HyperT5: Towards Compute-Efficient Korean Language Modeling

Dongju Park* **Soonwon Ka*** **Kang Min Yoo[1]*** **Gichang Lee** **Jaewook Kang**
NAVER Cloud [1]Seoul National University
{dongju.park,soonwon.ka,kangmin.yoo}@navercorp.com

Abstract

Pretraining and fine-tuning language models have become the standard practice in industrial natural language processing (NLP), but developing and deploying general-purpose language models without the abundant computation or data resources is a real-world issue faced by smaller organizations or communities whose main focus is languages with less accessible resources (e.g., non-English). This paper explores the sequence-to-sequence (seq2seq) language model architecture as a more practical and compute-efficient alternative to the decoder-oriented approach (e.g., GPT-3), accompanied by novel findings in compute-optimality analyses. We successfully trained billion-scale Korean-language seq2seq language models that strongly outperform other competitive models in Korean benchmarks. Moreover, we demonstrate that such language models can be more efficiently utilized by employing a heavy pre-finetuning strategy, by showcasing a case study on dialog-task adaptation. Our case study shows that adopting language models with more readily available domain-specific unlabeled data greatly improves fine-tuning data efficiency in low-resource settings.

1 Introduction

Pretraining large-scale Transformer-based language models and finetuning them for specific tasks have become the cornerstone of modern NLP pipelines. Among various Transformer-based language model architectures proposed in the field, generative decoder-based architectures, such as the GPT family (Brown et al., 2020), have gained more traction from their impressive ability to scale well into large language models (LLMs) (Kaplan et al., 2020; Chowdhery et al., 2022) and follow high-level natural language instructions with few or even in the absence of demonstrations (Wei et al., 2022).

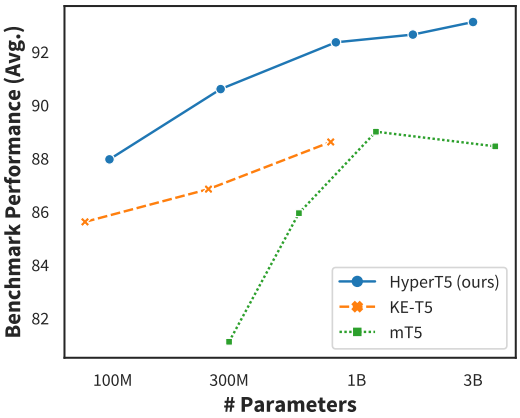


Figure 1: Benchmark of notable pretrained seq2seq language models with Korean capability. The benchmark is aggregated from key Korean understanding and reasoning tasks, including sentiment classification, topic classification, natural language inference, and reading comprehension. Our proposed model, HyperT5, strongly outperforms previous models including mT5 (Xue et al., 2021), a multilingual variant of the text-to-text transformer.

However, acquiring pretrained language models (PLMs) is a data- and compute-intensive process (Patterson et al., 2021), which many organizations cannot afford to pursue. This disparity is exacerbated for the communities of non-English or non-Latin languages (e.g., Korean) that have limited access to resources and share fewer linguistic features with English, making the cross-lingual transfer from the top language more challenging (Scao et al., 2022).

As a more cost-efficient alternative to the pure generative architecture, the sequence-to-sequence (seq2seq) Transformer (T5) (Raffel et al., 2022) may offer a reasonable middle ground between the generative LM and the encoder-oriented architecture (e.g., BERT (Devlin et al., 2019)), which are known to lack robustness in generation abilities. Additionally, T5 has been demonstrated to

* Equal contributions.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 5: Industry Track, pages 412–424

Data Source	Accessibility	Tokens
Blog	Proprietary	146.1B
Online Community	Proprietary	44.5B
News	Proprietary	39.4B
Crawled Comments	Proprietary	21.9B
Korean QA Website	Proprietary	14.6B
Modu Datasets	Public	3.2B
En. & Jp. Wikipedia	Public	2.8B
Others	Public / Proprietary	27.5B
Total		300B

Table 1: Data sources of the pretraining corpus.

be good few-shot learners (Liu et al., 2022a) and task/domain adapters (Aribandi et al., 2022; Gupta et al., 2022).

This paper aims to provide an industrial perspective on the language model pretraining strategies with small- to medium-scale budgets. We conduct compute-optimality analyses to find the optimal pretraining strategy given our compute budget and argue that the text-to-text Transformer architecture is a superior approach compared to decoder-only models under restricted compute resources (§5.1). Based on this finding, we share our experience with training HyperT5 (§3), the state-of-the-art seq2seq Transformer for the Korean language (Figure 1). Moreover, we showcase how HyperT5 can be further refined to improve data and modeling efficiency in specific domains (i.e. dialogs) using relatively abundant unlabeled resources (§4).

2 Related Work

PLMs and Efficiency As we gain more understanding of core PLM architectures (Devlin et al., 2019; Brown et al., 2020; Raffel et al., 2022) and their scaling laws (Kaplan et al., 2020), research efforts for improving their training efficiencies have diverged in various directions, including improving the scaling curve (Tay et al., 2022; Chung et al., 2022), maximizing optimality (Hoffmann et al., 2022), and efficient fine-tuning (Lester et al., 2021; Hu et al., 2022). Our work offers a comprehensive overview and case study of optimizing the language modeling efficiency for a less resourceful language.

Non-English Language Models Previously, significant works have explored pretraining language models on multiple languages to support low-resource languages and maximize cross-lingual knowledge transfer without explicit supervision (Devlin et al., 2019; Conneau and Lample, 2019; Xue et al., 2021; Scao et al., 2022). Recently, language models that target specific non-English languages started to become more common (Zeng et al., 2021; Kim et al., 2021a; Nagoudi et al., 2022; Fuadi et al., 2023), especially low-resource or non-Latin languages that share fewer commonalities with English. There have been several Korean text-to-text Transformers proposed in the past, such as KoBART* and KE-T5 (Kim et al., 2021b), but our work, among other Korean seq2seqs, is the first to systematically achieve powerful billion-scale seq2seq models and conduct extensive analyses in terms of efficiency and performance.

Dialog-Oriented Language Models Adapting language models for the purpose of building dialog agents has been a long-standing goal in the language modeling community (Zhang et al., 2020; Adiwardana et al., 2020; Roller et al., 2021). However, from the industrial application perspective, building dialog response generators is not the only task that can benefit from the advances in language modeling. In a more recent line of work, several approaches have been proposed to prepare language models for various dialog-related tasks (Mehri et al., 2019; Gu et al., 2021; Chen et al., 2022). In parallel to dialog adaptation, multi-task fine-tuning is another line of work that covers dialog-related tasks, as a subset of dialog-related tasks is included in the multi-task set, and expanding the task set to cover dialog-related tasks is trivial (Sanh et al., 2022; Aribandi et al., 2022).

3 HyperT5

This section describes the details of the pretraining corpus, pretraining strategy, and evaluation methods.

3.1 Pretraining Corpus

Inspired by the pretraining corpus proposed by Kim et al. (2021a), we design our pretraining data to cover a wide range of domains and data distributions to ensure that the model trained on top of the corpus will achieve robustness and generalizability. Various sources of written and spoken texts are included in the corpus (Table 1), although online web texts take a large portion of the data (Table 5). While online texts are certainly vulnerable to the compound effect of biases, the collectively massive and unfiltered nature provides a comprehensive impression of the language distribution

*https://github.com/SKT-AI/KoBART

Model	n_{layer}	d_{model}	n_{head}	d_{head}	d_{ff}
HyperT5$_{\text{SMALL}}$	16	512	6	64	1024
HyperT5$_{\text{BASE}}$	24	768	12	64	2048
HyperT5$_{\text{LARGE}}$	48	1024	16	64	2816
HyperT5$_{\text{1.7B}}$	48	1536	24	64	4096
HyperT5$_{\text{3B}}$	48	2048	32	64	5120

Table 2: Configurations of different model sizes.

(Scao et al., 2022). We conducted additional studies to investigate the feasibility of data composition re-adjustment through sampling (Appendix A.1). However, we found that the benefit was not clear-cut.

3.2 Pretraining Setup

Our research employs the transformer encoder-decoder architecture, similar to T5 of Google (Raffel et al., 2022). However, we have opted for the T5.1.1 structure, a variation of T5, due to its superior performance based on experimentation results. It is worth noting that our HyperT5$_{\text{1.7B}}$ model size is not a derivative of Google's T5, but rather an interpolation of the LARGE and the 3B model. Detailed information on the configuration of different model sizes can be found in Table 2.

For all model has been pre-trained on a total of 300B tokens, utilizing the replace corrupted spans method proposed by Google's T5 as one of their unsupervised objectives. Specifically, we set the corruption rate to 15%, while maintaining a mean span length of 3.

Furthermore, we employed the inverse square root learning rate schedule with 10k warmup steps at a learning rate of 0.01 when using the Adafactor optimizer (Shazeer and Stern, 2018). Both of our pretrained models were trained using a batch size of 1024 and a maximum sequence length of 512 for the encoder and decoder, respectively.

By using distributed data-parallel (Li et al.), we were able to parallelize the training process across multiple GPUs, effectively reducing the overall training time and enabling us to train larger models with higher performance. Specifically, we used 64 A100 GPUs for the small to large models and 1024 A100 GPUs for the 1.7B and 3B models.

3.3 Evaluation Methods

Benchmark The primary objective of the HyperT5 evaluation is to address various natural language processing tasks specific to the Korean language in a reproducible way. To quantify the effectiveness of our model in these tasks, we designed a series of benchmarking experiments that cover a wide range of tasks. The detailed components of our benchmark are described in Appendix A.2. Note that while all of our benchmark datasets are publicly available for reproducibility, some datasets (YNAT, KLUE-NLI, KLUE-STS, KorQuAD) have not made the test set publicly available, hence some of the report values are based on the development or validation set where the test set is inaccessible.

Baselines We compare not only structures that are identical to ours but also encoder and decode-exclusive architectures. Models based on the BERT and RoBERTa architecture released by KLUE (Park et al., 2021) are encoder-only models specialized for natural language understanding. On the other hand, HyperCLOVA (Kim et al., 2021a) is a decoder-only structure like GPT. Note that HyperCLOVA does not provide fine-tuning results, and thus, we compare the ICL and P-tuning (Liu et al., 2022b) results reported for this model. We also compare three models with the same structure as our model. KoBART has the encoder-decoder structure but follows the learning method and details of BART (Lewis et al., 2019). The mT5 (Xue et al., 2021) and KE-T5 (Kim et al., 2021b) models share the exact same structure as our HyperT5 model, with the difference being that mT5 is a multilingual model and KE-T5 is a Korean and English cross-lingual model.

3.4 Evaluation Results

Main Benchmark Results On our Korean benchmark, HyperT5 achieves state-of-the-art performances across all tasks (Table 3), outperforming other seq2seq architectures by a large margin. Specifically, the smallest version of our model (97M) was able to perform on par with the largest KE-T5 (large) on the average benchmark (87.96 vs 88.61). Compared to large-scale decoder architectures, our largest model (3B) is still able to outperform the 39B-scale HyperCLOVA with p-tuning (93.29 vs 93.00). Although a more comprehensive benchmark is desirable, the preliminary results on NSMC suggest that our approach has an advantage in scaling efficiency for the current compute-budget range (§5.1).

Parameter-Efficient Fine-Tuning To understand how our model can be further efficiently fine-tuned using parameter-efficient fine-tuning (PEFT) techniques, we benchmarked HyperT5 models that

Model	Params.	NSMC	YNAT	KLUE-NLI	KLUE-STS	KorQuAD	Avg.
Metrics		Acc.	F1	F1	Pearson	EM / F1	
Encoder-Only Pretrained Language Models							
KLUE-BERT$_{\text{BASE}}$	110M	-	85.73*†	81.63*†	90.85*†	-	-
KLUE-RoBERTa$_{\text{SMALL}}$	68M	-	84.98*†	79.33*†	91.54*†	-	-
KLUE-RoBERTa$_{\text{BASE}}$	125M	-	85.07*†	84.84*†	92.50*†	-	-
KLUE-RoBERTa$_{\text{LARGE}}$	355M	91.44	85.69*†	89.17*†	93.35*†	-	-
Decoder-Only Pretrained Language Models							
HyperCLOVA (ICL)	13B	87.2*	-	-	-	-	-
	39B	88.0*	-	-	-	-	-
	82B	88.2*	-	-	-	-	-
HyperCLOVA (P-Tuning)	137M	87.2*	-	-	-	-	-
	13B	91.7*	-	-	-	-	-
	39B	93.0*	-	-	-	-	-
Encoder-Decoder Pretrained Language Models							
KoBART$_{\text{BASE}}$	124M	90.24*	-	-	-	-	-
mT5$_{\text{SMALL}}$	300M	88.82	83.57	70.18	80.95	70.83 / 82.02	81.11
mT5$_{\text{BASE}}$	580M	89.59	86.57	78.27	89.09	75.74 / 86.17	85.94
mT5$_{\text{LARGE}}$	1.2B	90.81	87.17	89.96	91.69	80.03 / 88.35	89.00
mT5$_{\text{XL}}$	3.7B	90.34	86.58	87.20	90.58	78.58 / 87.53	88.45
KE-T5$_{\text{SMALL}}$	77M	89.78	86.44	74.37	87.55	80.98 / 89.91	85.61
KE-T5$_{\text{BASE}}$	247M	89.75	86.58	77.58	88.35	83.46 / 91.94	86.84
KE-T5$_{\text{LARGE}}$	783M	91.09	86.94	86.15	86.15	84.19 / 92.72	88.61
HyperT5$_{\text{SMALL}}$	97M	90.91	87.31	79.43	90.32	83.03 / 91.82	87.96
HyperT5$_{\text{BASE}}$	277M	91.82	87.83	87.48	91.87	85.97 / 93.98	90.60
HyperT5$_{\text{LARGE}}$	822M	93.02	88.31	92.39	93.09	87.98 / 94.95	92.35
HyperT5$_{\text{1.7B}}$	1.7B	93.11	88.43	93.02	93.43	88.32 / 95.22	92.64
HyperT5$_{\text{3B}}$	3B	**93.29**	**88.65**	**94.07**	**93.98**	**88.74 / 95.58**	**93.11**

* Reported by the authors. † Reported on the test set, which is not publicly available.

Table 3: Korean understanding and reasoning benchmark results for Korean language models of various architectures. Our model significantly outperforms all other models, regardless of size and architecture.

Model	Params.	NSMC	YNAT	KLUE-NLI	KLUE-STS	Avg.
Metrics		Acc.	F1	F1	Pearson	
LST (Sung et al., 2022)						
HyperT5$_{\text{SMALL}}$	1.3M	88.96 (-2.14%)	85.33 (-2.27%)	72.15 (-9.17%)	86.61 (-4.11%)	83.26 (-4.29%)
HyperT5$_{\text{BASE}}$	5.1M	89.80 (-2.20%)	86.34 (-1.70%)	78.87 (-9.84%)	89.09 (-3.03%)	86.03 (-4.15%)
HyperT5$_{\text{LARGE}}$	17.9M	91.21 (-1.95%)	88.35 (+0.05%)	86.61 (-6.26%)	91.14 (-2.09%)	89.33 (-2.59%)
HyperT5$_{\text{1.7B}}$	39.8M	91.77 (-1.44%)	88.50 (+0.08%)	89.18 (-4.13%)	91.65 (-1.91%)	90.28 (-1.87%)
HyperT5$_{\text{3B}}$	69.3M	92.02 (-1.36%)	88.10 (-0.62%)	90.10 (-4.22%)	92.00 (-2.11%)	90.56 (-2.10%)
LoRA (Hu et al., 2022)						
HyperT5$_{\text{SMALL}}$	0.2M	88.96 (-2.14%)	85.29 (-2.31%)	73.25 (-7.78%)	87.95 (-2.62%)	83.86 (-3.60%)
HyperT5$_{\text{BASE}}$	0.5M	90.60 (-1.33%)	86.31 (-1.73%)	84.43 (-3.49%)	91.02 (-0.93%)	88.09 (-1.85%)
HyperT5$_{\text{LARGE}}$	1.3M	92.22 (-0.86%)	88.12 (-0.22%)	91.46 (-1.01%)	92.68 (-0.44%)	91.12 (-0.64%)
HyperT5$_{\text{1.7B}}$	2M	92.63 (-0.52%)	88.58 (+0.17%)	91.55 (-1.58%)	92.97 (-0.49%)	91.43 (-0.61%)
HyperT5$_{\text{3B}}$	2.7M	93.19 (-0.11%)	88.47 (-0.20%)	93.43 (-0.68%)	93.44 (-0.57%)	92.13 (-0.39%)

Table 4: Parameter-efficient fine-tuning (PEFT) benchmarked on HyperT5. The relative performance loss in percentage is shown next to the corresponding results. Overall, a minor performance loss is observed across all tasks and PEFT techniques, despite using a small number of trainable parameters.

are fine-tuned using LoRA (Hu et al., 2022) and Ladder-Side Tuning (LST) (Sung et al., 2022), respectively, and compared the performances against the full fine-tuning results in Table 3. Results show that the performance degradation of employing PEFT compared to the full fine-tuning baseline is less than 5% on average, while the ratio of parameters used for training is less than 2.3%. And as the model scales larger, the issue of performance degradation is relatively alleviated, falling to 0.39% for HyperT5$_{3B}$ with LoRA. Model scaling and the specific PEFT technique to employ will be the key strategic factors for large-scale deployment.

4 Case Study: Efficient Adaptation for Dialog-Oriented Tasks

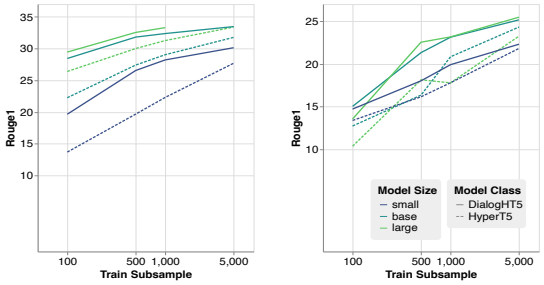
Domain and task-family adaptation can further improve the utilization of PLMs in low-resource settings (Maronikolakis and Schütze, 2021). This section explores the use case for adapting HyperT5 to dialog-related tasks.

4.1 Training Setup

For dialog adaptation, we propose to heavily train HyperT5 on a 1B-token unlabeled dialog-oriented data, with the *multiple utterance masking* (MUM) objective in the curriculum learning setting. We take the replace corrupted spans method to a more challenging strategy, MUM, to help the model hold a better understanding of dialog structures. During training, multiple utterances are randomly masked per dialog session with a pre-defined corruption rate. We further adopt curriculum learning to gradually raise the training difficulty by increasing the MUM corruption rate. HyperT5 models, from small to large, are trained for 5 epochs with a global batch size of 64 using 2 A100 GPUs. Like pretraining, the dropout rate is set to 0. MUM corruption rate sweeps sigmoidally from 5% to 40%.

Training Data We collect a dialog-oriented training corpus from both open-sourced and proprietary Korean dialog datasets (Appendix B.1). The corpus consists of 3.3M dialog sessions[†] in various domains (e.g. social chats, customer service, broadcast transcripts, etc.). The resulting corpus provides a wide range of topics and aspects of different dialog-oriented tasks, making it suitable for dialog adaptation.

[†]We preprocessed the dialog corpus by truncating and splitting the original dialogs into up to 20-turn sessions.



(a) Dialog in-filling. (b) Dialog resp. generation.

Figure 2: Data efficiency analysis. (AI Hub ToD).

4.2 Evaluation Methods

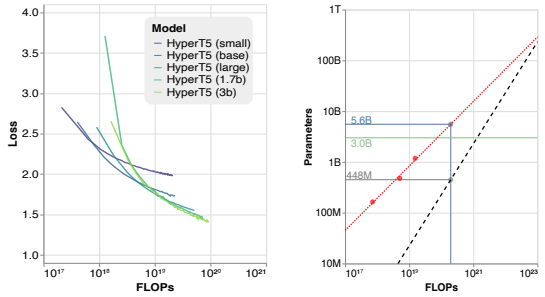
We conducted an extended series of benchmarking experiments for our dialog-adapted models (DialogHT5) in both scarce and full data settings. The benchmark results consist of three generative tasks, i.e., dialog in-filling (DI), dialog response generation (DR), and dialog summarization (DS), and one discriminative task, i.e., dialog classification (DC). We only use the open-sourced Korean dialog-oriented datasets from AI Hub[‡]. AI Hub task-oriented dialog (ToD) and open-domain dialog (ODD) datasets are used for both DI and DR, while AI Hub broadcasting media transcript summarization dataset (Script-Summ) for DS. Since the AI Hub ToD dataset is annotated with dialog topics, we used the dataset for DC (Appendix B.2).

4.3 Dialog Benchmark Results

Data Scarce Experiments To investigate the data efficiency of DialogHT5, we first compare their performance against HyperT5 over scarce data settings. We benchmarked DialogHT5 and HyperT5 as sweeping the number of training samples from 0.1k to 5k. Results describe that DialogHT5 mostly achieves a higher score against HyperT5 (more details in Appendix B.3). Especially, DialogHT5 outperforms HyperT5 with the gain of up to 8 R1 score in the dialog in-filling task, as shown in Figure 2. DialogHT5$_{SMALL}$ can save the in-domain data resource approximately ten times to score tie with HyperT5$_{SMALL}$ of 5k training samples (Figure 2a), which highlights the effectiveness of dialog adaptation.

Full Data Experiments Full data benchmark results show that DialogHT5 obtains higher performance compared to HyperT5 in all the experiments (Appendix B.4). Generative tasks show a gain of

[‡]https://aihub.or.kr/



(a) Validation loss curves. (b) Optimal model sizes.

Figure 3: Compute-optimality analysis. Based on the validation loss curves obtained from our pretraining experiments (shown left), we plot the best model sizes per compute level on the right. Using the limited optimality data points, we are able to safely fit a log-log linear line and extrapolate (red line). The regression indicates that the optimal model size for our compute budget is 5.6B, which is less than a binary order of difference with the largest model size we attained.

up to 0.6 R1 score whereas the discriminative task shows a gain of 0.7 F1 score.

5 Discussions

5.1 Compute-Optimality Analysis

To investigate whether the model configurations we experimented with are optimal given our pretraining compute budget and the pretraining tokens, we conducted compute-optimality analyses, similar to the work done by Hoffmann et al. (2022). The loss curves of our pretraining experiments were smoothed and interpolated as shown in Figure 3a. Using the curves, we map out the optimal model sizes for each given compute level. However, due to the very small number of model-size samples, we need to normalize the optimal model-size data points by selecting the mid-point of each optimality segment[§], as shown in Figure 3b. After fitting the regression line ($r^2 = 0.988$), we discover that the size of our largest model lies very close to the predicted optimal model size for our compute budget. Moreover, the predicted optimal model size (dashed line in the same figure) for the decoder-only architecture is significantly smaller (at 448M), but our benchmark results on NSMC (Table 3) show that small-scale decoder LMs (i.e., HyperCLOVA) falls behind in terms of performance, supporting the notion that seq2seq architectures are more economically viable for small and medium-scale compute budgets[¶].

5.2 Practical Advantages of Seq2Seq

Apart from the quantitative benefits in performance and efficiency demonstrated throughout the paper, Seq2Seq offers additional practical and real-world benefits. First, the encoder-decoder framework produces a parameter-efficient **text encoder as a by-product**, which can be utilized for extracting features and encoding purposes (Ni et al., 2022; Liu et al., 2021). Specifically, the encoder module extracted from seq2seq is capable of producing high-quality text embeddings superior to ones produced from encoders of similar sizes (Ni et al., 2022).

Second, the text-to-text architecture reduces the software complexity and management costs for large-scale deployment, as a result of (1) the unified nature of the input and output format, (2) the separation of the input and output sequences inherently supported by the encoder-decoder architecture, and (3) better parameter-efficiency. This translates to fewer engineering resources to support the same level of deployment scalability. The unified text nature of the data format allows existing deployment infrastructures to be easily expanded to handle new tasks. Also, the inherent distinct two-part architecture enables simpler and more streamlined serving infrastructure. Furthermore, due to the steeper model-scaling curve exhibited by decoder architectures, text-to-text transformers incur fewer operating costs to maintain the same quality of services.

6 Conclusion

In this paper, we introduced HyperT5 and DialogHT5 as state-of-the-art on Korean language modeling. We also demonstrated the feasibility of performing resource-aware strategization for language models. Through the compute-optimality analyses, we found that the seq2seq architecture may be more cost-efficient than decoders below a certain compute-budget threshold. For future work, we wish to generalize the domain adaptation approach and study the efficacy of multi-task learning (Aribandi et al., 2022) from the industrial perspective. Furthermore, we look forward to conducting comprehensive investigations into cross-architectural optimality.

[§]The compute range, where the smallest and the largest model sizes are chosen to be the optimal model, is omitted to prevent skewness.

[¶]Conversely, this means that decoder-only architectures scale better with larger compute budgets (Figure 3b).

Ethics Statement

The authors are aware that the language models proposed in this paper, either pretrained from our pretraining corpus or heavily fine-tuned using the dialog corpus (Appendix B.1), are all subject to social and unethical biases depending on the way the corpora were prepared. Internally, the authors and the relevant members of the affiliated organization are actively working to make sure that the deployed language models do not generate ethically questionable content that may cause harm or stress to the end user. The specific set of actions that we take include but are not limited to,

- Employing automated models to detect unethical content and perform automatic adversarial attacks on the language model before deploying them into services and products.
- Under safe and strict ethical guidelines, conducting human studies to identify prompts that could potentially cause the language model to generate unethical content. (red-teaming)
- Establishing strategies to mitigate or amend ethical issues exhibited by the language models raised from automated and human surveys.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Zhi Chen, Jijia Bao, Lu Chen, Yuncong Liu, Da Ma, Bei Chen, Mengyue Wu, Su Zhu, Jian-Guang Lou, and Kai Yu. 2022. Dialogzoo: Large-scale dialog-oriented task learning. *arXiv preprint arXiv:2205.12662*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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

Mukhlish Fuadi, Adhi Dharma Wibawa, and Surya Sumpeno. 2023. idt5: Indonesian version of multilingual t5 transformer. *arXiv preprint arXiv:2302.00856*.

Xiaodong Gu, Kang Min Yoo, and Jung-Woo Ha. 2021. Dialogbert: Discourse-aware response generation via learning to recover and rank utterances. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12911–12919.

Raghav Gupta, Harrison Lee, Jeffrey Zhao, Yuan Cao, Abhinav Rastogi, and Yonghui Wu. 2022. Show, don't tell: Demonstrations outperform descriptions for schema-guided task-oriented dialogue. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4541–4549, Seattle, United States. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large

language models. In *International Conference on Learning Representations*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Jeon Dong Hyeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, Heungsub Lee, Minyoung Jeong, Sungjae Lee, Minsub Kim, Suk Hyun Ko, Seokhun Kim, Taeyong Park, Jinuk Kim, Soyoung Kang, Na-Hyeon Ryu, Kang Min Yoo, Minsuk Chang, Soobin Suh, Sookyo In, Jinseong Park, Kyungduk Kim, Hiun Kim, Jisu Jeong, Yong Goo Yeo, Donghoon Ham, Dongju Park, Min Young Lee, Jaewook Kang, Inho Kang, Jung-Woo Ha, Woomyoung Park, and Nako Sung. 2021a. What changes can large-scale language models bring? intensive study on HyperCLOVA: Billions-scale Korean generative pretrained transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3405–3424, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

San Kim, Jin Yea Jang, Minyoung Jung, and Saim Shin. 2021b. A model of cross-lingual knowledge-grounded response generation for open-domain dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 352–365, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. Pytorch distributed: Experiences on accelerating data parallel training. *Proceedings of the VLDB Endowment*, 13(12).

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. Korquad1. 0: Korean qa dataset for machine reading comprehension. *arXiv preprint arXiv:1909.07005*.

Frederick Liu, Siamak Shakeri, Hongkun Yu, and Jing Li. 2021. Enct5: Fine-tuning t5 encoder for non-autoregressive tasks. *arXiv preprint arXiv:2110.08426*.

Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Antonis Maronikolakis and Hinrich Schütze. 2021. Multidomain pretrained language models for green NLP. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 1–8, Kyiv, Ukraine. Association for Computational Linguistics.

Shikib Mehri, Evgeniia Razumovskaia, Tiancheng Zhao, and Maxine Eskenazi. 2019. Pretraining methods for dialog context representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3836–3845, Florence, Italy. Association for Computational Linguistics.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. AraT5: Text-to-text transformers for Arabic language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 628–647, Dublin, Ireland. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Taehwan Oh, et al. 2021. Klue: Korean language understanding evaluation. *arXiv preprint arXiv:2105.09680*.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

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*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. LST: Ladder side-tuning for parameter and memory efficient transfer learning. In *Advances in Neural Information Processing Systems*.

Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q Tran, David R So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, et al. 2022. Transcending scaling laws with 0.1% extra compute. *arXiv preprint arXiv:2210.11399*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

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)*, pages 1112–1122. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, et al. 2021. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

A Supplementary Materials Regarding HyperT5 Pretraining

A.1 Experiments on Corpus Composition Re-sampling

Data Type	Tokens	Ratio
Web-Crawled	183.0B	61.0%
Books	61.2B	20.4%
News	55.8B	16.8%
Others	0.3B	0.1%
Korean	238.5B	79.5%
English	61.5B	20.5%

Table 5: Data composition of the experimental pretraining corpus by data type and language. This experimental pretraining corpus of HyperT5 was designed to contain a relatively higher proportion of knowledge-intensive data sources such as books and news. Moreover, a higher English proportion was employed to leverage high-resource language and promote the emergence of cross-lingual knowledge transfer.

Model (corpus) Metrics	**NSMC** Acc.	**YNAT** F1	**KLUE-NLI** F1	**KLUE-STS** F1 / Pearson	**Abs. Summ.** R1 / R2 / RL	**Avg.**
$\text{HyperT5}_{\text{BASE}}$ (original corpus)	**91.52**	**87.72**	**85.25**	84.95 / 93.27	52.70 / 24.81 / 49.53	71.22
$\text{HyperT5}_{\text{BASE}}$ (re-sampled corpus)	91.51	87.15	84.10	**85.50 / 93.49**	**53.64 / 25.24 / 50.27**	**71.36**

Table 6: Pretraining results on the experimental corpus re-sampled with an emphasis on knowledge-heavy data. Note that the pretraining setup is slightly different from the one described in the main section of the paper, hence the results of the base model may differ.

During pretraining, language models consume billions of weakly preprocessed tokens, which may impact the model performance to varying degrees. For example, web-crawled data which take up a large portion of the pretraining corpus is relatively noisy, thus prioritizing certain data sources that are thought to be dense with information may help to improve training efficiency, i.e., the number of tokens needed to converge towards a reasonable level of performance.

To investigate our hypothesis, we create an experimental pretraining corpus (Table 5) and made sure that the proportions of data sources that provide "hard" knowledge (e.g., books and news) are significantly higher by under-sampling other data types (Rae et al., 2021). Additionally, we augment the experimental pretraining corpus with English data sources, leveraging knowledge embedded in the world's most resource-accessible language. We theorized that the availability of weak-parallel corpora such as multi-lingual Wikipedia articles acts as mediums for cross-lingual knowledge transfer (Hu et al., 2020). As shown in the results (Table 6), the base version of our model improved in reading comprehension and abstract summarization tasks but slightly suffered in classification and NLI tasks, suggesting that composition of knowledge-oriented data sources in the pretraining corpus may help the language model in tasks related to language generation (answer generation and summary generation) at the risk of slight underfitting in discriminatory power.

A.2 Benchmark Datasets

This section provides more details on the benchmark datasets.

- **NSMC**[‖] is a movie review dataset constructed from NAVER Movie, a Korean movie review website, and consists of 150k training data and 50k test data samples, labeled with positive and negative classes.

- **YNAT** is a news-topic classification dataset as a part of the Korean Language Understanding Evaluation (KLUE) (Park et al., 2021) benchmark set. The dataset consists of titles for news articles

[‖] https://github.com/e9t/nsmc

and the corresponding news topic labels. The dataset has 45.6k training data samples, 91k validation data samples, and 91k test data samples.

- **KLUE-NLI** is a natural language inference dataset from the KLUE benchmark set. Similar to MNLI (Williams et al., 2018), each sample in the dataset contains a pair of premise-hypothesis sentences, and the goal is to label the pair with one of "entailment", "neutral", or "contradiction". It comprises 25k training data, 3k validation data, and 3k test data samples.

- **KLUE-STS** is designed to evaluate a model's ability to capture the semantic similarity between two sentences. Like YNAT and KLUE-NLI, KLUE-STS is also a part of the KLUE benchmark. The dataset consists of 11.6k training data, 519 validation data, and 1k test data samples.

- **KorQuAD** is a Korean question Answering dataset for machine reading comprehension (Lim et al., 2019), similar to SQuAD (Rajpurkar et al., 2016). The dataset consists of 60k question/answer pairs for training, 5.8k for validation, and 3.9k for testing.

B Supplementary Materials Related to Dialog-Oriented Adaptation

This appendix section contains supplementary materials related to the dialog-oriented adaption of HyperT5.

B.1 Dialog-oriented Adaptation Corpus

Here is the detailed list of data sources for constructing the dialog heavy-finetuning corpus presented (Table 7). The data consist of both open-sourced (Modu, AI Hub) and proprietary dialog corpus. Modu datasets are a collection of various dialog-oriented datasets collected by National Institute of Korean Language (NIKL)[**].

Source	Dataset	Dialog Type	Domain	# Dialogs	# Turns
Modu	TV Series, News	Spoken	Broadcast Contents	0.1M	2.2M
	Open-ended dialogs	Spoken	General	51.1k	1M
	SNS dialogs	Written	General	24.2k	0.5M
	Online communications	Written	Online Communications	98k	1.7M
	Korean parliamentary records	Spoken	Politics	0.3M	5.5M
AI Hub	Customer service QAs	Spoken	Customer Service	6.7k	0.1M
	Empathetic dialogs	Spoken	Empathetic dialog	45.5k	0.3M
	Dialog summarization	All	General	0.3M	3.5M
	Open-ended SNS dialogs	Written	Online Communications	1.8M	28.6M
	Shopping, Public sector, Finance QA	Spoken	Customer Service	0.1M	1.9M
Proprietary	TV Series, News	Spoken	Broadcast Contents	0.2M	3.3M
	Shopping QAs	Written	Customer Service	0.2M	1M
	Elderly care dialogs	Written	Empathetic dialog	40k	0.4M
	Character chatbot dialogs	Written	Empathetic dialog	32.1k	0.3M
Total				3.3M	50.2M

Table 7: Full list of data sources and corresponding statistics for dialog-oriented heavy-fine-tuning.

B.2 AI Hub Benchmark Datasets

This section provides more details on the dialog-oriented benchmark datasets. Note that the benchmark datasets are excluded from the dialog adaptation corpus.

- **AI Hub ToD** is a task-oriented dialog (ToD) dataset from AI Hub[††], which covers 20 different topics (restaurant booking, online shopping QA, etc.). We preprocess the corpus to build 38.5k of training data and 3.9k of test data. We used the ToD dataset for dialog in-filling and dialog response

[**]https://corpus.korean.go.kr/
[††]https://bit.ly/3S9Wxi6

generation. Each dialog session produces 3 or 4 training samples by random utterance selection. For the dialog response generation task, the turns before the selected utterance are only used for the dialog context, whereas both turns before and after the selected utterance are given as the context for dialog infilling. We also benchmarked the dialog topic classification using the ToD dataset with topic labels.

- **AI Hub ODD** is an open-domain dialog (ODD) dataset from AI Hub[‡‡], over 20 different topics (social issues, food, marriage, etc.). We built a dataset with 87.7k training data and 11.0k of test data for the aforementioned tasks in AI Hub ToD. Similarly to AI Hub ToD, each dialog session results in multiple training samples.

- **AI Hub Script-Summ** is a broadcasting media transcript summarization dataset from AI Hub. We built a dataset with 84.4k training data and 10k test data for dialog summarization. Finally, we use the ROUGE score for generation task evaluation and Macro F1-Score for classification.

B.3 Scarce Data Benchmark Results

Table 8 illustrates the scarce data benchmark results for our dialog-adapted models against HyperT5 models as baselines. We averaged the experimental results over five different random seeds. All the experiments are under the early stopping option with a patience level of 5. For each experiment, the best checkpoint is determined according to the evaluation metric. We set the learning rate to 5e-4 with linear learning decay.

Note that DialogHT5 shows a huge performance leap in DI tasks. This can be explained by the fact that dialog in-filling is essentially a single utterance masking (SUM), hence the MUM objective we used for dialog adaptation is a more challenging version of dialog in-filling.

In the extreme data-scarce settings (i.e., the training sample number of 0.1k), both HyperT5 and DialogHT5, regardless of the model size, tend to fail training without hyperparameter tuning on the learning rate. In general, using 5e-4 instead of 5e-5 enables tuning to begin working.

B.4 Full Data Benchmark Results

We further conduct the full data benchmarks. Results show that DialogHT5 models achieve higher scores compared to HyperT5 models in most cases (Table 9).

[‡‡] https://bit.ly/3kc75R5
https://bit.ly/3Izjmsv

Model Dataset	Params.	# Samples	DI ToD	DI ODD	DR ToD	DR ODD	DS Script-Summ	DC ToD
Metrics			R1	R1	R1	R1	R1	F1
HyperT5$_{\text{SMALL}}$	97M	100	13.69	8.58	13.38	8.82	27.03	30.13
		500	19.68	13.83	16.15	10.52	35.35	51.86
		1000	22.31	13.93	17.80	9.71	33.45	56.49
		5000	27.71	15.06	21.82	12.32	35.54	66.85
HyperT5$_{\text{BASE}}$	277M	100	22.26	16.24	12.72	10.72	28.86	-
		500	27.42	17.10	16.35	9.99	36.92	43.67
		1000	29.05	17.43	20.84	10.68	38.07	53.51
		5000	31.78	18.71	24.33	12.80	41.75	63.47
HyperT5$_{\text{LARGE}}$	822M	100	26.40	19.17	10.36	10.00	25.36	52.54
		500	30.02	-	18.14	10.67	36.04	-
		1000	31.28	-	17.79	13.05	37.79	-
		5000	33.40	-	23.26	14.38	41.16	49.64
Data Adaptation								
DialogHT5$_{\text{SMALL}}$	97M	100	19.66	16.10	14.71	11.42	18.86	32.15
		500	26.56	18.05	18.03	12.55	31.98	54.66
		1000	28.22	18.25	19.92	12.82	33.10	53.65
		5000	30.15	18.46	22.32	12.75	34.38	65.50
DialogHT5$_{\text{BASE}}$	277M	100	28.45	20.69	15.02	11.15	-	-
		500	31.82	21.35	21.33	13.05	-	43.32
		1000	32.38	21.51	23.14	14.23	-	56.25
		5000	33.48	21.66	25.17	15.16	41.73	65.36
DialogHT5$_{\text{LARGE}}$	822M	100	29.46	21.13	13.55	4.62	-	-
		500	32.56	18.67	22.55	14.07	-	-
		1000	33.31	-	23.17	11.13	37.84	-
		5000	-	-	25.49	15.20	41.28	64.31

Table 8: Scarce data benchmark results for dialog adaptation.

Model Dataset	Params.	DI ToD	DI ODD	DR ToD	DR ODD	DS Script-Summ	DC ToD
Metrics		R1	R1	R1	R1	R1	F1
HyperT5$_{\text{SMALL}}$	97M	37.0	21.3	26.9	15.8	42.9	70.2
HyperT5$_{\text{BASE}}$	277M	38.8	23.1	29.7	16.2	44.6	70.4
HyperT5$_{\text{LARGE}}$	822M	41.5	24.5	30.2	16.5	-	71.0
Data Adaptation							
DialogHT5$_{\text{SMALL}}$	97M	37.2	21.7	27.3	15.8	42.9	70.0
DialogHT5$_{\text{BASE}}$	277M	40.1	23.6	29.8	16.4	**44.7**	**71.7**
DialogHT5$_{\text{LARGE}}$	822M	**41.8**	**25.1**	**30.5**	**16.8**	-	68.6

Table 9: Full data benchmark results for dialog adaptation.