Foundation Models of Scientific Knowledge for Chemistry: Opportunities, Challenges and Lessons Learned

Anonymous ACL submission

Abstract

Foundation models pre-trained on large corpora demonstrate significant gains across many natural language processing tasks and domains e.g., law, healthcare, education, etc. However, only limited efforts have investigated the opportunities and limitations of applying these powerful models to science and security applications. In this work we develop foundation models of scientific knowledge for chemistry to augment scientists with the advanced ability to perceive and reason at scale previously unimagined. Specifically, we build large-scale (1.47B parameter) general-purpose models for chemistry that can be effectively used to perform a wide range of in-domain and out-of-domain tasks. Evaluating these models in a zero-shot setting, we analyze the effect of model and data scaling, knowledge depth, and temporality on model performance in context of model training efficiency.

Our novel findings demonstrate that (1) unlike previous work (Kaplan et al., 2020) we show that model performance is driven by the model size rather than the data size when evaluated in a zero-shot setting; (2) data quality (aka diversity) affects model performance more than data quantity; (3) similarly, unlike previous work (Luu et al., 2021) temporal order of the documents in the corpus boosts model performance only for specific tasks, e.g., SciQ; and (4) models pre-trained from scratch perform better on in-domain tasks than those tuned from general-purpose models like Open AI's GPT-2.

1 Introduction

The emergence of foundation models (Bommasani et al., 2021) such as large-scale autoencoding models (*e.g.,* BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019)) and autoregressive language models (*e.g.,* GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), Megatron-Turing (Smith et al., 2022) and Gopher (Rae et al., 2021)) as well as multimodal vision and language models, such as FLAVA (Singh et al., 2021) and Perceiver (Jaegle et al., 2021), established a paradigm shift in Artificial Intelligence (AI). These foundation models, also called *neural platforms*, are built using self-supervised pre-training at scale. They are then able to be easily adapted to a wide range of downstream tasks via transfer learning (Bommasani et al., 2021) and fine-tuning (Lee et al., 2019).

The wide community adoption of foundation models can be explained by their key properties, two of which are *emergent behavior* and *homogenization* – which also make foundation models appealing for adaption across science and security domains. Emergence, or emergent behavior, reflect new behaviors that a model introduces or is capable of that it was not explicitly trained to perform. Homogenization is the consolidation of methods for building machine learning systems across a wide range of tasks. Another key advantage of scaling language models is that they perform competitively on language tasks using in-context learning without fine-tuning or gradient updates. Thus, in-context learning allows foundation models to be effectively used across new downstream tasks with only simple instructions and a few optional examples.

In this work we focus on a science domain (chemistry) and demonstrate the value and limitations of large-scale language models evaluated across a wide range of in-domain (science-focused) and out-of-domain tasks. Unlike the majority of work on foundation models that focuses on pre-training these models on book corpora, web pages, Wikipedia and mixed sources, *e.g.,* the Pile (Gao et al., 2020), we pretrain our models on scientific literature. Using scientific literature presents unique opportunities and challenges. Opportunities include the scale and diversity of scientific literature, the explicit structure, and explicit alignment across different modalities in the papers, *e.g.,* table and figure references. Challenges include limited benchmarks that can be used to perform model

evaluation, model prompting and interactions.

There are three major contributions of this work: *(1)* we collect and release a 0.67TB dataset covering research publication data across 10+ sources for chemistry; *(2)* we release 27 auto-regressive foundation models for chemistry that have been pretrained from scratch; and *(3)* we present a rigorous evaluation of model performance on 15+ in-domain and out-of-domain tasks that investigates the effects of model and data scaling, knowledge depth (aka diversity), and temporal order on performance as described in research questions below.

(RQ1) Science-Focused Benchmarks What are the strengths and weaknesses of foundation models pretrained on scientific literature when evaluated on out-of-domain vs. in-domain tasks?

(RQ2) Scaling Effect How does model scale affect the downstream performance? Do neural scaling laws presented in (Kaplan et al., 2020; Tay et al., 2021) hold for the foundation models for science?

(RQ3) Diversity Effect How does the depth of scientific knowledge, *e.g.,* from paper abstracts vs. full text, affect downstream performance?

(RQ4) Temporal Effect How does the recency of scientific knowledge, *e.g.,* when manipulating the temporal order of the documents processed by the model, affect downstream performance?

2 Related Work

In this section we summarize previous efforts in two categories: *mixed-domain continual pretraining* that continues pretraining of a base model on domain data and *in-domain pretraining from scratch* that pretrains a from scratch on domain data. We present a model summary in Table 1.

Mixed-Domain Continual Pretraining Many efforts have focused on continual pretraining of a BERT (Devlin et al., 2018) base model. Several models have been developed for the biomedical domain and the most frequently used corpora for domain-specific continual preraining are PubMed abstracts and PubMed Central full-text articles (PMC) (Lee et al., 2020; Peng et al., 2019; Phan et al., 2021). In the Chemistry domain, (Guo et al., 2021) performed continual pretraining of a base BERT model on 200K chemistry journal articles for product extraction (ChemBERT) and reaction role labeling (ChemRxnBERT).

In-Domain Pretraining from Scratch Previous work has shown that pretraining models from scratch on domain-specific data has a significant benefit over continual pretraining of general-domain language models (Gu et al., 2021). This is mainly due to the availability of in-domain data for both generating the vocabulary and pretraining. SciBERT (Beltagy et al., 2019) is pretrained according to this procedure using the vocabulary generated from computer science and biomedical domains. PubMedBERT (Gu et al., 2021) is another example of pretraining the base BERT model from scratch using PubMed. Unlike any previous work, we use both continual and from scratch pretraining to build the largest foundation model for Chemistry (1.47B) on the largest (0.67TB) and the most diverse corpus (10+ sources) collected to date.

3 Model Pretraining

Unlike the majority of related models that rely on a base BERT (or variant) model, we adapt the OpenAI's GPT-2 transformer decoder architecture (Radford et al., 2019) to train autoregressive language models for Chemistry. To understand the impact of model shape and size (RQ2), we experiment with four different Transformer sizes: small (S), medium (M), large (L), and extra-large (XL). These models differ in the number of decoder layers, hidden size of the model, and the number of attention heads in transformer blocks as shown in Table 2.

Our experiments leverage the GPT-NeoX Python library (Andonian et al., 2021) developed with Megatron (Shoeybi et al., 2019) and DeepSpeed (Rasley et al., 2020). We optimize the autoregressive log-likelihood (*i.e.,* cross-entropy loss) averaged over a 2048-token context. We control the micro batch size per GPU as 4, set the learning rate to 2×10^{-4}, and rely on the cosine decay. We use an Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\sigma = 10^{-8}$ and clip the gradient norm at 1.0. In addition, ZeRO optimizer (Rajbhandari et al., 2019) was used to reduce memory footprint by distributing optimizer states across several processes.

To reduce memory and increase training throughput, we use mixed-precision training (Rasley et al., 2020) and the parallel attention and feed-forward implementations available in GPT-NeoX (Black et al., 2022). We also use the Rotary positional embeddings (Su et al., 2021) instead of the learned positional embeddings used in the GPT-2 model (Radford et al., 2019) because they offer performance

Table 1: Foundation models for science focus on the biomedical, math, computer science and chemistry domains. We use † to indicate models trained for chemistry.

Model		Data Source	Pretraining	Corpus	#Params (B)
Lee et al. 2020	BioBERT	Wiki + Books	continual pretraining	PubMed	0.11
Alsentzer et al. 2019	ClinicalBERT	Wiki + Books	continual pretraining	MIMIC[1]	0.11
Peng et al. 2019	BlueBERT	Wiki + Books	continual pretraining	PubMed + MIMIC	0.11
Liu et al. 2021	MATH-BERT	Arxiv	continual pretraining	Arxiv	0.11
Guo et al. 2021	Chem(Rxn)BERT †	Wiki + Books	continual pretraining	Chemistry Journals	0.11
Phan et al. 2021	SciFive	C4	continual pretraining	PubMed	0.22 0.77
Naseem et al. 2021	BioALBERT	Wiki + Books	continual pretraining	PMC + MIMIC-II	0.02
Lewis et al. 2020	BioRoBERTa	Wiki + Books	continual pretraining	PMC + MIMIC-III	0.30
Yuan et al. 2021	KeBioLM	PubMed	continual pretraining	PubMed + UMLS[2]	0.34
Shin et al. 2020	BioMegatron	PubMed	from scratch continual pretraining	PubMed	0.80 1.20
Kanakarajan et al. 2021	BioELECTRA	PubMed	from scratch	PubMed	0.11
Miolo et al. 2021	ELECTRAMed	PubMed	from scratch	PubMed	0.11
Beltagy et al. 2019	SciBERT	PMC + CS	from scratch	PMC + CS	0.11
Liu et al. 2021	OAG-BERT	OAG	from scratch	OAG	0.11
Gu et al. 2021	PubMedBERT	PubMed	from scratch	PubMed	0.34
Our Work (autoregressive) †		10+ sources (Chemistry)	from scratch continual pretraining	10+ sources (Chemistry)	1.47

Table 2: Our model configurations: d_L is the number of decoder layers, d_{dim} is the hidden size of the model, d_{heads} is the number of attention heads. We compare model configurations between NeoX and GPT-2.

Size	Model	d_L	d_{dim}	d_{heads}	#Params (B)
S	NeoX	12	768	12	0.18
	GPT-2	12	768	12	
M	NeoX	24	1024	16	0.40
	GPT-2	24	1024	16	
L	NeoX	24	1536	16	0.80
	GPT-2	36	1280	20	
XL	NeoX	24	2048	16	1.47
	GPT-2	48	1600	25	

advantages in tasks with longer texts by capturing relative position dependency in self-attention.

Our models are pretrained across multiple workers with data parallelism. As the largest model in our experiments fit on a single GPU, we didn't use the model (tensor) or pipeline parallelism. Models are pretrained from the scratch for a total of 320K steps. The original GPT-2 models that are fine-tuned for 150K steps. We perform experiments in 2 DGX-A100 machines with 16 80Gb GPUs.

4 Data Collection and Processing

We collected a large corpus of 53.45 million chemistry-focused scientific articles and abstracts, resulting in 670GB of text data. As shown in Table 3, our corpus was collected from 10 different data sources: Arxiv, Aminer (AMiner), CORD-19 (Wang et al., 2020b), CORE (Pontika et al., 2016), Microsoft Academic Graph (MAG) (Wang et al., 2020a), OSTI, PubMed (Gao et al., 2020) (abstracts and fulltexts), and the Web of Science (WoS). See Appendix A for full data descriptions.

Table 3: Dataset statistics: combined datasets are after the de-duplication process. We split datasets to those that include abstracts ⟨A⟩ vs. full texts ⟨FT⟩.

Source	#Articles (M)	#Tokens (B)	Size (Gb)
MAG ⟨A⟩	34.26	7.43	46
Aminer ⟨A⟩	18.50	5.80	35
S2ORC ⟨A⟩	10.44	2.05	32
WoS ⟨A⟩	7.90	3.31	18
CORD-19 ⟨A⟩	< 0.01	< 0.01	0.2
OSTI ⟨A⟩	0.05	< 0.01	0.1
Arxiv ⟨A⟩	0.38	0.04	0.4
PubMed ⟨A⟩	0.28	0.08	0.5
PubMed ⟨FT⟩	0.70	7.34	32
CORE ⟨FT⟩	7.27	215.50	743
Combined ⟨A⟩	46.94	16.18	67
Combined ⟨FT⟩	6.52	184.42	603
Combined ⟨A+FT⟩	53.45	200.61	670

Because the data sources we relied on comprise research publications from many science domains, we sampled articles using a list of domain-specific keywords for chemistry to create the dataset summarized in Table 3. These keywords were extracted by using a Correlation Explanation (Gallagher et al., 2017) topic model followed by manual filtering by subject matter experts. This resulted in a list of more than 1K chemistry-related entities, ranging from compound names like *ethyl acetate*, *methyl methacrylate*, *sulfoxide*, etc. to experiment and procedures like *tunneling microscopy*, *neutralization*, *enzymatic hydrolysis*, etc.

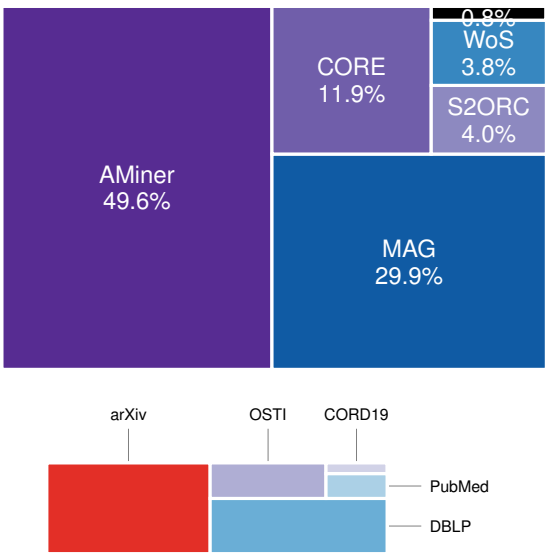


Figure 1: Summary of data source representation within the *Combined A+F* data sample. Coloring illustrates whether a data source contains *peer reviewed* (Blue), *mixed* (Purple), or *not peer reviewed* (Red) articles.

Data Cleaning Recent research has shown that duplicates in training data can significantly impact the downstream task performance of LLMs (Lee et al., 2021; Carlini et al., 2022). To this end, we performed deduplication of our corpus based on overlap of titles within and across data sources. We processed titles to strip punctuation and casefold and considered two articles A_1 and A_2 to be duplicates if they had the same processed title. With this technique, we were able to remove significant amounts of duplicate scientific articles both within and across sources. The deduplication process reduced our corpus from 875GB to 670GB (67.8M to 53.5M publications), removing 14.3M duplicates.

Tokenization As used in GPT-2 model, we use a Byte Pair Encoding (BPE) tokenizer. We train BPE tokenizers for each data sample with a vocabulary size of 64K as preliminary experiments varying vocabulary sizes from 64K to 256K for smaller scale model pretraining did not show significant differences in performance. We compare the GPT-2 vocabulary generated from the WebText and the in-domain vocabularies generated from our corpora and find that the in-domain vocabulary breaks chemical entities into fewer tokens. For example, *dimethylnitroxide* was tokenized into #dimethyl, #nitr, #oxide using the in-domain vocabulary and *#dim, #ethyl, #nit, #rox, #ide* using the GPT-2 vocabulary.

5 Analysis and Results

This section presents the analysis of 32 pretrained models evaluated on 15+ in-domain and out-of-domain downstream tasks (*RQ1*, Section 5.1). We investigate the effects of model and data scaling (*RQ2*, Section 5.2), knowledge diversity (*RQ3*, Section 5.3), and temporal order (*RQ4*, Section 5.4) on the downstream performance. We also compare the results from continual vs. from scratch pretraining (Section 5.5) and present the analysis of large-scale training efficiency (Section 5.6).

Baseline Models As we use a similar model architecture, we identify Open AI's GPT-2 (Radford et al., 2019) as a baseline comparison model. We compare our performance with four variants of the original GPT-2 models, corresponding to small (S), medium (M), large (L), and extra-large (XL) sized transformer architectures shown in Table 2. We note that GPT-2 models were pretrained on WebText – 8 million web documents (40Gb). Thus, we also include a base GPT-2 model (medium) that has been updated with continual pretraining using our *Combined* $\langle A{+}FT \rangle$ dataset.

5.1 Zero-shot Performance

We evaluate our models using several benchmarks to assess the effectiveness in both in-domain and out-of-domain tasks. The benchmarks we include are described in Appendix B. We use the lm-evaluation-harness Python repository (Gao et al., 2021) for the benchmark implementation.

In-domain Evaluation We consider five existing chemistry benchmarks, specifically Hendrycks-sTest (Hendrycks et al., 2020) for high school (HT-HC) and college (HT-CC) levels, and science-focused – ARC (Clark et al., 2018), SciQ (Welbl et al., 2017), OpenBookQA (Mihaylov et al., 2018), Pile-PubMed-Abstracts (Gao et al., 2020)). As shown in Table 4, one or more of our models outperform baseline GPT-2 models for the two chemistry tasks and the science-focused language modelling, with matched best performance on SciQ. Of the remaining tasks, our models perform within 1-4% of GPT-2 baselines.

Out-of-domain Evaluation We evaluate out-of-domain performance using 9 commonly used LLM benchmarks: BoolQ (Clark et al., 2019), CB (De Marneffe et al., 2019), WIC (Pilehvar and Camacho-Collados, 2018), WSC (Levesque et al.,

Table 4: Downstream Zero-shot In-Domain Task Performance. We use ‡ to indicate the baseline model tuned from the base GPT-2 model. Pile performance is reported using perplexity, with all other tasks reported using accuracy. We highlight the top-4 performance per task in bold, with top performance indicated with an underline.

Model	Size	HT-HC	HT-CC	ARC-E	ARC-C	SciQ	OpenBookQA	Pile
Baseline	S	0.22	0.25	0.44	0.19	0.75	0.16	96.50
	M	0.18	0.27	0.49	0.22	0.77	**0.19**	61.26
	L	0.18	0.28	**0.53**	0.22	**0.80**	**0.19**	48.86
	XL	0.18	0.26	**0.58**	**0.25**	**0.83**	**0.22**	42.29
	M‡	0.19	0.31	0.35	0.19	0.61	0.13	87.57
AMiner	S	0.18	0.27	0.43	0.21	0.70	0.17	38.40
	M	0.18	**0.34**	0.45	0.20	0.74	0.16	30.55
	L	0.23	**0.34**	0.49	**0.23**	0.78	0.18	**24.18**
	XL	0.23	**0.34**	0.50	**0.23**	0.77	0.17	**25.52**
CORE	S	0.19	0.28	0.36	0.19	0.69	0.15	78.24
	M	0.22	**0.34**	0.40	0.20	0.71	0.15	59.19
	L	0.17	0.30	0.41	0.19	0.75	0.14	52.95
	XL	0.20	0.28	0.47	0.21	0.78	0.15	39.46
MAG	S	0.24	0.28	0.41	0.20	0.66	0.17	38.03
	M	0.18	0.27	0.45	0.21	0.68	0.17	30.88
	L	0.19	**0.36**	**0.51**	**0.24**	**0.80**	0.18	**24.78**
	XL	0.20	**0.36**	0.50	0.22	**0.80**	**0.20**	26.09
PubMed-F	S	**0.26**	0.30	0.41	0.20	0.60	0.16	56.03
	M	0.19	0.27	0.43	0.21	0.68	0.18	45.69
	L	0.18	0.28	0.43	0.22	0.74	0.17	37.22
	XL	0.18	0.27	0.48	0.21	0.77	0.16	35.14
S2ORC	S	**0.26**	0.33	0.31	0.21	0.31	0.17	59.20
	M	**0.27**	0.22	0.33	0.18	0.31	0.16	45.60
	L	**0.28**	0.23	0.32	0.21	0.31	0.17	42.14
	XL	0.24	0.31	0.33	0.19	0.30	0.18	42.35
WoS	S	0.22	0.31	0.33	0.22	0.37	0.17	54.41
	M	0.25	0.32	0.32	0.20	0.34	0.16	48.31
	L	**0.27**	0.30	0.32	0.21	0.37	0.17	46.44
	XL	0.23	**0.34**	0.34	0.21	0.39	0.16	45.86
Combined-A	XL	0.17	0.28	**0.54**	**0.23**	**0.83**	0.18	**22.77**
Combined-F	XL	0.20	0.30	0.48	0.21	0.79	0.15	40.18
Combined-A+F	XL	0.18	0.30	0.48	0.22	0.79	0.17	31.03

2012), MathQA (Amini et al., 2019), PIQA (Bisk et al., 2020), PubMedQA (Jin et al., 2019), Lambada (Paperno et al., 2016) and WikiText (Merity et al., 2016). As shown in Table 5, our models outperform baseline GPT-2 models for CB, WIC and WSC and match the best accuracy for BoolQ but the GPT-2 baselines outperform on the remaining tasks, particularly Lambada and Wikitext – the two general language modeling tasks.

5.2 Scaling Effect

Previous work (Kaplan et al., 2020) has shown that upstream cross entropy loss scales as a power-law with model size, dataset size, and the amount of compute. More recently, Tay et al. (Tay et al., 2021) reported a distinct discrepancy when considering practical downstream performance along with the reported model loss. In this section, we revisit these claims on scaling Transformer architectures.

Analyzing upstream cross entropy loss During pretraining, we group each dataset into training/validation/test (949/50/1) splits. We report the model performance on validation data using cross entropy loss in nats. This measure will be averaged over the 2048-token context. We find that the cross entropy loss decreases as we increase the model size (as shown in Figure 2). Larger models reach a given loss value in a higher rate than the smaller models. This observation illustrates the relationship between model performance (as measured by the upstream cross entropy loss) and model size, confirming (Kaplan et al., 2020).

Our results also illustrate that the model shape, as measured by the number of decoder layers (depth) and hidden size (width), also contributes to the model performance (Tay et al., 2021). For example, when the hidden size increases from M to XL models (as reported in Table 2), the model performance increases (as shown in Figure 2).

Analyzing downstream task performance Can we speculate downstream task performance of a model from the pretraining performance? We find

Table 5: Downstream Out-of-domain Task Performance. We use ‡ to indicate the baseline model tuned from the base GPT-2 model. Performance on Lambada and Wikitext is reported using perplexity, all other tasks report accuracy . Top-4 performance highlighted in bold, with best performance indicated with underlines.

Model	Size	BoolQ	CB	WIC	WSC	MathQA	PIQA	PubMedQA	Lambada	Wikitext
Baseline	S	0.49	0.41	0.49	0.43	0.21	**0.63**	0.44	**40.06**	**37.37**
	M	0.59	0.43	0.50	0.40	**0.23**	**0.68**	0.53	**18.25**	**26.75**
	L	0.60	**0.45**	0.50	0.46	**0.23**	**0.70**	0.54	**12.97**	**22.61**
	XL	**0.61**	0.39	0.50	0.50	**0.24**	**0.71**	**0.59**	**10.63**	**20.38**
	M‡	**0.62**	0.34	0.50	0.36	0.20	0.55	**0.55**	2834.51	126.55
AMiner	S	0.41	0.39	0.50	0.44	0.22	0.56	0.46	2825.84	158.85
	M	0.40	0.39	**0.51**	0.41	0.21	0.57	0.43	1802.35	116.93
	L	**0.61**	**0.48**	0.50	0.47	0.22	0.58	0.36	661.81	87.23
	XL	0.50	0.39	0.50	0.37	0.21	0.58	0.43	786.22	91.28
CORE	S	**0.62**	0.41	0.50	0.37	0.20	0.55	**0.55**	671.43	100.53
	M	**0.62**	0.41	0.50	0.37	0.21	0.56	**0.55**	273.06	77.96
	L	**0.61**	0.41	0.50	0.37	0.21	0.57	0.51	173.15	69.62
	XL	**0.61**	0.38	0.50	0.37	0.22	0.58	0.45	79.95	50.47
MAG	S	0.41	0.23	0.50	0.40	0.21	0.56	0.43	1142.83	118.40
	M	0.38	0.07	0.50	0.37	0.21	0.57	0.41	628.72	91.36
	L	0.51	0.14	0.50	0.35	0.22	0.59	0.39	282.39	67.74
	XL	0.40	0.11	**0.51**	0.62	0.22	0.59	0.34	364.54	70.71
PubMed-F	S	0.58	0.41	0.50	0.45	0.21	0.57	0.54	2670.39	148.88
	M	**0.61**	0.39	0.50	0.38	0.20	0.58	0.49	1742.00	119.74
	L	0.57	0.41	0.50	0.38	0.21	0.59	0.42	843.83	95.75
	XL	0.60	0.41	0.50	0.39	0.22	0.59	0.49	679.80	90.38
S2ORC	S	0.38	0.41	0.50	**0.63**	0.20	0.57	0.34	122739.30	403.48
	M	0.38	0.43	0.50	**0.63**	0.22	0.56	0.34	80151.10	330.56
	L	0.38	**0.46**	0.50	**0.63**	0.21	0.56	0.34	89136.68	327.53
	XL	0.38	0.50	0.50	**0.63**	0.20	0.56	0.33	107065.48	351.81
WoS	S	0.38	0.39	0.50	**0.63**	0.21	0.55	0.34	140552.69	556.00
	M	0.38	**0.45**	0.50	**0.63**	0.19	0.54	0.34	182967.37	498.36
	L	0.41	0.36	0.47	0.54	0.21	0.56	0.42	148609.73	480.91
	XL	0.57	0.34	0.50	0.37	0.20	0.55	**0.56**	192970.64	509.06
Combined-A	XL	0.56	0.16	0.50	0.37	0.21	0.60	0.50	250.88	61.07
Combined-F	XL	**0.62**	0.38	0.50	0.37	0.22	0.57	**0.55**	72.50	48.96
Combined-A+F	XL	**0.61**	0.41	0.50	0.39	**0.23**	0.59	0.48	71.43	48.65

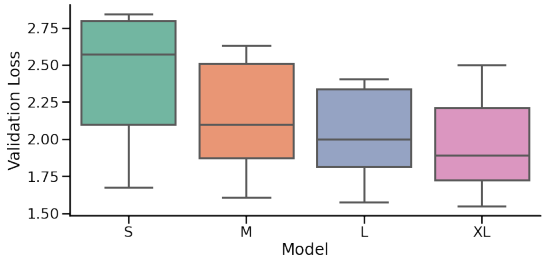


Figure 2: Distribution of validation loss by model size: performance improves as the model size increases.

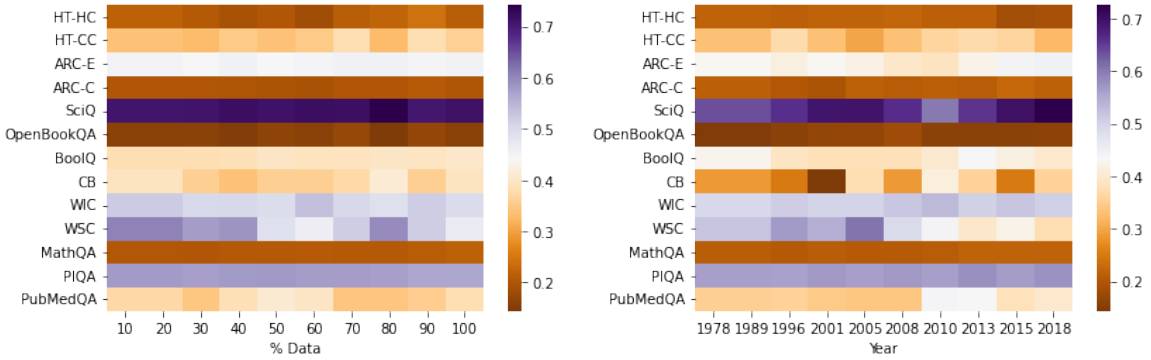
that the models perform considerably well on *Pile-PubMed-Abstract* in comparison to the *Lambada or WikiText*. This may be due to models capturing scientific language better than general language. As shown in Table 4, larger models perform well on these language modeling tasks.

Second, we find that zero-shot task performance in SciQ, HT-CC and ARC-E increases as we increase the model size (see Table 5). However, there is no clear relationship between the task performance and the model sizes in the rest of benchmark datasets. We suggest that pretraining performance may not be the ideal indicator to speculate the overall downstream task performance, especially in the zero-shot setting. However, model size significantly contributes to the task performance.

5.3 Diversity Effect

While abstracts often provide a summary of scientific publications, the full text contains more details. In this section, we analyze the performance of models trained on paper abstracts versus full texts.

First, the XL models trained with the combined abstract dataset achieve the lowest perplexity score (22.77) in the Pile language modeling task – a 45% performance advantage over the full text-based model. There are might be several factors that contribute to this, but one may be the focused language in abstracts. Second, the model trained with the combined abstracts achieves the best accuracy (0.83 in comparison to 0.79 for the full text model) in SciQ. Some of the models pretrained on individ-



(a) Random Order

(b) Temporal Order

Figure 3: The effect of temporal order of publications during pretraining. We align publications in the MAG corpus by year and split them into ten equal subsets. We repeat the process in a randomly-ordered corpus for comparison, recording model checkpoints after performing *continual pretraining* on each data subset.

ual abstract data achieve comparable performance in SciQ, *e.g.,* MAG and AMiner models achieve 0.8 and 0.78 accuracy, respectively. We believe the diversity of scientific knowledge provided from the abstract data is useful since SciQ questions span biology, chemistry, earth science, and physics.

When we compare model performance trained with abstracts vs. full texts in the Hendrycks Test (HT), we see that the best accuracy is achieved using the MAG and S2ORC datasets rather than the combined abstracts. Finally, combined full text model performs better than the model trained with the abstracts in all of the out-of-domain evaluation except PIQA. This performance difference may be due to the more expressive and diverse language presented in the full texts than in the abstracts. Thus, expanding full text coverage may improve out-of-domain task generalization.

5.4 Temporal Effect

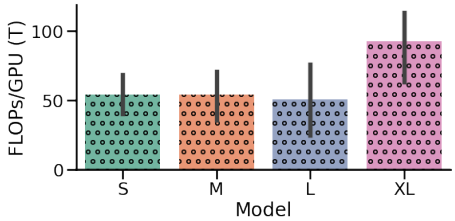
Scientific knowledge evolves over time reflecting new research ideas, innovations, and findings. In this section, we test how continual pretraining on temporal-aligned scientific publications impacts downstream performance. For this experiment, we maintain two variants of the MAG dataset with random-ordered and temporal-ordered articles, splitting each into ten equal subsets. We continue pretraining a base medium (M) sized model iteratively with the subsets in the order they appeared in the respective data variant. For example, in the temporally-aligned experiments, we first pretrain a model with 3.4M (10%) articles from before 1978, and then use it as the base model to continue pretraining with another 3.4M (10%) articles from between 1978 and 1989. We train the initial model for 150K steps and each subsequent model for 10K steps with additional data. Figure 3 shows the performance of model checkpoints across in-domain and out-of-domain tasks.

There are two key findings. First, SciQ and ARC-E zero-shot task performances improve over time with the models trained with temporally-ordered scientific texts (as shown in Figure 3b). For example, SciQ accuracy improves from 0.64 to 0.73 from the base model checkpoint to the final model checkpoint. Similarly, ARC-E accuracy improves from 0.43 to 0.45. This is due to the temporal order of the knowledge acquired by the model. When the model was pretrained with random-ordered data subsets, we observe only a slight ($< 1\%$) performance increase (as shown in Figure 3a).

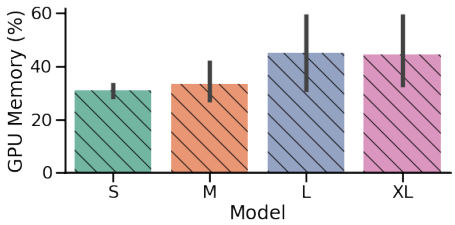
There are mixed patterns in performance across out-of-domain tasks. For example, a slight performance increase in the PIQA, CB, PubMedQA, and WIC over time with the models trained with temporally-ordered scientific texts. On the other hand, there is a performance drop in the BoolQ and WSC over time. This may be due to the *catastrophic forgetting* prevalent in continual learning (Ramasesh et al., 2021). Future work will investigate other confounding factors that may contribute to this performance patterns.

5.5 Continual vs. From Scratch Pretraining

In this section, we test whether the continual pretraining of a base GPT model with additional domain-specific data is helpful in the downstream task performance. We report the zero-shot performance of the tuned model across in-domain (Table 4) and out-of-domain (Table 5) tasks. We have two main observations from this experiment.

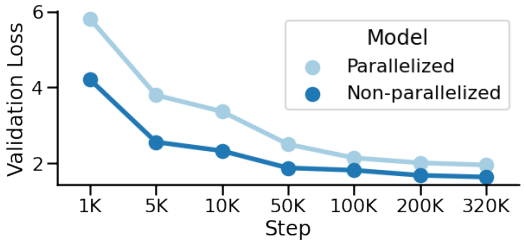


(a) GPU computation in #Floating Point Operations

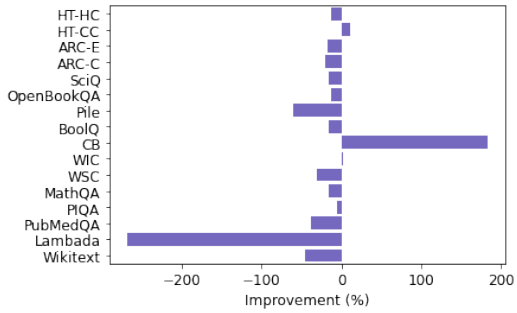


(b) GPU Memory Allocation

Figure 4: GPU system performance during pretraining.



(a) Validation loss with and without model parallelism.



(b) Downstream task performance: positive values indicate the performance advantage of the parallelized model.

Figure 5: The effect of model parallelism (MP) on validation loss and task performance.

First, fine-tuned models fall behind other baselines in a majority of in-domain tasks. HT-CC is the only in-domain task that the tuned model outperforms the rest of models, yet fails to outperform the best performing model trained from scratch.

Second, fine-tuned models have a significant performance drop in the general language modeling tasks (Lambada and Wikitext). For example, the tuned model records 6x performance drop in the Wikitext compared to the best performing model. There are several factors in the continual pretraining that may contribute to this. As the tuned model uses the original GPT-2 vocabulary, it must use the fragmented general subwords to tokenize the chemistry terms available in our corpora. On the other hand, the tuned model starts with the suboptimal initialization from the general-domain language model (Gu et al., 2021). This initialization may diverge the model in the optimization process that may not be recovered.

5.6 Training Efficiency

We use several dimensions to describe the training efficiency, *i.e.,* #FLOPs, throughput (speed), and memory. We compare these compute dimensions across the four model sizes described in the Table 2. The smallest (S) model has 59% FLOPs of the largest (XL) model, twice the speed (steps/s), 32% per device GPU memory savings, and 76% total parameter savings (see Figure 4). With such compute budget, small (S) models only outperforms the XL model in 21% in-domain and 34% out-of-domain evaluation tasks. This suggests the importance of scaling foundation models.

The largest (XL) model has 92 TFLOPs in average and use 44% GPU memory. While other larger models (M and L) have 54 TFLOPs in average, their GPU memory allocations ($> 33\%$) are quite comparable with the XL model.

Model Parallelism We introduce model parallelism (mp) to scale our models by increasing the width (hidden size) and depth (#layers) of the model architecture. To this end, we test a model architecture with 32 layers, 32 attention heads, and 2,560 hidden size resulting in 2.7B total parameters. We trained two such models with ($mp = 8$) and without ($mp = 1$) model parallelism using the PubMed full text corpus and the same hyper parameters. While model parallelism reduces overall training time, it introduces a penalty in performance. The non-parallelized model achieves the lowest validation loss value (1.64) in the pretraining while the parallelized model achieves 1.96 validation loss after 320K training steps.

6 Conclusions

In this paper, we collected and released 0.67TB of research publication data collected across 10+ sources for chemistry. We pretrained and released 25+ foundation models for chemistry. We rigorously analyzed model performance on 15+ in-domain and out-of-domain tasks.

References

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

AMiner. https://www.aminer.org/.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Shivanshu Purohit, Tri Songz, Phil Wang, and Samuel Weinbach. 2021. GPT-NeoX: Large scale autoregressive language modeling in pytorch.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Cord19. https://www.semanticscholar.org/cord19/download.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ryan J Gallagher, Kyle Reing, David Kale, and Greg Ver Steeg. 2017. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Jiang Guo, A. Santiago Ibanez-Lopez, Hanyu Gao, Victor Quach, Connor W. Coley, Klavs F. Jensen, and Regina Barzilay. 2021. Automated chemical reaction extraction from scientific literature. *Journal of Chemical Information and Modeling*, 0(0):null. PMID: 34115937.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, pages 4651–4664. PMLR.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Kamal Kanakarajan, Bhuvana Kundumani, and Malaikannan Sankarasubbu. 2021. Bioelectra: pretrained biomedical text encoder using discriminators. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 143–154.

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*.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2019. Mixout: Effective regularization to finetune large-scale pretrained language models. *arXiv preprint arXiv:1909.11299*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. KR'12, page 552–561. AAAI Press.

Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. 2020. Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Xiao Liu, Da Yin, Xingjian Zhang, Kai Su, Kan Wu, Hongxia Yang, and Jie Tang. 2021. Oag-bert: Pretrain heterogeneous entity-augmented academic language models. *arXiv preprint arXiv:2103.02410*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Michael Kinney, and Daniel S. Weld. 2020. S2orc: The semantic scholar open research corpus. In *ACL*.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A Smith. 2021. Time waits for no one! analysis and challenges of temporal misalignment. *arXiv preprint arXiv:2111.07408*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Giacomo Miolo, Giulio Mantoan, and Carlotta Orsenigo. 2021. Electramed: a new pre-trained language representation model for biomedical nlp. *arXiv preprint arXiv:2104.09585*.

Usman Naseem, Adam G Dunn, Matloob Khushi, and Jinman Kim. 2021. Benchmarking for biomedical natural language processing tasks with a domain specific albert. *arXiv preprint arXiv:2107.04374*.

OAG. https://www.microsoft.com/en-us/research/project/open-academic-graph/.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*.

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

Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2018. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*.

Nancy Pontika, Petr Knoth, Matteo Cancellieri, and Samuel Pearce. 2016. Developing infrastructure to support closer collaboration of aggregators with open repositories. *LIBER Quarterly*, 25(4):172–188.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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*.

S Rajbhandari, J Rasley, O Ruwase, and Y He. 2019. Zero: memory optimization towards training a trillion parameter models. arxiv e-prints arxiv: 11910.02054 (2019).

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2021. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Hoo-Chang Shin, Yang Zhang, Evelina Bakhturina, Raul Puri, Mostofa Patwary, Mohammad Shoeybi, and Raghav Mani. 2020. Biomegatron: Larger biomedical domain language model. *arXiv preprint arXiv:2010.06060*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2021. Flava: A foundational language and vision alignment model. *arXiv preprint arXiv:2112.04482*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2021. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020a. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020b. CORD-19: The COVID-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.

Zheng Yuan, Yijia Liu, Chuanqi Tan, Songfang Huang, and Fei Huang. 2021. Improving biomedical pretrained language models with knowledge. *arXiv preprint arXiv:2104.10344*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Data Descriptions

AMiner ArnetMiner (AMiner) is a service that crawls research publications, performs profile extraction of scientists, models academic networks by integrating publication data from the existing libraries. For the experiments described in this work, we use a sub-sampled version of the data presented in the Open Academic Graph (OAG) version of the AMiner dataset, which originally consisted of more that 172M articles, with 18.5M chemistry-related abstracts.

CORE COnnecting REpositories (CORE) (Pontika et al., 2016) is a large-scale aggregation system which provides an open access to the global network of scientific journals and publications. CORE currently contains more than 207M open-access articles collected from over 10 thousand data providers, out of which more than 92M are open access full-text research papers. We sub-sampled the original collect into our chemistry-specific corpus consisting of more than 7M full-text articles.

CORD-19 CORD-19 corpus contains COVID-19 (Cord19) and other coronavirus-related publications (e.g. SARS, MERS, etc.) from PubMed's PMC open access corpus, bioRxiv, and medRxiv pre-prints, in addition to COVID-19 articles maintained by the World Health Organization (WHO).

MAG Microsoft Academic Graph (MAG) is a heterogeneous graph created by extracting knowledge from scholarly publications on the web (Wang et al., 2020a). The data used in this work is a subsample from the OAG version of the MAG dataset, which originally consisted of $>$ 208M articles, with 34M chemistry-related articles with abstracts.

PubMed PubMed is a domain-specific data source that allows for search and retrieval of the biomedical and life sciences literature. It is maintained by the National Centre for Biotechnology Information (NCBI) at the U.S. National Library of Medicine (NLM). For this work we utilized the PubMed Central data provided in the Pile corpus (Gao et al., 2020). As presented in Table 3 the sub-sampled data consists of documents with more than 280K abstracts and 700K full text articles.

S2ORC The Semantic Scholar Open Research Corpus (S2ORC) (Lo et al., 2020) is a large academic corpus consisting of 81.1M documents. The data includes the metadata, abstracts, bibliographical references and full-text publications for over 8M open access research articles. In this work, we utilize the sub-sampled version of the original data specific to chemistry, which includes more than 10M abstracts.

WoS The Web of Science (WoS) is a multi-discipline citation database produced by the Institute of Scientific Information. The platform hosts over 171M records across various disciplines, which, when sub-sampled for our chemistry domain, rounded to more than 7M records with abstracts available.

B Task Descriptions

HendrycksTest-Chemistry The Hendrycks Test (Hendrycks et al., 2020) is a large scale collection of multiple choice questions covering 57 subjects. In our experiments, we subsampled college chemistry (HT-CC) and high school chemistry (HT-HC). HT-CC contains 100 questions related to analytical, organic, inorganic, physical, etc. and HT-HC contains 203 questions related chemical reactions, ions, acids and bases, etc.

ARC The ARC dataset (Clark et al., 2018) contains 7,787 genuine grade-school level, science MCQs and is partitioned into a Challenge Set (ARC-C) and an Easy Set (ARC-E). Additionally, 14M science-related sentences are provided with relevant knowledge to answer the ARC questions.

SciQ The SciQ dataset (Welbl et al., 2017) contains 13,679 crowdsourced multiple-choice science exam questions about Physics, Chemistry and Biology, among others.

OpenBookQA The OpenBookQA (Mihaylov et al., 2018) dataset consists of 5,957 multiple choice questions and 1,326 elementary-level science facts. The facts alone do not contain enough information to correctly answer the multiple choice questions, therefore the task is designed to evaluate systems beyond paraphrase matching.

Pile PubMed Abstracts The Pile dataset (Gao et al., 2020) contains 800GB of diverse text sources for benchmarking language models. We limit this task to only include abstracts from the Pile's PubMed collection. As this is framed as a language modeling task, we report word level perplexity.

BoolQ BoolQ (Clark et al., 2019) is a reading comprehension dataset comprised of 16k real, naturally formed queries to the Google search engine

with a yes or no answer. Each question-answer pair is accompanied by a Wikipedia article providing evidence to support the correct answer.

CB Commitment Bank (CB) (De Marneffe et al., 2019) is a 3-way classification of textual entailment (true, false, unknown) from 1,200 short text segments where at least one sentence contains an embedded clause. The dataset contains passages from three sources: the Wall Street Journal, the British National Corpus, and Switchboard.

WIC The Word-in-Context dataset (WIC) (Pilehvar and Camacho-Collados, 2018) is a benchmark for evaluating context-sensitive word embeddings. The task is to classify if a target word has the same meaning in two context sentence.

WSC The Winograd Schema Challenge (WSC) (Levesque et al., 2012) dataset is a collection of 804 sentences in which the task is to resolve coreferences.

MathQA MathQA (Amini et al., 2019) is a dataset containing 37k multiple choice math word problems built from the existing dataset, AQuA (Ling et al., 2017).

PIQA The Physical Interactions: Question Answering (PIQA) (Bisk et al., 2020) benchmark dataset provides 21k questions about the physical world and plausible interactions encountered by humans. Annotators provided correct and incorrect answers to questions extracted from instructables.com, a website of instructions for completing many everyday tasks.

PubMedQA The PubMedQA dataset (Jin et al., 2019) is a collection of 273.5k biomedical research questions and related PubMed articles with yes/no/maybe answers.

Lambada Lambada (Paperno et al., 2016) contains passages and target sentences from 5,325 novels collected from Book Corpus (Zhu et al., 2015), and the goal is to predict the last word of the target sentence given the context passage. This task was designed to test genuine language understanding since accurate prediction of the final word would be improbable without the context passage.

WikiText The Wikitext benchmark (Merity et al., 2016) is a language modeling dataset of 29k articles from Wikipedia. Only articles classified as *Good* or *Featured* by Wikipedia editors are included since they are considered to be well written and neutral in language. All results are reported on Wikitext-2.