MAster of PuPpets: Model-Agnostic Meta-Learning via Pre-trained Parameters for Natural Language Generation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Pre-trained Transformer-based language models have been an enormous success in generating realistic natural language. However, how to adapt these models to specific domains effectively remains unsolved. On the other hand, Model-Agnostic Meta-Learning (MAML) has been an influential framework for few-shot learning, while how to determine the initial parameters of MAML is still not well-researched. In this paper, we fuse the information from the pre-training stage with meta-learning to learn how to adapt a pre-trained generative model to a new domain. In particular, we find that applying the pre-trained information as the initial state of meta-learning helps the model adapt to new tasks efficiently and is competitive with the state-of-the-art results over evaluation metrics on the Persona dataset. Besides, in few-shot experiments, we show that the proposed model converges significantly faster than naive transfer learning baselines.

1 Introduction

Model-Agnostic Meta-Learning (MAML) [5] has been a widely applied framework for few-shot learning in many domains, such as computer vision (CV), natural language processing (NLP) and speech recognition (SR) [8, 25, 29, 31]. The goal of MAML is to learn a set of initial parameters that can quickly adapt to a new downstream task. Despite the effectiveness of MAML on few-shot learning, there is an unsolved problem in MAML training. Since MAML is a gradient-based optimization method, it requires a set of initial parameters too. We call this set of parameters as meta-initial parameters in the paper. It raises a question about how to determine the meta-initial parameters at the beginning of the MAML procedure.

Transfer learning is another method frequently adopted for few-shot learning [19, 22, 28]. Among all of the pre-training model architecture used for transfer learning, Transformer [23] is the most widely applied and researched in the field of NLP. There are plenty of works pre-train the Transformer-based model and achieve huge successes on NLP tasks such as detecting semantic similarity, language modeling, natural language inference, and machine translation [4, 10, 12, 18, 24]. However, transferring these models with a large number of parameters usually requires a lot of fine-tuning data [15, 17].

To tackle these challenges, we consider initializing the meta-initial parameters by the parameters of a pre-trained model instead of randomly initialized parameters. Fig 1 shows a high-level intuition of the difference between these two initializing strategies. By adopting the pre-trained parameters, we remarkably reduce the possible states of meta-initial parameters into a subset that can be more similar to downstream tasks.

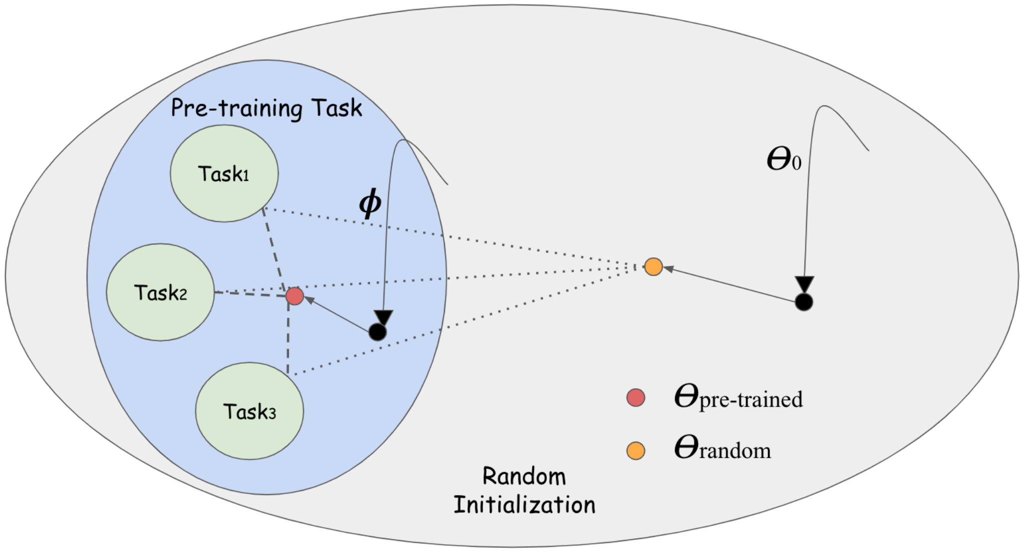


Figure 1: The difference between meta-learning from a random initialized point θ_0 and a point with pre-training information ϕ. The solid line represents the optimization path of meta-learning procedure and the dashed line represents the fine-tuning path. Because the pre-training task is a super set containing downstream tasks, the meta-learning procedure can find a better start point for fine-tuning.

In this paper, we propose a meta-learning framework, called MAMLviaPP, composed of MAML and pre-training information for natural language generation. In spite of the simplicity of MAMLviaPP, the improvement of the performance is significant. We summarize the contributions of this paper as:

- We investigate the possibilities of utilizing MAML on the pre-trained Transformer-based model. To the best of our knowledge, this is the first work taking advantage of MAML on the pre-trained model with this scale of parameters.
- We propose a method to initialize the starting point of MAML, which is a problem rarely surveyed. The experiments show promising results of this simple yet effective strategy.
- By combining meta-learning with pre-train/transfer learning, the relationship between these two domains is slightly clarified that they are not entirely disjointed.

2 Related Work

Meta-Learning The goal of meta-learning is to learn the learning algorithm itself [1, 2, 21]. Among these meta-learning algorithms, MAML [5] is widely used for few-shot learning due to the ability of fast adapting to a new domain. Several MAML-based models are proposed to solve few-shot image recognition [31], text classification [29], speech recognition [8] and neural architecture searching [11]. However, most of these works focus on utilizing meta-learning on applications, while the meta-initial parameters are all random initialized.

The most related meta-learning work to our paper is Meta-transfer learning (MTL) [20]. MTL meta-trains the model on multiple tasks and then trains the scaling and shifting functions of DNN weights for a specific domain to achieve the transfer learning in downstream tasks. In contrast to MTL, which requires a manually pre-training procedure, our proposed method MAMLviaPP generalizes well to all kinds of neural network architectures and pre-trained models.

Transformers Transformers [23] have made enormous impact in many fields of CV and NLP such as object detection [3], detecting semantic similarity, natural language inference and machine translation [4, 10, 12, 18, 24]. Directly fine-tuning the pre-trained Transformer model on a new task is a classical approach to transfer the learned information. However, it demands a lot of fine-tuning data to transfer the model to a new domain effectively [15, 17]. To deal with this difficulty, we take advantage of quickly adapting achieved by meta-learning with pre-trained Transformers.

Algorithm 1 In-place Model-Agnostic Meta-Learning via Pre-trained Parameters

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
Require: ϕ: pre-trained model parameters
Initialize $\theta \leftarrow \phi$
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Evaluate $\nabla_\theta \mathcal{L}_i(f_\theta)$ with respect to K examples
    Compute adapted parameters with gradient descent: $\hat{\theta}_i = \theta - \alpha \nabla_\theta \mathcal{L}_i(f_\theta)$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_i(f_{\hat{\theta}_i})$
end while

3 Proposed Method

The goal of the proposed algorithm is to train a learner quickly adapts to a new domain given a group of pre-trained parameters. To accomplish this, the model enters the meta-training phase with the pre-trained parameters. In Section 3.1, we describe the problem setups and the proposed algorithm, and in Section 3.2 we investigate the feasibility of using a generative pre-trained transformer as the base learner in the proposed framework.

3.1 Model-Agnostic Meta-Learning via Pre-trained Parameters

The general form of MAML is defined as follows. Consider a set of tasks $\mathcal{T} = \{\mathcal{T}_{train}, \mathcal{T}_{test}\}$, $\mathcal{T}_{train} = \{\mathcal{T}_{train_1}, \mathcal{T}_{train_2}, \mathcal{T}_{train_3}, \ldots, \mathcal{T}_{train_N}\}$ and $\mathcal{T}_{test} = \{\mathcal{T}_{test_1}, \mathcal{T}_{test_2}, \mathcal{T}_{test_3}, \ldots, \mathcal{T}_{test_M}\}$, learner f, meta-learned parameters θ, loss function of $\mathcal{T}_m$, summation of task losses $\mathcal{L}$ and parameters of model after fine-tuning $\hat{\theta}$. MAML framework aims to minimize the objective function:

$$\mathcal{L}(f_\theta) = \sum_{m=1}^{M} \mathcal{L}_m(f_{\hat{\theta}_m}) \tag{1}$$

In contrast to original work, which randomly initializes θ at the beginning of the meta-training phase, we propose two methods: one is to adopt pre-trained parameters ϕ as the initialization, and the other is to make the initialized θ as close to ϕ as possible. To be specific, in the case of initializing θ with ϕ, the proposed algorithm iteratively meta-trains θ with ϕ as the initial state. We call this method In-place MAMLviaPP. On the other hand, in the second method, named Extra-place MAMLviaPP, the model is integrated with additional parameters Φ while the model's outputs over all possible inputs remain the same. Formally, the limitation is defined as follows:

$$f_\phi(x \sim \mathcal{T}_i) \approx f_{[\phi,\Phi]}(x \sim \mathcal{T}_i) \; \forall \mathcal{T}_i \in \mathcal{T} \tag{2}$$

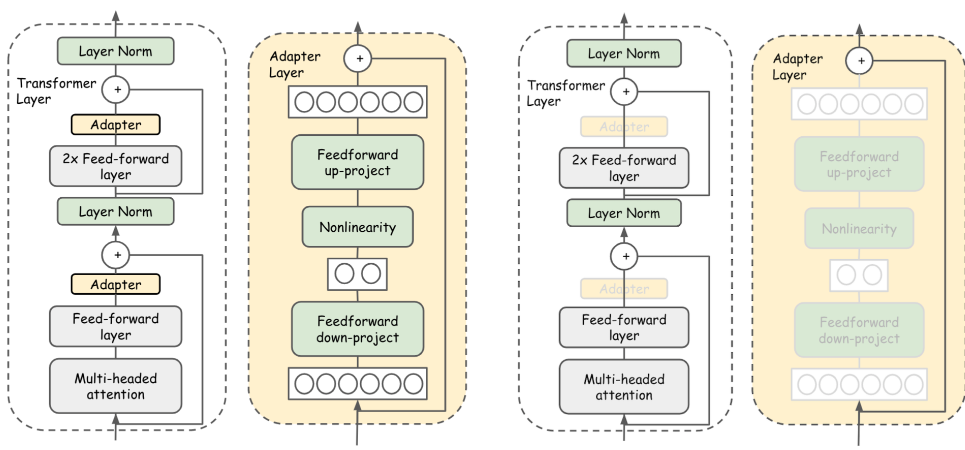
The limitation in Eq (2) ensures that the initialized model $f_{[\phi,\Phi]}$ behaves the same as the pre-trained model f_ϕ on the space of $\mathcal{T}$. Therefore, the initial state of meta-training retains the information from pre-training. Furthermore, in the meta-training phase of Extra-place MAMLviaPP, the pre-trained parameters ϕ is fixed and only the extra-integrated parameters Φ is trained with gradient-decent. This mechanism ensures the model preserving the information learned in the pre-training stage and enriches the model capacities for domain adaption. The training details of algorithm is shown in Algorithm 1 and Algorithm 2 respectively.

3.2 Generative Pre-trained Transformer as the Base Learner

To demonstrate the use case of the proposed method, we investigate the application of generative pre-trained transformer in this section. We use GPT-2 [18] as the base learner in this paper, while the choices of the base model are not limited to GPT-2.

Algorithm 2 Extra-place Model-Agnostic Meta-Learning via Pre-trained Parameters

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
Require: ϕ: pre-trained model parameters
Initialize $\theta \leftarrow [\phi, \Phi]$
Fix ϕ in the training procedure
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Evaluate $\nabla_\Phi \mathcal{L}_i(f_{[\phi,\Phi]})$ with respect to K examples
    Compute adapted parameters with gradient descent: $\hat{\theta}_i = [\phi, \hat{\Phi}_i] = \theta - \alpha \nabla_\Phi \mathcal{L}_i(f_{[\phi,\Phi]})$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\Phi \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_i(f_{\hat{\theta}_i})$
end while



(a) GPT-2 architecture with adapters. (b) The approximately zero initialization of adapters.

Figure 2: The placement of adapters and the initialization procedure. As shown in 2b, when the adapters are zero initialized, the residual connection forwards the inputs. Therefore, since the adapters are skipped, Eq 2 is fulfilled.

Adopting GPT-2 in In-place MAMLviaPP is straightforward. In-place MAMLviaPP algorithm directly utilizes pre-trained GPT-2 parameters ϕ as the initialization of MAML. The rest of the algorithm is the same as the regular MAML training.

In the case of Extra-place MAMLviaPP, it requires specialized modification to fit the GPT-2 model into the framework. According to Eq (2), the parameter-integrated model $f_{[\phi,\Phi]}$ must behave the same as GPT-2 model. In this paper, we choose to make use of adapters proposed by [7] as the trainable parameters Φ in Extra-place MAMLviaPP. By initializing the weights of adapters $\Phi \approx 0$, so are the outputs of adapters. Therefore, the integrated model $f_{[\phi,\Phi]}$ follows the limitation. The model architecture and the procedure of initializing adapters are shown in Figure 2.

4 Experiments

The dataset used in the experiments is **Persona-chat** [30]. We follow the experimental settings of PAML [14], which views persona groups as tasks in the MAML scenario. Meta-tasks set is created by matching the dialogues by respective persona description and splitting them into train, validation, and test by the same persona split in [30]. We list the details of the experimental setups in Appendix B.

Table 1: Results of evaluation metrics.

Model	Perplexity ↓	Hits@1(%) ↑	F1(%) ↑
REINIT GPT-2	72.36	7.8	8.21
Transfer Pre-trained GPT-2	27.35	10.9	11.32
REINIT GPT-2 + MAML	57.79	10.1	10.41
In-place MAMLviaPP	14.10	13.2	15.93
Extra-place MAMLviaPP without Fixing ϕ	23.79	11.7	12.55
Extra-place MAMLviaPP	**13.21**	16.4	19.38
Transfertransfo [27]	17.51	**82.2**	19.09
P^2 BOT [13]	15.12	81.9	**19.77**
P^2 BOT without Next Utterance Prediction	N/A	17.6	18.11

4.1 Evaluation Metrics

Following the official metrics used by [30], we evaluate the proposed model with three metrics: **Hits@1**, **Perplexity(ppl)** and **F1 score**. The detailed descriptions are listed as follows.

- **Hits@1**: The metric consists of fetching 19 distracting responses from other dialogues. The model is requested to select the best response among $19 + 1$ candidates. The score is the percentage of the model ranking the correct response as the top-1 selection.
- **Perplexity(ppl)**: Perplexity is the normalized inverse probability of the correct sequence. Since all the models are the probability model, we can evaluate the perplexity of generators conditioned on the real data.
- **F1 score**: F1 score is the harmonic mean of word-level precision and recall considering the generations and the real dialogues.

4.2 Ablation Study

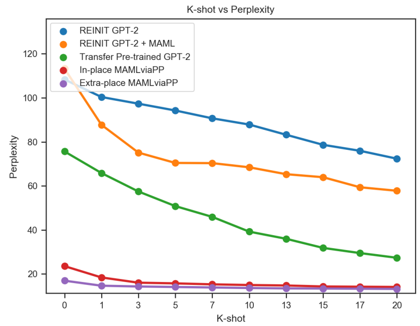
There are two types of training methods: normal training and meta training. The normal training method trains the model on the meta-training sets by the same objective function in the meta-testing sets. On the other hand, the meta training method trains the model by inner-loop and outer-loop meta-training procedures on meta-training sets. On the testing stage, both methods fine-tune and then evaluate the trained model on the meta-testing sets.

Normal training: **REINIT GPT-2** a random initialized model with the same model architecture as GPT-2; **Transfer Pre-trained GPT-2** a GPT-2 model loaded with pre-trained parameters.

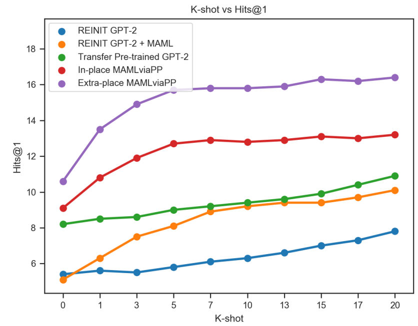
Meta training: **REINIT GPT-2 + MAML** REINIT GPT-2 trained by MAML; **In-place MAMLviaPP** a pre-trained GPT-2 trained by Alg 1; **Extra-place MAMLviaPP** a pre-trained GPT-2 with additional adapters as shown in Fig 2 trained by Alg 2; **Extra-place MAMLviaPP without Fixing ϕ** the same model as Extra-place MAMLviaPP without fixing ϕ.

4.3 Results

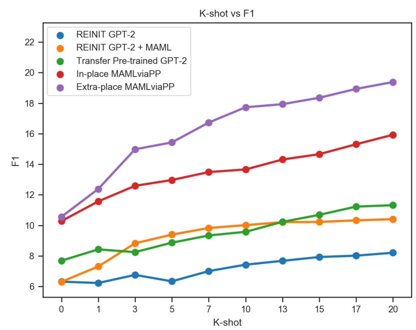
Table 1 compares the experimental results of different settings and previous works. Generally, the pre-trained parameters support the model to be a better generator. Comparing the results of REINIT GPT-2 + MAML and In-place MAMLviaPP, we find that MAML with the pre-trained parameters as the initial state significantly outperforms randomly initialized MAML in all metrics, which indicates the effectiveness of combing meta-learning and pre-trained models. Besides, fixing pre-trained parameters ϕ in the meta-training procedure preserves the information in ϕ and makes the model adapt to a new domain more effectively as shown in the results comparing Extra-place MAMLviaPP with Extra-place MAMLviaPP without Fixing ϕ. To compare our results with the state-of-the-art, we list Transfertransfor [27] and P^2 BOT [13] in the table. Our best method Extra-place MAMLviaPP significantly outperforms both previous works on perplexity and is competitive on the F1 score. In [13], the authors show that models trained with the Next Utterance Prediction (NUP) task are significantly improved on the Hits@1 metric, while our models are trained only with the language modeling task. As a result, we compare our method with P^2 BOT without NUP on Hits@1. We



(a) K-shot vs Perplexity



(b) K-shot vs Hits@1



(c) K-shot vs F1 Score

Figure 3: The results of K-shot experiments for different settings. The proposed two methods both adapt to a new domain immediately and consistently outperform baseline models.

show that our best model is close to the performance of the state-of-the-art model without NUP on this metric.

To analyze the ability to adapt to a new task, we evaluate our trained models with a k-shot experiment. The K in k-shot represents the number of dialogues available in each task in the fine-tuning stage. Results are shown in Fig 3. As shown in the figure, the proposed two methods not only adapt to a new domain quickly but also outperform the transfer learning baselines on all metrics, which proves the effects of merging meta-learning with pre-trained parameters. Besides, we also show generated samples from the proposed models and baseline models in Appendix C to better understand the behavior of the generators.

5 Conclusion

In this paper, we propose MAMLviaPP, a simple yet effective method for merging MAML with a pre-trained model. The benchmark experiments demonstrate that MAMLviaPP improves the quality of generated sentences and enhances the ability of fast adapting to a new domain as measured by various metrics and k-shot experimental settings. In terms of implementation difficulties, the proposed method is straightforward and can be generalized to whatever the pre-trained model is.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

[2] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, volume 2. Univ. of Texas, 1992.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[4] 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.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[6] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[7] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv:1902.00751*, 2019.

[8] J. Hsu, Y. Chen, and H. Lee. Meta learning for end-to-end low-resource speech recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7844–7848, 2020.

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

[10] 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.

[11] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[12] 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.

[13] Qian Liu, Yihong Chen, Bei Chen, Jian-Guang Lou, Zixuan Chen, Bin Zhou, and Dongmei Zhang. You impress me: Dialogue generation via mutual persona perception. *arXiv preprint arXiv:2004.05388*, 2020.

[14] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, 2019.

[15] Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. A bert-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*, pages 928–940. Springer, 2019.

[16] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[17] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*, 2019.

[18] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[19] Mohammad Rostami, Soheil Kolouri, Eric Eaton, and Kyungnam Kim. Deep transfer learning for few-shot sar image classification. *Remote Sensing*, 11(11):1374, 2019.

[20] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2019.

[21] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[22] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3081–3088. IEEE, 2010.

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

[24] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[25] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286, 2018.

[26] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

[27] Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*, 2019.

[28] Xiaodong Yu and Yiannis Aloimonos. Attribute-based transfer learning for object categorization with zero/one training example. In *European conference on computer vision*, pages 127–140. Springer, 2010.

[29] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. Diverse few-shot text classification with multiple metrics. *arXiv preprint arXiv:1805.07513*, 2018.

[30] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*, 2018.

[31] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.