{\cdot\|s}$: pos,neg	$p_{\cdot\|s}$: :),:(	BonG	mLSTM	BERT
Non-sentiment								
DBPedia	14	96.1%	88.5%	-	-	98.6% $(n = 5)$	-	98.3%
Yahoo	10	69.9%	57.8%	-	-	68.5% $(n = 5)$	-	64.7%
TREC	6	94.2%	88.0%	-	-	89.8% $(n = 3)$	-	91.0%
Sentiment								
IMDb	2	87.7%	83.0%	76.1%	72.3%	89.8% $(n = 3)$	92.3%	81.9%
IMDb*	-	87.8%	84.3%	77.8%	74.3%	-	-	83.7%
CR	2	92.3%	85.5%	80.0%	73.8%	78.3% $(n = 3)$	91.4%	90.5%
CR*	-	92.4%	90.5%	79.6%	81.4%	-	-	88.3%
MPQA	2	87.9%	82.1%	71.0%	70.5%	85.6% $(n = 3)$	88.5%	88.3%
MPQA*	-	88.5%	87.1%	71.6%	78.4%	-	-	88.4%

G.2 Testing Quad objective

We first compare downstream performance of the Quad objective to the cross-entropy objective by training GPT-2 on the IMDb dataset [Maas et al., 2011]. Table 3 shows that features learned by Quad perform comparably to $g_{f,\Phi}$ for f learned by the cross-entropy objective, which fits our theory since both are linear functions of $p^*_{\cdot|s}$.

Table 3: Comparing Quad features to standard cross-entropy features for GPT-2 trained on IMDb [Maas et al., 2011].

Task	$f(s)$ (xent)	$g_{f,\Phi}(s)$ (xent)	$f(s)$ (Quad, fixed Φ)	$f(s)$ (Quad, learned Φ)
SST	82.1%	79.9%	77.1%	77.3%
SST*	83.1%	81.1%	78.5%	80.7%

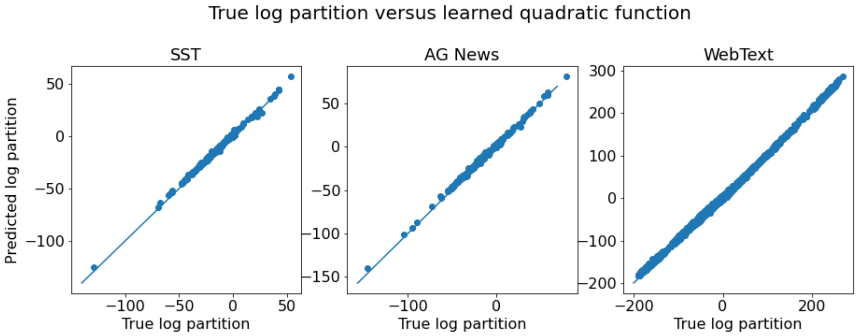

Figure 1: Fit of the learned quadratic function to the log partition function on various datasets for features computed by the full, pre-trained GPT-2. We also plot the $y = x$ line for reference.

We then test two models with the same parametrization and initializations, one trained using our Quad objective and another trained with the standard language modeling objective using the Amazon product review dataset [McAuley et al., 2015] instead of IMDB. We slightly modify the standard architecture of GPT-2 to generate Tables 3 and 4. We add a single linear layer after the Transformer to add expressivity. Furthermore, instead of tying the input and output embeddings, we learn them separately so that f and Φ are independent functions. We fix the input embeddings and the positional embeddings to be the parameters from the pre-trained GPT-2. We initialize Φ, the output embeddings, using the singular vectors of the pre-trained word embeddings Φ. Given our parameterization, initializing with the singular vectors is as expressive as initializing with the pretrained embeddings Φ themselves; however it lends a better optimization landscape and speeds up training for our new objective.

We observe that even on a large dataset, training using Quad yields comparable performance to the language model on the SST task. Furthermore, adding a prompt consistently improves performance for both objectives.

Table 4: Comparing the downstream performance of features learned using Quad to $\Phi p_{f(s)}$, where $f(s)$ is from an LM trained on the standard KL objective. All models were trained on the Amazon dataset. An asterisk indicates that we added the prompt "This movie is " to each input. Note that the validation loss was still decreasing at the time of measurement.

Task	$f(s)$ (xent)	$\Phi p_{f(s)}$ (xent)	$f(s)$ (Quad, learned Φ)
SST	89.4%	89.7%	79.2%
SST*	89.7%	89.2%	84.3%

G.3 Learning the quadratic approximation of the log-partition function

In Assumption 4.1, we assert that there is a quadratic fit for the log partition function, which allows us to show in Lemma 4.3 that a linear relation holds between f^* and Φp_{f^*}. We validate these theoretical findings by fitting a quadratic function to the log partition function for a subset of embeddings from the IMDb, SST, and AG News datasets (Figure 1). Here, we describe how we learned $\boldsymbol{A}$, $\boldsymbol{b}$ and c. To ensure $\boldsymbol{A}$ is symmetric and positive semi-definite as required, we parametrize $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{U}^T$. Let $\mu_\theta = \Phi p_\theta$. We minimize the following objective function:

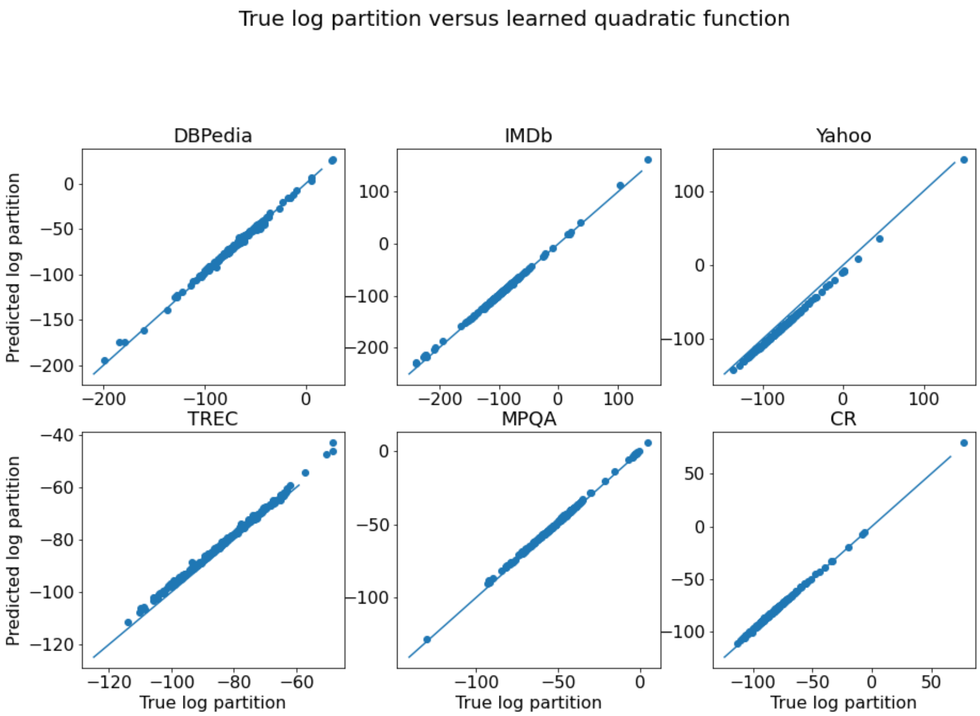

Figure 2: Fit of the learned quadratic function to the log partition function on various datasets for features computed by the full, pre-trained GPT-2. We also plot the $y = x$ line for reference.

$$\mathcal{L}(\boldsymbol{U},\boldsymbol{b},c) = \mathop{\mathbb{E}}_{\theta}\left[\lambda_1\left(\log(Z_\theta) - \frac{1}{2}\theta^\top \boldsymbol{U}\boldsymbol{U}^\top\theta - \theta^\top\boldsymbol{b} - c\right)^2 + \lambda_2\left\|\Phi p_\theta - \boldsymbol{U}\boldsymbol{U}^\top\theta - \boldsymbol{b}\right\|^2\right]$$

In practice, we train only on the regression loss (i.e., $\lambda_1 = 0$, $\lambda_2 = 1$) for the most promising results. We use 20,000 examples from a mix of IMDb, SST, and AG News embeddings as the training set. We used the Adam [Kingma and Ba, 2014] optimizer with learning rate 1e-3 for $\boldsymbol{U}$ and learning rate 1e-4 for $\boldsymbol{b}$ and c. We decayed the learning rate every 50 steps by a factor of 0.1. We found the fit after 8 epochs of training.

We further demonstrate the quality of the learned fit by plotting the true log partition and estimated log partition function for embeddings from other datasets in Figure 2.