Bilevel Optimization to Learn Training Distributions for Language Modeling under Domain Shift

David Grangier, Pierre Ablin, Awni Hannun
Apple, {grangier,pablin,awni}@apple.com

Abstract

Language models trained on very large web corpora have become a central piece of modern language processing. In this paradigm, the large, heterogeneous training set rarely matches the distribution of the application domain. This work considers modifying the training distribution in the case where one can observe a small sample of data reflecting the test conditions. We propose an algorithm based on recent formulation of this problem as an online, bilevel optimization problem. We show that this approach compares favorably with alternative strategies from the domain adaptation literature[1].

1 Introduction

Large models pretrained from massive, heterogeneous datasets have impacted various application domains [5], including natural language processing [9], computer vision [28] and audio processing [38]. These models are typically trained on two different distributions, a *generic distribution* for pretraining and a *specific distribution* for fine tuning. Only the specific distribution matches the test conditions while the generic distribution offers an abundant source of data with some similarities to the specific data. This novel paradigm builds upon multitask learning [6], transfer learning [4] and domain adaptation [29]. For all these methods, the accuracy of a model on the specific task crucially depends on selecting an appropriate distribution over generic auxiliary tasks and data.

Prior work has proposed automatic methods to adjust the generic training distribution in order to improve generalization on the specific task. The domain adaptation literature has explored variants of importance sampling, using importance weights to emphasize or select some generic examples. These weights have been determined via domain classifiers [1, 14], via gradient alignment and fine-tuning [42, 13] or via the estimation of the label distribution [30]. Related to domain adaptation, the removal of label noise in the generic distribution has received attention with methods based on influence functions [22, 33, 37], data models [17, 18] and data Shapley values [12, 20].

As an alternative to static weighting, the literature also explored dynamic weighting where the distribution over generic examples is adapted during training. Two main strategies have been leveraged: reinforcement learning and direct optimization. Reinforcement learning does not assume that the specific task loss can be differentiated with respect to the weighting parameters. Instead, a parameterized model of the generic distribution is adjusted through reinforcement: the current model proposes generic distributions and their reward is measured as the specific loss after a few steps of generic training over a proposal distribution [23, 44, 48]. On the other hand, direct optimization assumes a differentiable functional dependency between the weighting parameters and the specific training loss. This dependency can be derived through meta learning by unfolding the generic update [35, 16, 39, 46]: one gradient update step minimizing the weighted generic loss depends on the weighting parameters. This update can be evaluated by computing the post update specific loss which can then be differentiated with respect to the weighting parameters. As an alternative to update

[1] arXiv:2311.11973 shows an extended version of this work with additional datasets, tasks and analyses.

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

unfolding, a bilevel formulation of the reweighting problem also allows for direct optimization [10]. Our work builds upon this formulation: we propose an online optimization strategy that leverages the cost asymmetry between the inexpensive evaluation of training example weights and the costly evaluation of the gradient of a large language model at these points.

Other research areas intersect with generic sample reweighing. Prior work considered learning a distribution over training data augmentation [15, 26, 49]; curriculum learning has been introduced to visit successive training distributions based on training instance difficulty [3, 24, 19, 36]; multi-task learning research has considered gradient projection to minimize negative interactions between tasks [45, 8, 27]. Importance sampling for accelerated stochastic training [47, 21] is also relevant.

Our work has multiple contributions. Our work (i) formalizes data selection as a bilevel optimization problem, (ii) connects Differentiable Data Selection (DDS) and Stochastic Bilevel Algorithm (SOBA) which were proposed independently, (iii) introduces an online algorithm to perform data selection while training large models. Finally, (iv) we demonstrate the benefit of our method through an empirical comparison with the main alternative strategies.

2 Method

We aim to identify the parameters θ of a neural language model that achieves good generalization performance (held-out likelihood) over the specific distribution. For that purpose, we are given a large generic training set $\mathcal{D}_{\text{generic}}$ and small specific training set $\mathcal{D}_{\text{specific}}$. Only the latter set is representative of the test conditions. The generic training problem minimizes the weighted loss,

$$\mathcal{L}_{\text{generic}}(\theta, \alpha) = \sum_{x \in \mathcal{D}_{\text{generic}}} w(x; \alpha) \ell(x; \theta)$$

where $w(x; \alpha)$ denotes a smaller, secondary *weighting neural network* which defines a distribution over $\mathcal{D}_{\text{generic}}$, i.e. $\forall x, w(x; \alpha) > 0$ and $\sum_{x \in \mathcal{D}_{\text{generic}}} w(x; \alpha) = 1$. We denote the solution to the generic training problem as

$$\theta^*(\alpha) \in \arg\min_{\theta} \mathcal{L}_{\text{generic}}(\theta, \alpha) \tag{1}$$

Our goal is to find the parameter of the weighting network such that the loss on the *specific* training set is minimal, i.e. minimizing the following with respect to α,

$$\mathcal{L}_{\text{specific}}(\theta^*(\alpha)) := \sum_{x' \in \mathcal{D}_{\text{specific}}} \ell(x'; \theta^*(\alpha)). \tag{2}$$

2.1 Data Selection as a Bilevel Optimization Problem

Our notations make clear that finding the optimal weighting network can be cast as a bilevel optimization problem: with a fixed weighting network, the optimal parameters for the main model are found by minimizing the weighted loss over the generic dataset, Eq. (1). The optimal main model parameters θ^* depends explicitly on the weighting network parameters α; indeed, changing α changes the optimization problem in (1) and its solution. The selection of α is driven by the specific set loss, Eq. (2).

Equations (1) and (2) form a *bilevel optimization problem* [10]: the outer problem (2) depends implicitly on α through the solution to the inner problem (1). One of the strengths of such a bilevel formulation is that the weighting network must adapt to the main model: the question is to learn a weighting network such that the main model trained with that network leads to good specific performance. This has the potential to go beyond a simple model-agnostic scheme that would, for instance, build $w(x)$ based on the similarity between x and the specific set. While a large body of the literature is devoted to solving bilevel problems where the inner problem (1) is convex in θ [11, 2], in our case (1) corresponds to the training problem of a neural network which is non-convex. This leads to several difficulties:

- The $\arg\min$ in (1) is not a single element since there are multiple minimizers. Therefore, the function $\theta^*(\alpha)$ is not properly defined.
- In order to use gradient-based methods to find the optimal α, we have to compute the approximate Jacobian of $\theta^*(\alpha)$. This is usually done using the implicit function theorem, which only applies when the loss function in (1) is locally convex and such property is hard to check in practice.

Furthermore, we want a method with a computational cost similar to the standard training of the main model. In other words, we have enough budget to solve (1) only once: learning α and θ must be carried out synchronously. This has an important consequence: the bilevel methods that we study update α based on the current main model state θ and not on the optimal solution $\theta^*(\alpha)$. Hence, this is a slight deviation from the bilevel formalism. This also means that the weighting network adapts to the current state of the main model and, ideally, tries to up-weight generic data that is useful *at the current state of learning*. We explore online algorithms to solve the bilevel problem when the main language model is large. These algorithms alternate θ and α updates and leverage the asymmetry in computation cost between evaluating the large language model and the small weighting network.

2.2 Updating the main model: the big-batch trick.

To update the main model, we fix α and do a step to minimize (1). A first, natural idea would be to take a mini-batch of generic data B_{generic} of size b, compute the corresponding gradient $g = \frac{1}{b}\sum_{x\in B_{\text{generic}}} w(x;\alpha)\nabla_\theta \ell(x;\theta)$ and then use it to update θ, either implementing SGD by doing $\theta \leftarrow \theta - \eta \times g$ with $\eta > 0$ a learning rate, or by using it into a more involved optimizer like Adam. However, the computation of g with the previous equation can be wasteful when a significant fraction of the examples of B_{generic} are assigned small weights $w(x;\alpha)$. These examples do not contribute much to g while still requiring the expensive computation of their gradient $\nabla_\theta \ell(x;\theta)$.

To accelerate the optimization of θ, we leverage the asymmetry between the cost of evaluating the weighting network and the main model: computing $w(x;\alpha)$ only requires inference of a small network while computing $\nabla\ell(x;\theta)$ requires inference *and* back-propagation through a large network. We start by sampling a large batch $B^{\text{big}}_{\text{generic}}$ from the generic dataset and compute $w(x;\alpha)$ for each x in $B^{\text{big}}_{\text{generic}}$. From there we can take a smaller batch $B^{\text{small}}_{\text{generic}}$ from $B^{\text{big}}_{\text{generic}}$, either by sampling from the distribution defined by $w(x;\alpha)$ or by taking the examples with the highest $w(x;\alpha)$. The first option is an unbiased solution corresponding to importance sampling, while the second option is biased but observed to work better in practice. In both cases, we compute the gradient to update θ with uniform weights, using $g = \frac{1}{b}\sum_{x\in B^{\text{small}}_{\text{generic}}} \nabla_\theta \ell(x;\theta)$.

2.3 Updating the weighting model

We consider two alternatives to update the weighting model. With scalability in mind, we only consider *stochastic* methods, i.e., that update the weighting network parameters α using only a mini-batch of specific data B_{specific} and a mini-batch of generic data B_{generic}.

2.3.1 One gradient step unrolling - differentiable data selection (DDS)

This method is similar to [43], and updates the weighting network by doing a descent step on the loss

$$\mathcal{L}(\alpha) = \sum_{x'\in B_{\text{specific}}} \ell'(x'; u(\theta,\alpha)) \text{ with } u(\theta,\alpha) = \theta - \rho \times \sum_{x\in B_{\text{generic}}} w(x;\alpha)\nabla\ell(x,\theta), \tag{3}$$

which corresponds to the value of the specific loss on the mini-batch B_{specific} after a gradient descent step for θ on the generic mini-batch B_{generic} using the current weights. The idea behind this method is that $u(\theta,\alpha)$ is a reasonable approximation to $\theta^*(\alpha)$. This method requires backpropagating through a gradient descent step, which requires only a little overhead compared to a standard gradient computation. In the limit where the step size ρ in the gradient update $u(\theta,\alpha)$ goes to 0, we see that $\mathcal{L}(\alpha) \simeq \rho\langle g_{\text{specific}}, g_{\text{generic}}\rangle$, with $g_{\text{specific}} = \sum_{x'\in B_{\text{specific}}} \nabla\ell'(x';\theta)$ and $g_{\text{generic}} = \sum_{x\in B_{\text{generic}}} w(x;\alpha)\nabla\ell(x,\theta)$. Hence, the loss $\mathcal{L}$ approximately measures the alignement between specific and generic gradients. Taking derivatives gives $\nabla\mathcal{L}(\alpha) \simeq \rho\sum_{x\in B_{\text{generic}}}\langle g_{\text{specific}}, \nabla\ell(x,\theta)\rangle\nabla w(x;\alpha)$.

2.3.2 Stochastic Bilevel Algorithm (SOBA)

We also implement the SOBA method of [7], which is a scalable method to solve the bilevel problem, developed in a setting where the inner function (1) is convex. This algorithm approximates a gradient descent on $h(\alpha) = \mathcal{L}_{\text{specific}}(\theta^*(\alpha))$. The chain rule gives $\nabla h(\alpha) = \frac{\partial\theta^*}{\partial\alpha}\nabla\mathcal{L}_{\text{specific}}(\theta^*(\alpha))$. The optimum $\theta^*(\alpha)$ satisfies the first order condition $\nabla_\theta\mathcal{L}_{\text{generic}}(\theta^*(\alpha),\alpha) = 0$. Under the assumption that the Hessian $\nabla^2_{\theta\theta}\mathcal{L}_{\text{generic}}(\theta^*(\alpha),\alpha)$ is invertible, the implicit function theorem applied to the

Table 1: Model architecture

Language model
Transformer decoder with 12 layers, 8 attention heads, residual dimension of 256, feed-forward latent dimension of 1,024.
Weight model
Convolutional network with 2 layers followed by mean pooling, latent dimension of 128.

Table 2: Log-perplexity on specific (Reuters).

Method	Pre-train	Fine-tune
Baseline	1.197	0.864
Mixing	0.860	0.846
CDS	1.071	0.830
Domain classif.	1.099	0.892
MetaWeightNet	1.212	0.867
LTR	1.150	0.877
Sparse DDS	1.033	0.822
Sparse SOBA	**1.018**	**0.819**

previous equation gives $\frac{\partial \theta^*}{\partial \alpha} = -\nabla^2_{\alpha\theta}\mathcal{L}_{\text{generic}}(\theta^*(\alpha),\alpha)\left[\nabla^2_{\theta\theta}\mathcal{L}_{\text{generic}}(\theta^*(\alpha),\alpha)\right]^{-1}$, which overall yields $\nabla h(\alpha) = -\nabla^2_{\alpha\theta}\mathcal{L}_{\text{generic}}(\theta^*(\alpha),\alpha)\left[\nabla^2_{\theta\theta}\mathcal{L}_{\text{generic}}(\theta^*(\alpha),\alpha)\right]^{-1}\nabla\mathcal{L}_{\text{specific}}(\theta^*(\alpha))$. SOBA approximates this quantity in two ways: first, $\theta^*(\alpha)$ is replaced by the current iterate θ in the above gradient. Second, in addition to θ and α, SOBA has an additional variable v of the same size as θ that keeps track of the quantity $-\left[\nabla^2_{\theta\theta}\mathcal{L}_{\text{generic}}(\theta,\alpha)\right]^{-1}\nabla_\theta\mathcal{L}_{\text{specific}}(\theta)$. This is done using the stochastic iterations $v \leftarrow v - \eta \times dv$ with $dv = \sum_{x\in B_{\text{generic}}} w(x;\alpha)\nabla^2\ell(x;\theta)v + \sum_{x'\in B_{\text{specific}}}\nabla\ell'(x';\theta)$. The first part in dv is a Hessian-vector product that can be computed efficiently at a cost similar to that of a gradient [32]. Then, the parameters α are moved in the direction $d\alpha = \sum_{x\in B_{\text{generic}}}\langle\nabla\ell(x;\theta),v\rangle\nabla w(x;\alpha)$, which is a stochastic approximation of $\nabla^2_{\alpha\theta}\mathcal{L}_{\text{generic}}(\theta,\alpha)v$, which is itself an approximation of $\nabla h(\alpha)$.

3 Experimental Evaluation

Our language modeling experiments relies on the C4 dataset [34] as the generic set and the RCV1 [25] as the specific set. C4 is a dataset of English language web pages [31], while RCV1 consists of Reuters newswire stories. This setup is representative of a generic large corpus spanning different types of examples (C4) while the specific task contains an homogeneous set of examples from the same domain and from the same source (RCV1). Our setup uses 30m examples from C4 and 10k examples from RCV1. We compare our results with contrastive data selection, CDS [40, 42], domain classifier selection [14], meta-weight net [39]). and learning to re-weight, LTR [35]. We also consider mixing, i.e. training with a trade-off between the specific and generic loss. We denote bilevel methods with Sparse DDS and Sparse SOBA, to highlight the difference in size between $B^{\text{small}}_{\text{generic}}$ and $B^{\text{big}}_{\text{generic}}$. Our experiments rely on a size ratio of $1/8$.

Our language model is a byte-level language model based on the transformer [41]. The weighting network is a small convolutional network. Table 1 gives architectural details. We also use the same architecture for the domain classifier baseline. We report performance in terms of log-perplexity, i.e. negative log likelihood.

Table 2 reports two types of results. Pretraining results rely on the specific set only to adjust the weighting network, while the language model is trained solely on the weighted generic loss (except for the mixing method). The second set of results evaluate whether the benefit from the adjusted pretraining generic distribution can be complementary to the most common domain adaptation technique, i.e. fine tuning the model on the specific set after pretraining. In both cases, bilevel methods are advantageous and SOBA performs better than DDS.

4 Conclusions

We presented a bilevel optimization strategy to learn training distributions for learning language models in situation where most of the training comes from a distribution different from the targeted test distribution. Our proposal emphasizes scalability by (i) considering a distribution parameterized with a neural network much smaller by the learned language model and (ii) an online algorithm which lets the optimization of the language model focus only on the most beneficial examples. Its advantage is demonstrated empirically to alternative domain adaptation strategies. We plan to investigate if our approach can also be beneficial in other application domains in the future.

References

[1] Roee Aharoni and Yoav Goldberg. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online, July 2020. Association for Computational Linguistics.

[2] Michael Arbel and Julien Mairal. Amortized implicit differentiation for stochastic bilevel optimization. *arXiv preprint arXiv:2111.14580*, 2021.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM, 2009.

[4] Paul N Bennett, Susan T Dumais, and Eric Horvitz. Inductive transfer for text classification using generalized reliability indicators. In *Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*, 2003.

[5] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2021.

[6] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[7] Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. In *NeurIPS*, 2022.

[8] Lucio M Dery, Yann Dauphin, and David Grangier. Auxiliary task update decomposition: The good, the bad and the neutral, 2020.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[10] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

[11] Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

[12] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[13] David Grangier and Dan Iter. The trade-offs of domain adaptation for neural language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3802–3813. Association for Computational Linguistics, 2022.

[14] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*, 2020.

[15] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR, 2019.

[16] Zhiting Hu, Bowen Tan, Russ R Salakhutdinov, Tom M Mitchell, and Eric P Xing. Learning data manipulation for augmentation and weighting. *Advances in Neural Information Processing Systems*, 32, 2019.

[17] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. In *ICML*, 2022.

[18] Saachi Jain, Hadi Salman, Alaa Khaddaj, Eric Wong, Sung Min Park, and Aleksander Madry. A data-based perspective on transfer learning. *arXiv preprint arXiv:2207.05739*, 2022.

[19] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR, 2018.

[20] Bojan Karlaš, David Dao, Matteo Interlandi, Bo Li, Sebastian Schelter, Wentao Wu, and Ce Zhang. Data debugging with shapley importance over end-to-end machine learning pipelines. *arXiv preprint arXiv:2204.11131*, 2022.

[21] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

[22] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[23] Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. Reinforcement learning based curriculum optimization for neural machine translation. *arXiv preprint arXiv:1903.00041*, 2019.

[24] M. Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[25] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

[26] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32, 2019.

[27] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

[28] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11206 of *Lecture Notes in Computer Science*, pages 185–201. Springer, 2018.

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

[30] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018.

[31] Jay M Patel. Introduction to common crawl datasets. In *Getting Structured Data from the Internet*, pages 277–324. Springer, 2020.

[32] Barak A Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

[33] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930, 2020.

[34] 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 e-prints*, 2019.

[35] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

[36] Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. Data parameters: A new family of parameters for learning a differentiable curriculum. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[37] Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8179–8186, 2022.

[38] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

[39] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019.

[40] Marlies van der Wees, Arianna Bisazza, and Christof Monz. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[42] Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[43] Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Jaime Carbonell, and Graham Neubig. Optimizing data usage via differentiable rewards. In *International Conference on Machine Learning*, pages 9983–9995. PMLR, 2020.

[44] Jinsung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In *International Conference on Machine Learning*, pages 10842–10851. PMLR, 2020.

[45] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[46] Zizhao Zhang and Tomas Pfister. Learning fast sample re-weighting without reward data. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 705–714. IEEE, 2021.

[47] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *international conference on machine learning*, pages 1–9. PMLR, 2015.

[48] Linchao Zhu, Sercan O Arik, Yi Yang, and Tomas Pfister. Learning to transfer learn: Reinforcement learning-based selection for adaptive transfer learning. In *European Conference on Computer Vision*, pages 342–358. Springer, 2020.

[49] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *European conference on computer vision*, pages 566–583. Springer, 2020.