Improving Language Model Fine-tuning with Information Gain Filtration*

Javier S. Turek
Intel Labs
javier.turek@intel.com

Richard Antonello
UT Austin
rjantonello@utexas.edu

Nicole M. Beckage
Intel Labs
nicole.beckage@intel.com

Alexander G. Huth
UT Austin
huth@cs.utexas.edu

Abstract

Language model fine-tuning is essential for modern natural language processing. The effectiveness of fine-tuning is limited by the inclusion of training examples that negatively affect performance. Here we present *Information Gain Filtration*, a general fine-tuning method, for improving the overall final performance of a fine-tuned model. We define Information Gain of an example as the improvement on a validation metric after training on that example. A secondary learner is then trained to approximate this quantity. During fine-tuning, this learner filters informative examples from uninformative ones. We show that our method is robust and has consistent improvement across datasets, fine-tuning tasks, and language model architectures.

1 Introduction

Language modeling is the task where a model predicts the conditional probability of the next token based on the context of previously observed tokens. Recent advances in transformers-based models (Vaswani et al., 2017), lead to language modeling success as a pre-training objective for self-supervised representation learning. Once pre-trained, language models (LMs) can be updated for downstream tasks through fine-tuning (Devlin et al., 2019; Radford et al., 2019). Hence, improving fine-tuning leads to higher quality models.

Several methods have been proposed to improve LM fine-tuning performance. These include regularization techniques Lee et al. (2020), supplementary training on supervised tasks Phang et al. (2018), incorporating out of domain data Moore and Lewis (2010), and using features from intermediate transformer layers Tenney et al. (2019); Liu et al. (2019). In addition, its instability of this process has been investigated with relation to insufficiently general training sets (Mosbach et al., 2021), and to optimization techniques (Zhang et al., 2021).

Recently, Dodge et al. (2020) showed that the fine-tuning process has high variability between runs being sensitive to data ordering. They reduce this variability by fine-tuning models using many random seeds and keeping the best. While this improves performance, the reasons for the high variability between random seeds are unknown.

In this work, we propose a novel approach to improving the effectiveness of fine-tuning by carefully selecting "informative" samples. Our approach uses a secondary learner to estimate the usefulness of each example, and then selects only informative examples for fine-tuning. We show that this technique works well and is applicable in a variety of settings. We further analyze the secondary learner capabilities.

2 Background

A language model L is a function with parameters θ, which, when given an ordered sequence of tokens $X = \{x_1, \ldots, x_n\}$ as input, outputs a probability distribution over the next token y, $L(X;\theta) = \hat{p}(y|X)$. Given a test set $\mathcal{T}$ of (sequence, next token) pairs, $\mathcal{T} = \{(X_1, y_1), \ldots, (X_n, y_n)\}$, the perplexity $\Lambda(\mathcal{T};\theta)$ of the language model $L(X;\theta)$ over the set $\mathcal{T}$ is defined as $\Lambda(\mathcal{T};\theta) = 2^{-\sum_{(X_i,y_i)\in\mathcal{T}} \bar{p}(y_i)\cdot\log_2 L(X_i;\theta)}$, where $\bar{p}(y_i)$ denotes the one-hot probability distribution that assigns all of its probability mass to the token y_i. LMs are trained to minimize perplexity on very large training corpora.

In practice, pre-trained LMs are often fine-tuned using a new corpus or transferred to a new task (Howard and Ruder, 2018). Formally, let $\mathcal{F} = \{(X_i, y_i)\}_i$ be a target set. Fine-tuning on the set $\mathcal{F}$ minimizes the expected value of the loss Λ:

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}(\log_2 \Lambda(\mathcal{F};\theta)). \quad (1)$$

*This work has been previously published as part of ACL.

The initial parameterization $\hat{\theta}_0$ of the LM is defined by its pre-trained parameters $\hat{\theta}_0 = \theta$. The fine-tuning problem in Eq. (1) is solved by applying stochastic gradient descent (SGD) on samples from $\mathcal{F}$. We refer to methods that randomly sample contexts to update pretrained model parameters as *standard fine-tuning*.

3 Information Gain Filtration

3.1 Informativeness of an Example

Given a pre-trained language model $L(X;\theta)$ and a target dataset $\mathcal{F}$, we define the informativeness of an example $(X, y) \in \mathcal{F}$ as the improvement that it will grant to the model. Namely, we define the *information gain* (IG) of (X, y) over an objective set $\mathcal{O}$ as the difference in perplexity measured on $\mathcal{O}$ before and after training on (X, y),

$$IG_{\mathcal{O}}(X, y) = \Lambda(\mathcal{O}; \theta'(X, y)) - \Lambda(\mathcal{O}; \theta), \quad (2)$$

where θ is the initial parameterization of the LM and $\theta'(X, y)$ is the parameterization after training with the example (X, y). The objective set $\mathcal{O} = \{(X_1, y_1), \ldots, (X_n, y_n)\}$ is a held-out subset of training data that informs our decision about which contexts are informative.For brevity, we denote $IG_{\mathcal{O}}(X, y)$ as $IG(X)$. In practice, the objective set could be a subset of the set $\mathcal{F}$.

3.2 Filtering Examples

Next, we propose Information Gain Filtration (IGF), a new method, that exploits $IG(X)$ for fine-tuning. Given a new example (X, y) the method chooses between a) updating the model parameters θ backpropagating (X, y), and b) skipping it, leaving the model parameters unchanged. For this purpose, we define the function $q(X, action)$ and assign a value to each of the actions above:

$$q(X, \textsc{Backprop}) = IG(X) \quad (3)$$

$$q(X, \textsc{Skip}) = T_{\textsc{Skip}}, \quad (4)$$

where $T_{\textsc{Skip}}$ is a free "threshold" parameter for deciding which $IG(X)$ values are sufficiently high to warrant a model update. Following this definition, we apply a greedy policy for filtering examples during fine-tuning: $\pi(X) = \arg\max_{a \in \{\textsc{Backprop}, \textsc{Skip}\}} q(X, a)$. By filtering examples in this way, we aim to reduce the variability effect in data order (Dodge et al., 2020), and improve the generalizability of our training set (Mosbach et al., 2021).

3.3 Approximating Information Gain

Computing $IG(X)$ in Eq. (2) entails a backpropagation step, making direct application of $q(X, action)$ as expensive as standard fine-tuning. Thus, we approximate $IG(X)$ with a *secondary learner* model $\hat{Q}(X)$. First, we construct a training dataset $\mathcal{D}$ by drawing a random subset of examples from the fine-tuning set $\mathcal{F}$ and measuring $IG(X)$ on the objective set $\mathcal{O}$. Each entry in $\mathcal{D}$ is of the form $(X_i, IG(X_i))$. Using $\mathcal{D}$, we train the secondary learner $\hat{Q}$ to predict a normalized $IG(X)$. Normalization helps standarize the threshold $T_{\textsc{Skip}}$ during filtration. The resulting $\hat{Q}$ is applied to filter examples for fine-tuning.

The effectiveness of the learner at distinguishing "high quality" from "low quality" examples should degrade as the parameters diverge from their initial values θ_0 used for constructing $\mathcal{D}$. To ameliorate this problem, we modify the threshold $T_{\textsc{Skip}}$ during the fine-tuning process. Since $\hat{Q}$ is most accurate at the first step, we switch from highly selective (a high value) to highly permissive (a low value). This allows the model to take advantage of the accurate predictions for $IG(X)$ early in the fine-tuning process.

4 Experimental Results

We next analyze IGF's performance across different choices of datasets, fine-tuning tasks, and models. We tested these results on a Books dataset (Zhu et al., 2015), a "Mixed" dataset composed from the Books and a corpus of scraped Reddit comments (Huth et al., 2016), and WikiText-103 (Merity et al., 2017). The Books corpus allows us to fairly compare standard fine-tuning against IGF, whereas the Mixed corpus allows us to analyze the effectiveness of the method at separating informative contexts from uninformative ones. We implement standard fine-tuning with Adam (Kingma and Ba, 2015). Our secondary learner, $\hat{Q}$, represents the input text X by embedding it with 768-dimensional byte-pair embeddings (Gage, 1994), followed by a convolution with kernel width 3, a max-pooling operation over the time axis, and a 2-layer network.

4.1 Fine-tuning Performance

We compare the performance of IGF directly to standard fine-tuning on the pre-trained GPT-2 Small model (Wolf et al., 2020). Figure 1 depicts the results of fine-tuning the model on the Mixed

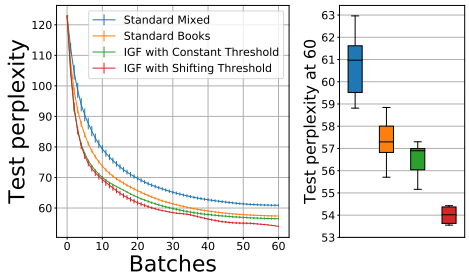

Figure 1: Performance comparison of standard fine-tuning and IGF of GPT-2 Small on the mixed corpus over 50 runs. IGF obtains the best performance (red).

dataset (batch size 16, learning rate 5×10^{-5}, Adam $\beta_1 = 0.9, \beta_2 = 0.999$). As GPT-2 was trained originally including Reddit data, we expect some Mixed dataset examples to be uninformative. Hence, we included standard fine-tuning on the Books corpus as a reference run with more informative data. Standard fine-tuning on Books (orange) achieves a median perplexity of 57.3, compared to 56.9 for IGF with a constant threshold and 54.0 for IGF with a shifting threshold.

We further tested IGF with shifting threshold across several choices of dataset, fine-tuning specifications, and model architecture. We tested on WikiText-103 when fine-tuning GPT-2 Small (Perplexity: IGF 67.8 vs 69.8), GPT-2 Medium (Perplexity: IGF 27.1 vs 27.4), BERT (Devlin et al., 2019) (Masked perplexity IGF 4.29 vs 4.33), and on Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) (Accuracy IGF 94.27 vs 94.06). In every case, IGF exceeds the performance of standard fine-tuning, suggesting that it is a method broadly applicable to a variety of modalities and domains.

4.2 Understanding the Secondary Learner

As the secondary learner aims to approximate the informativeness of a sample, we analyze next the quality of such approximation. For this purpose, we created a dataset of 10,000 $(X, IG(X))$ pairs from the Mixed corpus using an objective set of 160 contexts with 32 tokens each drawn solely from the Books corpus. Then, we trained the secondary learner on this dataset and tested it on randomly sampled contexts from the Mixed corpus. Because the objective set contains only examples from one corpus, we expect the secondary learner to assign higher $IG(X)$ values to other examples from the same corpus. Figure 2 shows that there is a significant difference in the distributions of $\hat{Q}$ values between the two corpora, demonstrating its separating capabilities.

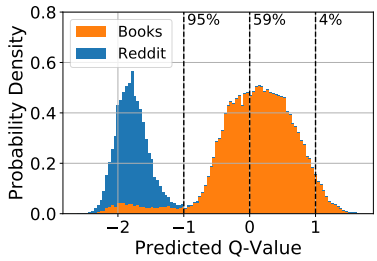

Figure 2: Normalized predicted Q(X) values by the secondary learner. Good separation is achieved using the information gain $IG(X)$ metric despite computing the true q-value using a small objective set.

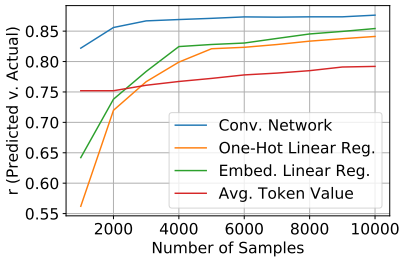

Figure 3: Comparison of the sample efficiency of secondary learners as a function of the training size. Correlation coefficient of the prediction vs. ground truth.

Above, we used a convolutional neural network as $\hat{Q}$. Here we explore how other simpler methods perform. We encoded the contexts both by using the standard GPT-2 Small word embedding, and with a one-hot encoding of the token identities. Standard linear regression performed on both encoding types (30K parameters for word embeddings and 450K parameters for one-hot encoding) performs nearly as well at approximating $IG(X)$ with a convolutional model. We also tested a learner with only 25K parameters that assigned each token a value by averaging the $IG(X)$ values for contexts that contained that token. Figure 3 compares the performance of these architectures across different training data sizes. The convolutional network is the most sample efficient method, as it can effectively learn $IG(X)$ with as few as 2K training examples.

5 Conclusion

In the context of LM fine-tuning, we have shown that a secondary learner can efficiently and effectively distinguish between informative and uninformative training examples. This secondary learner can select useful training examples in a method we call Information Gain Filtration, leading to better model performance than standard fine-tuning.

References

Léon Bottou. 1991. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Alexander G Huth, Wendy A De Heer, Thomas L Griffiths, Frédéric E Theunissen, and Jack L Gallant. 2016. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532(7600):453–458.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020. Mixout: Effective regularization to finetune large-scale pretrained language models. In *International Conference on Learning Representations*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 3645–3650. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning*, 8(3-4):279–292.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame,

Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *International Conference on Learning Representations*.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.