Supplementary Material for "COMET: Learning Cardinality Constrained Mixture of Experts with Trees and Local Search"

Shibal Ibrahim
Massachusetts Institute of Technology
Cambridge, MA, USA
shibal@mit.edu

Wenyu Chen
Massachusetts Institute of Technology
Cambridge, MA, USA
wenyu@mit.edu

Hussein Hazimeh
Google Research
New York, NY, USA
hazimeh@google.com

Natalia Ponomareva
Google Research
New York, NY, USA
nponomareva@google.com

Zhe Zhao
Google DeepMind
Mountain View, CA, USA
zhezhao@google.com

Rahul Mazumder
Massachusetts Institute of Technology
Cambridge, MA, USA
rahulmaz@mit.edu

ACM Reference Format:
Shibal Ibrahim, Wenyu Chen, Hussein Hazimeh, Natalia Ponomareva, Zhe Zhao, and Rahul Mazumder. 2023. Supplementary Material for "COMET: Learning Cardinality Constrained Mixture of Experts with Trees and Local Search". In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, Long Beach, CA, USA, 4 pages. https://doi.org/10.1145/3580305.3599278

S1 ADDITIONAL DETAILS FOR SECTION 5.1

S1.1 Datasets

MovieLens. MovieLens [6] is a movie recommendation dataset containing records for $\sim 4,000$ movies and $\sim 6,000$ users. Following [16], for every user-movie pair, we construct two tasks. Task 1 is a binary classification problem for predicting whether the user will watch a particular movie. Task 2 is a regression problem to predict the user's rating (in $\{1, 2, \cdots, 5\}$) for a given movie. We use 1.6 million samples for training and 200, 000 for each of the validation and testing sets.

Jester. Jester [5] is a joke recommendation dataset containing records for $\sim 74k$ users and ~ 100 jokes. This gives a dataset of 7.4 million records. Similar to MovieLens above, for every user-joke pair, we construct two tasks. Task 1 is a binary classification problem for predicting whether the user will rate a particular joke. Task 2 is a regression problem to predict the user's rating (in $[-10, 10]$) for a given joke. We use 5.1 million samples for training and 1.1 million samples for each of the validation and testing sets.

Books. Books [19] is a book recommendation dataset containing records for $\sim 105k$ users and $\sim 340k$ books. We filter users and books with each atleast 5 records. This gives a subset of 18, 960 users and 31, 070 books. This gives a subset of 556,724 records. Similar to MovieLens above, for every user-book pair, we construct two tasks. Task 1 is a binary classification problem for predicting whether the user will read a particular book. Task 2 is a regression problem to predict the user's rating (in $\{1, 2, \cdots, 10\}$) for a given book. We use 389,706 samples for training and 83,509 for each of the validation and testing sets.

Digits. We use a mixture of MNIST [3] and SVHN [10] datasets. MNIST is a database of 70, 000 handwritten digits. SVHN is a much harder dataset of $\sim 600,000$ images obtained from house numbers in Google Street View images. We divided the dataset into training, validation and testing as follows: MNIST (#train: 50,000, #validation: 10,000, #test: 10,000) and SVHN (#train: 480,420, #validation: 75,000, #test: 75,000). We combined the corresponding splits to get the train, validation and test sets for the mixture.

MultiMNIST/MultiFashionMNIST. We consider multi-task variants of MNIST/MultiFashionMNIST [3]. The datasets are constructed in a similar fashion as given in [7, 13]: (i) uniformly sample two images from MNIST and overlay them on top of each other, and (ii) shift one digit towards the top-left corner and the other digit towards the bottom-right corner (by 4 pixels in each direction). This procedure leads to 36×36 images with some overlap between the digits. We consider two classification tasks: Task 1 is to classify the top-left item and Task 2 is to classify the bottom-right item. We use 100,000 samples for training, and 20, 000 samples for each of the validation and testing sets.

CelebA. CelebA [9] is a large-scale face attributes dataset with more than 200, 000 celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. We consider 10 of the face attributes in a multi-task learning setting. We use $\sim 160,000$ images for training, and $\sim 20,000$ for each of validation and testing.

S1.2 Architectures

MovieLens. We consider a multi-gate MoE architecture, where each task is associated with a separate gate. The MoE architecture consists of a shared bottom subnetwork comprising two embedding layers (for users and movies). The 128-dimensional embeddings from both layers are concatenated and fed into an MoE Layer of 16 experts, where each expert is a ReLU-activated dense layer with 256 units, followed by a dropout layer (with a dropout rate of 0.5). For each of the two tasks, the corresponding convex combination of the experts is fed into a task-specific subnetwork. The subnetwork is

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599278

composed of a dense layer (ReLU-activated with 256 units) followed by a single unit that generates the final output of the task.

Books/Jester. We consider a multi-gate MoE architecture, where each task is associated with a separate gate. The MoE architecture consists of a shared bottom subnetwork comprising two embedding layers (for users and books/jokes). The 64-dimensional embeddings from both layers are concatenated and fed into an MoE Layer of 9/16 experts, where each expert is a ReLU-activated dense layer with 128 units, followed by a dropout layer (with a dropout rate of 0.5). For each of the two tasks, the corresponding convex combination of the experts is fed into a task-specific subnetwork. The subnetwork is composed of a dense layer (ReLU-activated with 256 units) followed by a single unit that generates the final output of the task.

Digits. We use a single-gate MoE with 8 experts. Each of the experts is a CNN that is composed (in order) of: (i) convolutional layer 1 (kernel size = 5, number of filters = 10, ReLU-activated) followed by max pooling, (ii) convolutional layer 2 (kernel size = 5, number of filters = 20, ReLU-activated) followed by max pooling, and (iii) a ReLU-activated dense layer with 50 units. The subnetwork specific to the prediction task is composed of a stack of 3 dense layers: the first two have 50 ReLU-activated units and the third has 10 units followed by a softmax.

MultiMNIST/MultiFashionMNIST. We use a multi-gate MoE with 16/5 experts. Each of the experts is a CNN that is composed (in order) of: (i) convolutional layer 1 (kernel size = 5, #filters = 10, ReLU-activated) followed by max pooling, (ii) convolutional layer 2 (kernel size=5, #filters = 20, ReLU-activated) followed by max pooling, and (iii) a sequence of 2 ReLU-activated dense layers with 50 units each. The subnetwork specific to each of the 2 tasks is composed of a stack of 3 dense layers: the first two have 50 ReLU-activated units and the third has 10 units followed by a softmax.

CelebA. We use a multi-gate MoE with 6 experts. Each of the experts is a CNN that is composed (in order) of: (i) convolutional layer 1 (kernel size = 3, #filters = 4, ReLU-activated) followed by max pooling, (ii) convolutional layer 2 (kernel size=3, #filters = 4, ReLU-activated) followed by max pooling, (iii) convolutional layer 3 (kernel size=3, #filters = 4, ReLU-activated) followed by max pooling, and (iv) convolutional layer 4 (kernel size=3, #filters = 1, ReLU-activated) followed by max pooling, and (v) flatten layer. The subnetwork specific to each of the 2 tasks is composed of a dense layer followed by a sigmoid.

S1.3 Hyperparameters and Tuning

We performed 500 tuning trials for each gate with a random search over the hyperparameter space described below (for each dataset). For each gate, we selected Top 5% of the trials based on validation loss. We report the (average) test loss for the Top 5% trials along with the standard errors in Tables **??** and **??**.

MovieLens.
- Learning Rates: Uniform in the log range $[5\times10^{-5}, 5\times10^{-4}]$ for Adam.
- Batch-size: 512.
- Epochs: 100 with early stopping (patience=25) based on validation set.
- γ: Discrete uniform in the set $\{0.01, 0.1, 1, 5, 10\}$ for DSelect-k and *COMET*. γ is fixed to 10 for *COMET+*.
- Entropy: Discrete uniform in the set $\{0.05, 0.1, 0.5, 1, 5, 10\}$ for DSelect-k and *COMET* and *COMET+*.
- Number of epochs for permutation learning: Discrete uniform in the set $\{1, \cdots, 10\}$ for *COMET+* and *Top-k+*.
- ζ (for permutation): 10^{-4}
- n (number of experts): 16.
- k: 2 for all sparse (trainable) gates.
- For Hash-r (and *Hash-r+*), users are randomly pre-allocated to experts (similar to how words in vocabulary are pre-allocated randomly in LLMs)
- Number of tuning trials per gate: 500

Books.
- Learning Rates: Uniform in the log range $[5\times10^{-5}, 5\times10^{-4}]$ for Adam.
- Batch-size: 2048.
- Epochs: 100 with early stopping (patience=25) based on validation set.
- γ: Discrete uniform in the set $\{0.1, 0.5, 1, 5, 10\}$ for DSelect-k and *COMET*. γ is fixed to 0.5 for *COMET+*.
- Entropy: Discrete uniform in the set $\{1, 5, 10, 50, 100\}$ for DSelect-k and *COMET* and *COMET+*.
- Number of epochs for permutation learning: Discrete uniform in the set $\{1, \cdots, 10\}$ for *COMET+* and *Top-k+*.
- ζ (for permutation): 10^{-4}
- n (number of experts): 9.
- k: 4 for all sparse (trainable) gates.
- For Hash-r (and *Hash-r+*), users are randomly pre-allocated to experts (similar to how words in vocabulary are pre-allocated randomly in LLMs)
- Number of tuning trials per gate: 500

Jester.
- Learning Rates: Uniform in the log range $[5\times10^{-5}, 5\times10^{-4}]$ for Adam.
- Batch-size: 2048.
- Epochs: 100 with early stopping (patience=25) based on validation set.
- γ: Discrete uniform in the set $\{0.001, 0.02, 0.1, 1, 5, 10\}$ for DSelect-k and *COMET*. γ is fixed to 0.01 for *COMET+*.
- Entropy: Discrete uniform in the set $\{0.05, 0.1, 0.5, 1, 5, 10\}$ for DSelect-k and *COMET* and *COMET+*.
- Number of epochs for permutation learning: Discrete uniform in the set $\{1, \cdots, 10\}$ for *COMET+* and *Top-k+*.
- ζ (for permutation): 10^{-4}
- n (number of experts): 16.
- k: 2 for all sparse (trainable) gates.
- For Hash-r (and *Hash-r+*), users are randomly pre-allocated to experts (similar to how words in vocabulary are pre-allocated randomly in LLMs)
- Number of tuning trials per gate: 500

Digits.
- Learning Rates: Uniform in the log range $[1\times10^{-5}, 5\times10^{-4}]$ for Adam.
- Batch-size: 512.

- Epochs: 200 with early stopping (patience=25) based on validation set.
- γ: Discrete uniform in the set $\{0.001, 0.01, 0.1, 1\}$ for DSelect-k and *COMET*. γ is fixed to 0.001 for *COMET+*.
- Entropy: Discrete uniform in the set $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ for DSelect-k and *COMET* and *COMET+*.
- Number of epochs for permutation learning: Discrete uniform in the set $\{1, \cdots, 10\}$ for *COMET+* and *Top-k+*.
- ζ (for permutation): 10^{-4}
- n (number of experts): 8.
- k: 2 for all sparse (trainable) gates.
- Number of tuning trials per gate: 500

MultiMNIST.

- Learning Rates: Uniform in the log range $[1\times10^{-4}, 1\times10^{-3}]$ for Adam.
- Batch-size: 512.
- Epochs: 200 with early stopping (patience=25) based on validation set.
- γ: Discrete uniform in the set $\{0.001, 0.01, 0.1, 1, 5, 10\}$ for DSelect-k and *COMET*. γ is fixed to 0.01 for *COMET+*.
- Entropy: Discrete uniform in the set $\{0.0001, 0.001, 0.01, 0.1, 1, 5, 10\}$ for DSelect-k and *COMET* and *COMET+*.
- Number of epochs for permutation learning: Discrete uniform in the set $\{1, \cdots, 10\}$ for *COMET+* and *Top-k+*.
- ζ (for permutation): 10^{-4}
- n (number of experts): 16.
- k: 4 for all sparse (trainable) gates.
- Number of tuning trials per gate: 500

MultiFashionMNIST.

- Learning Rates: Uniform in the log range $[1\times10^{-4}, 1\times10^{-3}]$ for Adam.
- Batch-size: 512.
- Epochs: 200 with early stopping (patience=25) based on validation set.
- γ: Discrete uniform in the set $\{0.001, 0.01, 0.1, 1, 5, \}$ for DSelect-k and *COMET*. γ is fixed to 0.01 for *COMET+*.
- Entropy: Discrete uniform in the set $\{0.001, 0.01, 0.1, 1, 5\}$ for DSelect-k and *COMET* and *COMET+*.
- Number of epochs for permutation learning: Discrete uniform in the set $\{1, \cdots, 10\}$ for *COMET+* and *Top-k+*.
- ζ (for permutation): 10^{-4}
- n (number of experts): 6.
- k: 2 for all sparse (trainable) gates.
- Number of tuning trials per gate: 500

CelebA.

- Learning Rates: Uniform in the log range $[1\times10^{-4}, 1\times10^{-3}]$ for Adam.
- Batch-size: 512.
- Epochs: 100 with early stopping (patience=25) based on validation set.
- γ: Discrete uniform in the set $\{0.001, 0.01, 0.1, 1, 5\}$ for DSelect-k and *COMET*. γ is fixed to 5 for *COMET+*.
- Entropy: Discrete uniform in the set $\{0.001, 0.01, 0.1, 1, 5\}$ for DSelect-k and *COMET* and *COMET+*.
- Number of epochs for permutation learning: Discrete uniform in the set $\{1, \cdots, 10\}$ for *COMET+* and *Top-k+*.
- Entropy for permutation: 10^{-4}
- k: 2 for all sparse gates.
- Number of tuning trials per gate: 100

S2 ADDITIONAL DETAILS FOR SECTION 5.2

S2.1 Datasets

GLUE. General Language Understanding Evaluation (GLUE) benchmark [15], is a collection of natural language understanding tasks. Following previous works on model distillation, we consider SST-2 [14], CoLA [17], MRPC [4], STSB [1], QQP, and MNLI [18] and exclude STS-B [1] and WNLI [8] in the experiments. The datasets are briefly summarized below:

- SST-2 [14] is a binary single-sentence classification task that classifies movie reviews to positive or negative;
- CoLA [17] is a linguistic acceptability task;
- MRPC [4] is a paraphrase detection task;
- QQP is a duplication detection task;
- MNLI [18], QNLI [12], and RTE [2] are natural language inference tasks.

Dataset details are summarized in Table S1.

Table S1: Summary of GLUE benchmark.

Corpus	Task	#Train	#Dev	#Test	#Label	Metrics
Single-Sentence Classification (GLUE)						
CoLA	Acceptability	8.5k	1k	1k	2	Matthews correlation
SST-2	Sentiment	67k	872	1.8k	2	Accuracy
Pairwise Text Classification (GLUE)						
MNLI	NLI	393k	20k	20k	3	Accuracy
RTE	NLI	2.5k	276	3k	2	Accuracy
QQP	Paraphrase	364k	40k	391k	2	Accuracy/F1
MRPC	Paraphrase	3.7k	408	1.7k	2	Accuracy/F1
QNLI	QA/NLI	108k	5.7k	5.7k	2	Accuracy

SQuAD. We evaluate our sparse routing approaches on question answering dataset: SQuAD v2.0 [11]. This task is treated as a sequence labeling problem, where we predict the probability of each token being the start and end of the answer span. Statistics of the question answering dataset (SQuAD v2.0) are summarized in Table S2.

Table S2: Summary of SQuAD benchmark.

Corpus	Task	#Train	#Dev	Metrics
SQuAD v2.0	Question Answering	130k	11.9k	F1/EM

S2.2 Tuning Procedure for MoEBERT and *COMET-BERT*

Following [20], we followed the 3-step process as outlined in the MoEBERT codebase[1]:

[1] https://github.com/SimiaoZuo/MoEBERT

- We finetuned BERT on each downstream task for a set of 50 random hyperparameter trials over the following set:
  - Learning Rate: Discrete uniform over the set $\{1\times10^{-5}, 2\times10^{-5}, 3\times10^{-5}, 5\times10^{-5}\}$
  - Batch size: Discrete uniform over the set $\{8, 16, 32, 64\}$
  - Weight Decay: Discrete uniform over the set $\{0, 0.01, 0.1\}$
  - Epochs: 10

  Note that this step matched the performance numbers reported for BERT-base in Table 1 of [20]. We used the best model (for each dataset) for the remaining steps below.
- Compute importance weights in FFN layers to construct an MoEBERT/COMET-BERT model, where FFN layers are replaced with MoE layers with the weight assignment strategy in [20].
- Distill BERT into MoEBERT or *COMET-BERT* on the downstream task with a layer-wise discrepancy loss. For MoEBERT, we used the optimal hyperparameters reported (based on $\sim$ 1000 trials per dataset) in Table 7 of Supplement in [20]. For *COMET-BERT*, we performed 100 tuning trials via random search with each *COMET* and *COMET+* and picked the best results based on development datasets. The hyperparameters were randomly selected from the following sets:
  - Learning Rate: Discrete uniform over the set $\{1\times10^{-5}, 2\times10^{-5}, 3\times10^{-5}, 5\times10^{-5}\}$
  - Batch size: Discrete uniform over the set $\{8, 16, 32, 64\}$
  - Weight Decay: Discrete uniform over the set $\{0, 0.01, 0.1\}$
  - Distillation Regularization ($\lambda_{distill}$ in [20]): Discrete uniform over the set $\{1, 2, 3, 5\}$.
  - γ (for smooth-step for *COMET*): Discrete uniform over the set $\{0.01, 0.1, 1.0\}$.
  - λ (for entropy regularization for *COMET*): Discrete uniform over the set $\{0.05, 0.1, 0.5, 1, 5, 10\}$.
  - Epochs: 50 for small datasets (CoLA, RTE, MRPC) and 25 for large datasets (SST-2, MNLI, QQP, QNLI, SQuADv2.0). Best model was recovered on development set on best checkpoint.

REFERENCES

[1] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, 1–14. https://doi.org/10.18653/v1/S17-2001

[2] Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 177–190.

[3] Li Deng. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine* 29, 6 (2012), 141–142.

[4] William B. Dolan and Chris Brockett. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. https://aclanthology.org/I05-5002

[5] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. 2001. Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information Retrieval* 4, 2 (2001), 133–151.

[6] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (dec 2015), 19 pages. https://doi.org/10.1145/2827872

[7] Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. 2021. DSelect-k: Differentiable Selection in the Mixture of Experts with Applications to Multi-Task Learning. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.). https://openreview.net/forum?id=tKlYQJLYN8v

[8] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning* (Rome, Italy) *(KR'12)*. AAAI Press, 552–561.

[9] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[10] Yuval Netzer, Tao Wang, Adam Coates, A. Bissacco, Bo Wu, and A. Ng. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning.

[11] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, 784–789. https://doi.org/10.18653/v1/P18-2124

[12] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 2383–2392. https://doi.org/10.18653/v1/D16-1264

[13] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing between Capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 3859–3869.

[14] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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*. Association for Computational Linguistics, Seattle, Washington, USA, 1631–1642. https://aclanthology.org/D13-1170

[15] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJ4km2R5t7

[16] Yuyan Wang, Zhe Zhao, Bo Dai, Christopher Fifty, Dong Lin, Lichan Hong, and Ed H. Chi. 2020. Small Towers Make Big Differences. *ArXiv* abs/2008.05808 (2020).

[17] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics* 7 (09 2019), 625–641. https://doi.org/10.1162/tacl_a_00290 arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00290/1923083/tacl_a_00290.pdf

[18] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. https://doi.org/10.18653/v1/N18-1101

[19] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving Recommendation Lists through Topic Diversification. In *Proceedings of the 14th International Conference on World Wide Web* (Chiba, Japan) *(WWW '05)*. Association for Computing Machinery, New York, NY, USA, 22–32. https://doi.org/10.1145/1060745.1060754

[20] Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2022. MoEBERT: from BERT to Mixture-of-Experts via Importance-Guided Adaptation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 1610–1623. https://doi.org/10.18653/v1/2022.naacl-main.116