Multi-Task Zero-shot modeling with test Domain Shift: an exploration of sampling and fine-tuning techniques on DistilGPT-2 and BIG-bench

Stanford CS224N Custom Project

Lara Malinov, malinov@stanford.edu

Abstract

The research conducts an investigation of sampling and fine-tuning approaches to multi-task train DistilGPT-2 and evaluate the models on unseen domain tasks. BIG-bench tasks are separated into training tasks and evaluation tasks in order to have no overlap in associated keywords. The results show that, while different sampling and fine-tuning techniques may prove useful for out-of-sample training tasks observations, they may not necessarily adapt well on unseen domain tasks. In the end, the model trained using all of the training observations and where only the last linear layer was fine-tuned showed to perform best on tasks in new domains. This result was achieved at a very specific moment during training after the model had trained for a couple of epochs and before the model specialized further on the training tasks. At last, the results show that it is possible for a model to train well in the domains of mathematics, logical reasoning, contextual question-answering and numerical response, and adapt to the domains of emotional intelligence and emotional understanding.

Introduction

Language is humans way to express ideas and communicate with each other. There are ample ways to express one idea, spoken and written words bridge the gap between signifiers, "the conceptual material form", and the signified, "the conceptual ideal form" following Louis Hjelmslev. All in one, our constructed languages are a means to an end, and while there are hundreds of ways to phonetically, pictographically and manually express a concept, the common denominator is that language is a construct to externalise our own rationale and logic, and share it with each other.

It is thus no surprise that in the era of intelligent machines, we try to explore how we can instil rational thinking into models by the means of our own written expression of thought. The last decade has seen a phenomenal improvement in the language capabilities of models. These have been due to advancements in computing power, the availability of data, and changes in modeling architectures. In 2019, OpenAI's research team published how their 1.5 billion parameters Generative Pre-trained Transformer 2 (GPT-2 [1]) outperformed predecessors. The year 2022 has witnessed the success story of ChatGPT with its 175 billion parameters, which has taken the world by storm due to its ability to respond to text input prompts and return a concatenation of the information available on the web. The newly released GPT-4 has close to 100 trillion parameters and its preliminary version outperformed ChatGPT by 26% on the Multistate Bar Examination (MBE), "beating humans in five of seven subject areas" [2].

While these achievements are undeniably impressive, by comparison to human capabilities, these models are still largely inefficient. Humans are estimated to have between 80 and 100 billion neurons with around 100 trillion synapses, but scientists have debunked that we only use 10% of our brains. In fact "every part of our brain is integral to our daily life", but some brain images support that 10% of our brain are particularly more active than other areas depending on the task [3]. Nonetheless, human brains use around 20% of the body's energy. Meanwhile, ChatGPT is estimated to emit around 25 tCO2eq a day, around twice the yearly carbon footprint of the average American [4]. Thus, both the additional number of parameters and the high energy consumption undermine the achievements of Large Language Models (LLM) in comparison to human capabilities.

Ultimately, future research should consider the ecologically impact that their machine learning products have, especially when used at scale. To that end, more efficient models need to be created by maximizing the learning procedure per observation employed and the metric improvement rate for each additional parameter. Therefore, this research focuses on applying sampling and fine-tuning techniques to multi-task train DistilGPT-2, a relatively small language model with 82 million parameters, on a subset of the Beyond the Imitation Game Benchmark (BIG-bench [5]). In addition, the models are also evaluated on tasks that have no overlap with the training tasks, to investigate how learning some skills transfers to different ones, similarly to how humans approach new problems.

Related Work

This research is primarily inspired by the work of the researchers behind Meta In-Context Learning (MetaICL [6]) which is based on GPT-2 Large[1]. The authors use 142 different datasets to train models and evaluate them on previously unseen data from hold-out datasets. Thus, "Meta" refers to the fact that the model needs to learn to differentiate between types of datasets using the context contained in the in-context learning examples. Specifically, the model receives k examples (x_i, y_i) with $i = 1...k$ as well as x_{k+1} as input for which it needs to predict y_{k+1}. In one of their experiment setting, High Resource to Low Resource (HR→LR), they train MetaICL, and other baselines on all datasets with more than 10,000 observations and evaluate on held-out smaller target datasets, among which they are some that have no domain overlap with the training datasets. In their results, they show that MetaICL outperforms the other baselines regarding average accuracy for all target tasks. However, MetaICL performs similarly to the Multi-Task 0-shot baseline on target datasets in unseen domain. Here, the baseline is equivalent to MetaICL with k=0. Thus, adding in-context observations does not make a difference on datasets on which the model was not trained on. Nonetheless, Multi-Task 0-shot does reach higher accuracy that the 0-shot baseline (GPT-2 Large without any further training), implying that the model was able to transfer its learning to target datasets in unseen domains. As such, this research will further investigate how to train a Multi-Task 0-shot model, such that it can also perform relatively well on datasets with different domains.

Furthermore, the authors also experimented with the different model sizes available for GPT-2 (Small, Medium and Large). They show that the results are inconclusive on whether a larger parameterized model leads to better performance. In fact, it depends on the experimental setup. On most occasions, GPT-2 Small has a lower performance than GPT-2 Large but a better performance than GPT-2 Medium. Therefore, this research will experiment with DistilGPT-2, the smallest of the GPT-2 models, to further encourage the improvement of smaller models. Also, the research of [7] does not suggest the need to to train a model from scratch, as using and fine-tuning large language models leads to a marginal improvement in the performance on downstream tasks. Next, research has shown that much can be achieved by efficiently engineering the learning environment of the model, on which this research focuses. For example, in the case of multi-task learning, one can consider upsampling and downsampling tasks to encourage the learning of all tasks, similar to how one would do for unbalanced training sets in classification problems. Furthermore, surgical fine-tuning, meaning fine-tuning a subset of layers, can prove useful to adapt to shifts in data distribution [8]. Here the authors show how surgical fine-tuning improves the accuracy by 2 to 4% compared to full model fine-tuning on three different scenarios of distribution shift. Thus, these ideas and techniques will be applied to a multi-task zero-shot framework with domain shift at test time to investigate how to best transfer learning from the training tasks.

Approach

The experiments are conducted using DistilGPT-2 using the HuggingFace library [9] with Pytorch, which is a condensed version of OpenAI's GPT-2 [1] following a similar procedure as for DistilBERT [10]. This consists in training a student model to imitate the teacher (GPT-2) and minimize the error against the true label as well as the error against the teacher model (see Figure 1). The final model consists of 6 blocks compared to 24 in GPT-2 and achieves a perplexity score of 21.1 after fine-tuning compared to 16.3 for GPT-2 on the WikiText-103 benchmark. Perplexity refers to how well a distribution is approximated on the training text. Following, all training experiments are trained on the same tasks for 20 epochs in batches of 8, 16 or 32, depending on how many parameters are trained on Google Colab 15GB GPU. The optimization algorithm is AdamW with decoupled weight decay with arbitrary learning equal to 0.005 and other default parameters. Hyperparameters tuning is left for future research and here the validation dataset is primarily used to select the epoch at which the model generalizes best to out-of-domain tasks, as well as in-domain tasks (further explained below).

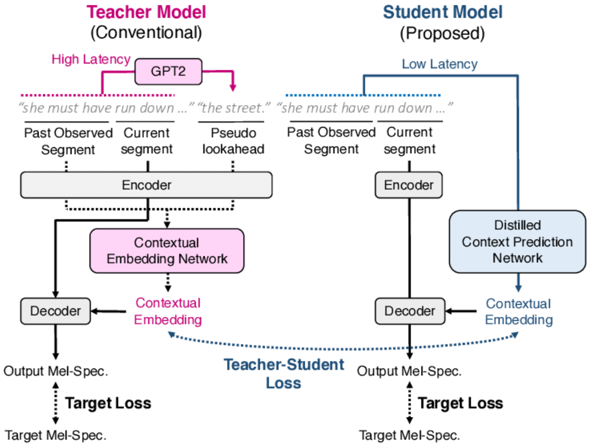


Figure 1: DistilGPT-2 training procedure
Source: https://iq.opengenus.org/distilled-gpt2/

Given that multiple tasks are used to train the models, different sampling procedures are used to see how well a model learns all tasks equivalently instead of the ones with the most observations. As such, in a first experimental setting, no specific sampling is conducted. In the second setup, all training tasks have equal weight in training by down-sampling and up-sampling individual tasks to each contain 500 observations. In the third setting, observations are sampled to have relatively equal domain representation. For example, one dataset 'causal judgment' falls in the domain of 'social reasoning', 'common sense' and 'reading comprehension' as defined per BIG-bench [5]. Thus, all domains are sampled to have relatively equal weights between 10 and 15% during training to see if a model can learn to be equally good in all disciplines.

Next, the following fine-tuning techniques are used to train models for all different sampling experiments. Inspired by the surgical fine-tuning research [8], different blocks or layers are trained as well as one setup where blocks are trained one after the other in a cascade fashion and one setting where all the parameters are trained for comparison:

- Linear Layer
- First Block (0) and Linear Layer
- Middle Block (2) and Linear Layer: the middle block 2 was arbitrarily decided
- Last Block (5) and Linear Layer
- Cascade: the first block is trained for 5 epochs, while the following blocks are trained for 3 epochs. The linear layer is trained during all epochs.
- Reversed Cascade: only the linear layer is trained for 5 epochs, the following blocks 5, 4 and 3 are trained for 3 epochs, and the first two blocks 1 and 0 are trained for 2 epochs. The linear layer is trained during all epochs.
- All Blocks and Linear Layer

The cascade technique follows the idea to fine-tune the model for upstream generalization across tasks to downstream specialized text generation. In reversed cascade, the procedure can use the earlier pre-trained weights longer before making changes to them in order to adapt to this specific multi-task setting at the downstream level.

Finally, the models are validated and tested using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE [11])-LSum metric. The metric is a variant of the ROUGE score which computes the recall for overlapping word sequences between prediction and human references, but it also penalizes for too short or too long sequences.

Experiments

Data and Experimental Setup

The Beyond the Imitation Game benchmark (BIG-bench [5]) tasks were selected in this experimental research as it falls within the scope and the motivation of this research to investigate how small models can get better at complex tasks. All of the tasks were initially run on GPT-2[1] in order to filter and select a subset of tasks which had ROUGE-LSum scores larger than zero.

Now, each of the BIG-bench tasks have associated keywords and since the aim of the research was to see how a model would perform on tasks it hadn't been trained on, the tasks were separated in order to have no overlapping keywords, or domains, between training and evaluation. Here, the MetaICL [6] research also inspired the research two separate the tasks in a way that imitates their High Resource to Low Resource experimental setup, thus training on the tasks with the highest numbers of observations and evaluate the models on tasks with the lowest amount of observations.

To that end, the Apriori Frequent Itemset [12] algorithm was applied on the set of keywords of each task to select the most frequent keywords and keywords combination. Thus, the tasks with keywords: common sense, mathematics, numerical response, social reasoning, reading comprehension, contextual question-answering, logical reasoning and free response; were selected for training. The tasks which had no overlap with training task keywords were selected for evaluation. These were characterized by the keywords: analogical reasoning, emotional understanding, morphology, non-English, medicine, emotional intelligence, dialogue systems and intent recognition. Most of the evaluation tasks are difficult and are not expected to do well at testing, such as non-English and medicine, but are included nonetheless, for the sake of the research.

Overall, there were around 10,000 observations for training available and 4,000 observation for evaluation. In addition, 707 observations from the training observations were held out for validation and testing in order to get an understanding of what the model had learned during training. Next, a validation data set was constructed at random using 25% of the held-out training observations and 25% of the evaluation task observations amounting to around 1,000 observations in total, and the remaining 75% of the held-out training tasks and evaluation tasks were used for testing, amounting to 3,000 observations (see Appendix Table 1). Thus, 80% of both validation and test data sets are evaluation tasks observations and have the highest weight since out-of-domain evaluation is the focus of the research. In hindsight, the held-out training observations could have been only used during testing. However, at the time of the research, it felt necessary to give the models some credit for what they had learned on the training tasks and they were therefore included in the validation sample.

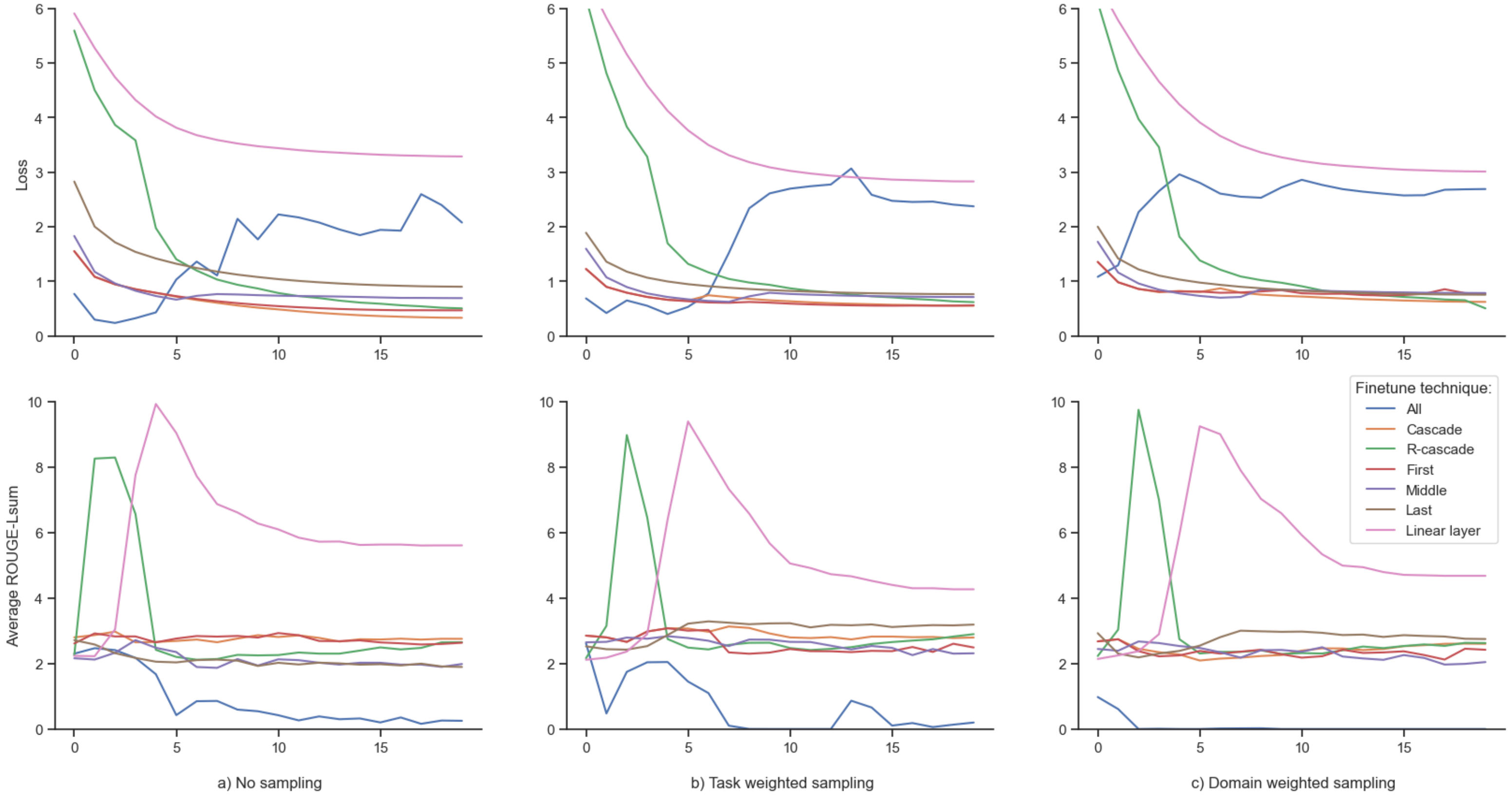
Results

All the fine-tuning techniques were applied with the three training sampling approaches for 20 epochs (see Figure 2). The results show that the loss curves decrease for all training approach, except in the case where all parameters of DistilGPT 2 were trained which starts to diverge at around 5 epochs. However, it reaches the overall lowest loss for the task weighted training sample and the case where no sampling was applied. All the models where only one transformers block was trained at a time reach similar levels towards the end with the difference being narrower for the task weighted and domain weighted training samples. Of them the cascade and reverse cascade techniques have the lowest loss at the end of training, the latter only for the domain weighted sample. The loss curve of the model where only the linear layer was trained is the highest in all training scenarios.

However, even though it is the highest, it does reach the best validation ROUGE-LSum scores on the validation dataset and peaks at around 5 epochs in all cases. Similarly, the reverse cascade validation curve peaks as well because at that stage of training only the linear layer was updated before training the previous blocks in a reversed fashion. Thus, both linear layer and R-cascade are nearly equivalent

[1] Initially the research wanted to focus on GPT-2 fine-tuning but due to limited computational resources, DistilGPT-2 was selected to be able to apply all the sampling and fine-tuning techniques in time for the completion of the project.

at this time, the main difference is that the former is trained in batches of 32 and the latter in batches of 16 to accommodate for memory requirements on the GPU. The other models where only one block was trained at a time have relatively stable validation ROUGE-LSum scores with the last block trained model being the highest for the task weighted samples and the domain weighted samples. In the case of the unsampled training set, the cascade and first block trained models have similar performance with Cascade being slightly better. Only the validation curve of the fully trained model decreases to zero with training duration.



a) No sampling

b) Task weighted sampling

c) Domain weighted sampling

Figure 2: Training Loss (top) and Validation ROUGE L-sum scores (bottom) per Epoch

Epoch (in order of sampling)	Model	-			task weighted			domain weighted		
		All	Train	Evaluation	All	Train	Evaluation	All	Train	Evaluation
-	Raw DistilGPT-2	1.73	4.92	0.93	-	-	-	-	-	-
3, 4, 2	Middle	2.13	5.89	1.19	2.48	6.2	1.55	2.22	5.94	1.29
1, 0, 0	All	2.33	6.47	1.29	2.32	6.45	1.28	0.88	2.08	0.58
0, 6, 7	Last	2.53	5.40	1.82	3.25	**11.08**	1.29	2.97	**10.23**	1.15
2, 7, 1	Cascade	2.58	5.76	1.78	2.88	6.04	2.09	2.51	6.14	1.6
10, 4, 1	First	2.60	6.19	1.70	2.81	6.86	1.79	2.51	6.14	1.6
2, 2, 2	R-Cascade	7.97	6.11	8.44	8.14	6.12	8.65	8.75	6.31	9.36
4, 5, 5	Linear layer	**9.40**	**6.73**	**10.06**	**8.26**	6.11	**8.8**	**8.78**	6.3	**9.39**

Table 1: Testing ROUGE-LSum scores

To analyze, the models on the test datasets, the models were selected at the epoch and model checkpoint where the validation Rouge-LSum score was highest. Overall, the linear layer fine-tuned models perform best on the test data as a whole and on the evaluation tasks observations. Note again that reverse cascade and linear layer fine-tuning are near equivalent but differ in batch sizes during training, showing how training choices can impact the final results. For the out-of-sample training observations, one can observe that training the last block, together with the linear layer, reaches the highest Rouge-Lsum scores for both the task-weighted training sample and the domain weighted training sample. This implies that taking into account the difference in numbers of observations per task does improve the multitasking performance on training like test observations. In addition, domain weighting the training data does have an advantage over task weighting on the evaluation tasks. Nonetheless using all of the training observations and training the linear layer leads to the highest ROUGE-LSum score on the evaluation tasks, being even higher than the performance on out-of-sample training observations. Thus, the linear layer fine-tuned model was able to beat the pre-trained DistilGPT-2 by a factor of 10 for the evaluation tasks and the last block fine-tuned and task weighted model by a factor of 2 for the out-of-sample training observations. A last observation is that, on average, the best scores were obtained after a few training epochs.

Analysis

Taking a deeper look at the results, we can see that the average ROUGE-LSum per task is consistent with prior findings where the best result comes from training the linear layer with no training sampling procedure. In addition, domain weighed and task weighted training samples are more likely to lead to better testing results, judging from the ranking in Table 2. However, for held-out training tasks observations, the average ROUGE-LSum per task is highest for linear layer fine-tuning with no sampling procedure, compared to the prior finding that task-weighted training with last block fine-tuning led to an overall best ROUGE-Lsum scores on training-like observations. Thus, there are some tasks in which the model performs better than in others and which have more representation in the held-out training observations. For indicative purposes, average ROUGE-LSum scores per task for GPT2-Large are shown, but the results are not directly comparable as these were computed using all of the BIG-bench observations per task in the preliminary stages of the research. However, they show that the smaller fine-tuned DistitGPT-2 reaches better scores on training tasks and similar scores for the evaluation tasks, despite having 692M fewer parameters.

Sampling	Model	All Tasks	Training Tasks	Evaluation Tasks
domain weighted	All	1.38	1.62	0.43
-	Raw DistilGPT-2	4.51	5.40	0.96
-	Cascade	4.73	5.55	1.46
-	Middle	4.78	5.75	0.88
domain weighted	Middle	4.89	5.86	1.01
task weighted	Cascade	5.06	5.90	1.69
-	First	5.10	6.04	1.35
task weighted	Middle	5.17	6.18	1.12
-	Last	5.27	6.15	1.77
domain weighted	Cascade	5.45	6.47	1.37
domain weighted	First	5.45	6.47	1.37
-	All	5.69	6.85	1.08
domain weighted	Last	5.71	6.88	0.99
task weighted	All	5.76	6.98	0.92
task weighted	Last	6.00	7.18	1.30
task weighted	First	6.05	7.18	1.54
task weighted	R-Cascade	6.89	7.15	5.84
task weighted	Linear layer	6.94	7.13	6.16
domain weighted	Linear layer	7.08	7.21	6.57
domain weighted	R-Cascade	7.13	7.33	6.33
-	R-Cascade	7.32	7.75	5.58
-	Linear layer	**7.79**	**8.01**	**6.90**
-	GPT-2 Large	6.05	5.85	6.89

Table 2: Average Rouge-LSum score per task

Finally, domain scores are analyzed by mapping the ROUGE-LSum cores by pairwise keyword combinations in Figure 3. In the lower left, we can see that the highest scores are obtained for combinations of domains: logical reasoning, mathematics, numerical response, and contextual question-answering. These refer specifically to the tasks 'key_value_maps' and 'sufficient_information' which increased by a factor of 20 with fine-tuning, and 'identify_math_theorems' on which pre-trained DistilGPt-2 was already performing well (see Appendix Table 2). On the evaluations tasks, the unseen domains in which the fine-tuned model performed best were for the keyword combination emotional understanding and emotional intelligence ('social_support' task), followed by the intent recognition, dialogue system combination ('intent_recognition' task) on which pre-trained DistilGPT-2 performed poorly. This is a surprising finding as the tasks related to the emotional realm were not expected to perform as well with a model that shows more expertise in mathematics and logical reasoning. Interestingly, it shows that a model is able to transfer knowledge from training to unseen tasks and that more research is necessary to, potentially, understand the mechanism.

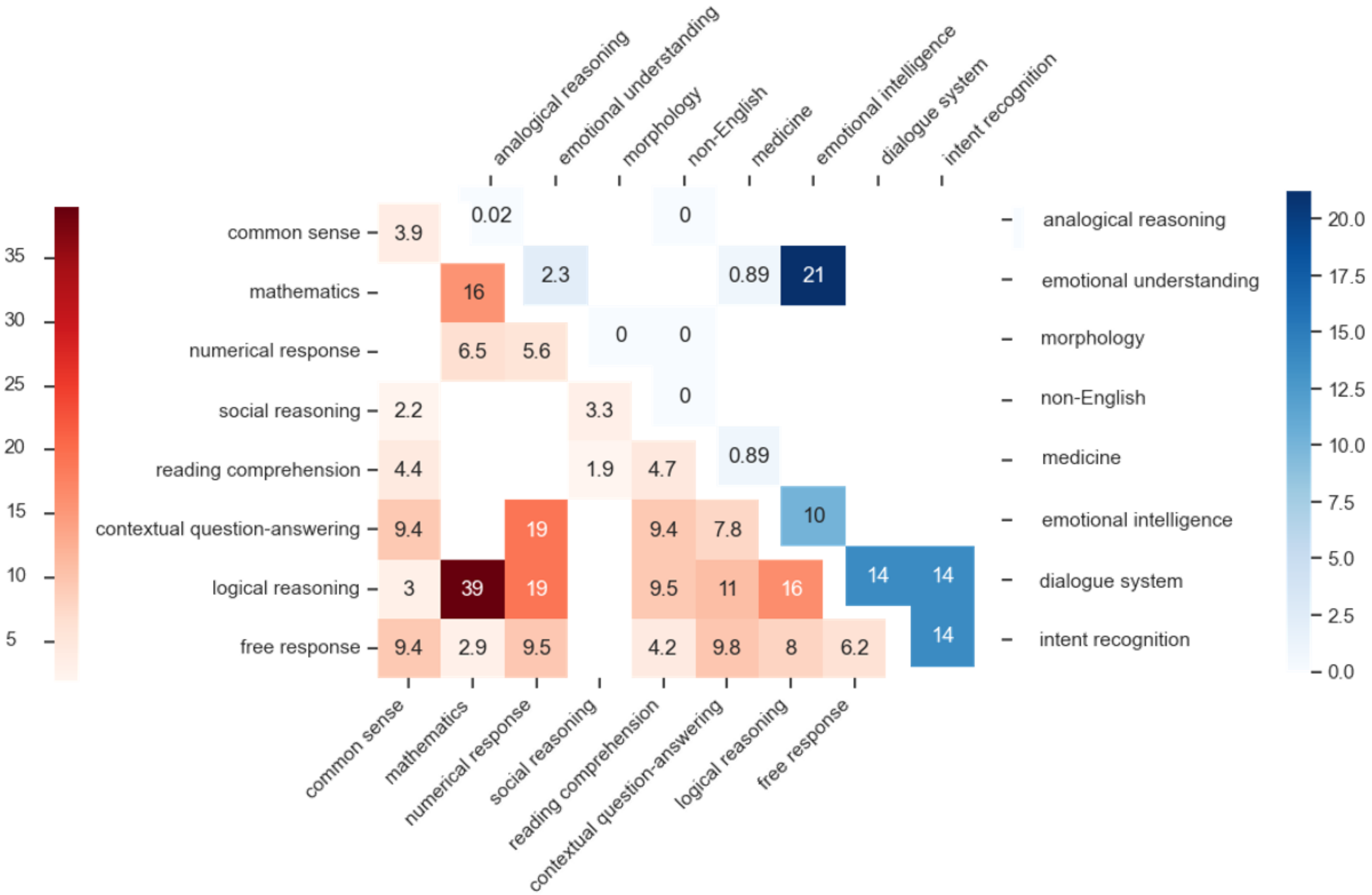


Figure 3: Rouge L-sum scores per combination of domains for out-of-sample train tasks (in red) and unseen domain evaluation tasks (in blue) using the unweighted DistilGPT-2 with fine-tuned linear layer

Conclusion

In summary, the aim of the research was to investigate how to make small models perform better on unseen tasks in domains they had not been trained on. BIG-bench tasks were separated to have no overlap in domains, or keywords, between training and evaluation tasks. Next, different sampling techniques (no sampling, task-weighted sampling, domain-weighted sampling) were applied to construct the training datasets which in turn were used to fine-tune DistilGPT-2 in various ways.

These consisted in updating certain blocks only at a time, all of the parameters, blocks in a consecutive and reversed way, and simply by training the last linear layer. The last approach proved to perform best on unseen domain tasks and did not suggest the need to task- or domain-weight the training observations. Thus, the combination of the pre-trained parameters of DistilGPT-2 with the updated linear layer showed that a model was able to get better at training tasks as well as unseen evaluation tasks.

Specifically, the analysis showed that the model learned to perform well on trained tasks related to mathematics, logical reasoning, contextual question-answering and numerical response, and improve metrics related to emotional understanding and emotional intelligence by a factor of 20 compared to pre-trained DistilGPT-2 results. The experiment also showed that this result does not need long training time and that performance on unseen domain tasks peaks at a very specific moment during the training procedure. It also showed that low training loss may indicate good performance on trained tasks, but the model will not necessarily adapt well to unseen domains the more the model is specialized in certain tasks even if the training set covers multiple domains. In addition, the results show that larger parameterized models are not necessarily the key to making models better, but that we need to develop stronger model learning procedures and understanding that will ultimately reduce the computational resources required to power the models in the long run.

The finding is encouraging in the sense that it indicates that knowledge can be transferred between tasks that we would not necessarily associate with each other. Therefore, more research is necessary to understand why this is possible, and more model interpretation and analysis techniques may help us in that regard. At last, it should humble us towards what we think works, and what does not, and it should encourage us to believe that machine learning models can help us uncover mechanisms we have yet to understand.

References

[1] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[2] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. Gpt-4 passes the bar exam. *Available at SSRN 4389233*, 2023.

[3] Alane Lim. What percentage of the human brain is used? `https://www.thoughtco.com/percentage-of-human-brain-used-4159438`, February 2018. Accessed: 2023-3-18.

[4] Chris Pointon. The carbon footprint of ChatGPT. `https://medium.com/@chrispointon/the-carbon-footprint-of-chatgpt-e1bc14e4cc2a`, December 2022. Accessed: 2023-3-18.

[5] Aitor Lewkowycz, Ambrose Slone, Anders Andreassen, Daniel Freeman, Ethan S Dyer, Gaurav Mishra, Guy Gur-Ari, Jaehoon Lee, Jascha Sohl-dickstein, Kristen Chiafullo, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. 2022.

[6] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States, July 2022. Association for Computational Linguistics.

[7] Kundan Krishna, Saurabh Garg, Jeffrey P Bigham, and Zachary C Lipton. Downstream datasets make surprisingly good pretraining corpora. *arXiv preprint arXiv:2209.14389*, 2022.

[8] Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

[9] 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 Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Scao, Sylvain Gugger, and Alexander Rush. Transformers: State-of-the-art natural language processing. pages 38–45, 01 2020.

[10] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC*$^2 Workshop$, 2019.

[11] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[12] Rakesh Agarwal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference*, volume 487, page 499, 1994.

Appendix

BIG-bench Tasks	Keywords	train	validation	test
bridging_anaphora_resolution_barqa	common sense, reading comprehension, contextual QA., free response	619	7	22
causal_judgment	common sense, social reasoning, reading comprehension	152	9	29
common_morpheme	morphology, non-English	-	17	33
crash_blossom	common sense	22	6	10
discourse_marker_prediction	common sense	786	24	47
few_shot_nlg	free response	123	4	26
general_knowledge	common sense	54	2	14
geometric_shapes	mathematics, free response	288	19	52
hhh_alignment	common sense, emotional intelligence	179	9	33
identify_math_theorems	mathematics, logical reasoning	37	7	9
implicit_relations	social reasoning, reading comprehension	68	2	15
intent_recognition	dialogue system, intent recognition	-	160	533
international_phonetic_alphabet_nli	reading comprehension	101	5	20
key_value_maps	mathematics, logical reasoning	80	5	16
moral_permissibility	common sense, social reasoning, reading comprehension	274	17	51
movie_recommendation	emotional intelligence	-	131	369
nonsense_words_grammar	contextual question-answering, logical reasoning	34	5	11
object_counting	logical reasoning, free response	1000	-	-
operators	mathematics, numerical response, free response	168	12	30
parsinlu_qa	analogical reasoning	-	269	781
penguins_in_a_table	reading comprehension, logical reasoning, free response	120	11	18
presuppositions_as_nli	common sense, logical reasoning	688	15	32
semantic_parsing_in_context_sparc	contextual question-answering, free response	1124	9	22
semantic_parsing_spider	free response	1028	2	4
simple_arithmetic_json_subtasks	mathematics, numerical response, free response	15	4	11
social_support	emotional understanding, emotional intelligence	-	216	681
strange_stories	social reasoning, emotional understanding	140	7	27
sufficient_information	numerical response, contextual QA, logical reasoning, free response	23	3	13
suicide_risk	emotional understanding, medicine	-	15	25
swedish_to_german_proverbs	numerical response, analogical reasoning, non-English	56	1	15
symbol_interpretation	reading comprehension, logical reasoning	895	17	78
temporal_sequences	reading comprehension, logical reasoning	1000	-	-
Total		9074	1010	3027

Table 1: BIG-bench tasks information and train, validation and test splits

BIG-Bench Tasks	Keywords	Test obs.	DistilGPT-2	Fine-tuned Model
simple_arithmetic_json_subtasks	mathematics, numerical response, free response	11	0.00	0.00
crash_blossom	common sense	10	0.00	0.00
penguins_in_a_table	reading comprehension, logical reasoning, free response	18	0.35	0.00
general_knowledge	common sense	14	0.44	0.00
implicit_relations	social reasoning, reading comprehension	15	0.00	0.00
moral_permissibility	common sense, social reasoning, reading comprehension	51	0.00	0.00
nonsense_words_grammar	contextual question-answering, logical reasoning	11	0.73	0.91
presuppositions_as_nli	common sense, logical reasoning	32	0.88	3.04
causal_judgment	common sense, social reasoning, reading comprehension	29	0.06	3.45
discourse_marker_prediction	common sense	47	26.42	3.58
semantic_parsing_spider	free response	4	5.82	4.06
semantic_parsing_in_context_sparc	contextual question-answering, free response	22	4.26	4.70
few_shot_nlg	free response	26	9.29	7.14
operators	mathematics, numerical response, free response	30	0.87	8.89
bridging_anaphora_resolution_barqa	common sense, reading comprehension, contextual QA.	22	2.43	9.45
symbol_interpretation	reading comprehension, logical reasoning	78	3.47	11.73
sufficient_information	numerical response, contextual question-answering	13	1.60	19.12
key_value_maps	mathematics, logical reasoning	16	1.19	25.39
identify_math_theorems	mathematics, logical reasoning	9	57.59	63.14
common_morpheme	morphology, non-English	33	0.21	0.00
suicide_risk	emotional understanding, medicine	25	1.14	0.89
parsinlu_qa	analogical reasoning	781	0.14	1.25
movie_recommendation	emotional intelligence	369	1.70	4.27
intent_recognition	dialogue system, intent recognition	533	1.87	13.80
social_support	emotional understanding, emotional intelligence	681	0.71	21.21

Table 2: Test tasks ROUGE-LSum scores of DistilGPT-2 and the linear layer fine-tuned model on unsampled training observations