Pacific Northwest
NATIONAL LABORATORY

PNNL-37221

Mega AI: Scaling AI for Science and Security

December 2024

Maria F Glenski
Robin J Cosbey
Shivam Sharma
Megha Subramanian
Anurag Acharya
Ellyn M Ayton

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-37221

Mega AI: Scaling AI for Science and Security

December 2024

Maria F Glenski
Robin J Cosbey
Shivam Sharma
Megha Subramanian
Anurag Acharya
Ellyn M Ayton

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

State-of-the-art, large-scale language models or multimodal foundation models incorporating a text modality are trained on large collections of pretraining data, largely focusing on general-purpose language and/or vision data sources. However, performance often degrades when applying foundation models trained on general-purpose datasets to science and security domains, such as handling the vocabulary shift between general language versus domain knowledge in areas like molecular chemistry and climate. By leveraging a large collection of scientific literature, the Mega AI project focused on developing next-generation foundation models addressing science and security missions. The project explored the tradeoffs of development choices (pretraining from scratch, fine-tuning off-the-shelf base models, and targeted fine-tuning and/or task-prompts) and model performance to support: on premise model use, mission informed training/tuning of usable LLMs, & traceable model development and evaluation.

Acknowledgments

This research was supported by the **Mathematics for Artificial Reasoning in Science (MARS)** Initiative under the Laboratory Directed Research and Development (LDRD) Program at Pacific Northwest National Laboratory (PNNL). PNNL is a multi-program national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL01830.

Contents

Figures

Tables

1.0 Introduction

Foundation models pre-trained on large corpora demonstrate significant gains across many natural language processing tasks and domains e.g., law, healthcare, education, etc. Unlike discriminant (or “narrow”) AI models, a massive-scale foundation model is a single model that learns from huge amounts of raw unlabeled data and is multi-purpose. Therefore, it can be rapidly adapted to a wide range of useful tasks, such as knowledge summarization, information extraction, hypothesis generation and validation, question answering, classification, recommendation etc. Limited efforts had explored the opportunities and limitations of applying these models to science and security applications when the Mega AI project began, motivating the design of experiments exploring the development of foundation models for these missions and the cost-performance tradeoffs of pre-training from scratch, tuning, and usability across a range of downstream tasks.

Over the course of the project, there were three use cases: Molecular Chemistry, Climate Security, and Cybersecurity. The focus by fiscal year is shown in Table 1. The project's objectives were to develop foundation models tailored to each use case, evaluate these models alongside open-source baseline models, and evaluate their performance on both in-domain and out-of-domain benchmarks and downstream tasks. Evaluations focus not only on performance assessments but contextualization of strengths (performance improvements, robustness) and weaknesses (challenges or limitations of downstream use) and the development choices (pre-training from scratch vs. tuning of open-source base models, parameter choices, or length of training used, etc.).

Table 1. Project focus by fiscal year on each of the three use cases.

Use Case	FY22	FY23	FY24
Molecular Chemistry			
Climate Security			
Cybersecurity			

Overall, the project aimed to develop large-scale, multi-purpose foundation AI models to enable generative solutions for tasks within science and security mission domains. For Climate in FY22 and Chemistry in FY22 and FY23, objectives included pretraining AI models with over 1 billion parameters using large-scale scientific text datasets curated by the project and molecular database information. In FY24, the focus shifted to Cybersecurity and Code, where the goal was focused around targeted fine-tuning and adaptation of these models to enhance performance in multi-purpose tasks, from zero-shot to instruction-tuned applications. These tasks were focused on enabling efficient on-premises, mission-informed tasks like vulnerability assessment at scale.

2.0 Scaling AI for Science Missions

In the first two years, the project developed foundation models of scientific knowledge for chemistry (FY22 and FY23) and climate (FY22) to augment scientists. Specifically, we built large-scale (1.47B parameter) general-purpose models that could 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 analyzed the effect of model and data scaling, knowledge depth, and temporality on model performance in context of model training efficiency.

2.1 Pretraining Datasets

State-of-the-art large-scale language models or foundation models are trained on large corpora of data. The project team constructed several large pre-training datasets (ranging from 21.4M to 294.8M documents) constructed from scientific literature to support science and security targeted development of foundation models. These datasets were collected to support domain-specific pretraining at scale, enabling domain-focused models via pretraining from scratch or continual pretraining of other SOTA models trained on general text.

We leveraged scientific literature from a variety of data sources when constructing each of our datasets. When constructing our five datasets, we sampled domain-focused scientific literature from nine data sources. This included sampling from six existing academic literature datasets: the Semantic Scholar Open Research Corpus (S2ORC), ArnetMiner's "AMiner" dataset, the Microsoft Academic Graph (MAG), PubMed publications from the Pile, the COnnecting REpositories (CORE) dataset, and the CORD-19 dataset. We also used API-based sampling from three publication databases or scholarly search systems: the dblp computer science bibliography (DBLP), Clarivate's Web of Science (WoS), and the Office of Scientific and Technical Information's OSTI.gov engine (OSTI). Data sources also included two pre-print archive and distribution services: arXiv and bioRxiv.

We used a combination of source-based, venue-based, and keyword-based collection strategies to construct our two domain-focused samples for Chemistry and Climate, and a third General Science sample. The General Science sample was created by aggregating documents from across the data sources. Table 2 illustrates the use of each strategy across datasets.

Table 2. Summary of Collection Strategies Used by Sample.

Dataset	Data Source	Venue-Based	Keyword-Based
Chemistry		✓	✓
Climate			✓
General Science	✓		

Our chemistry dataset was collected through a keyword-based collection strategy using a collection of keywords extracted by using a Correlation Explanation topic model followed by manual filtering by subject matter experts. This resulted in a list of more than 1300 chemistry-related entities, ranging from compound names like ethyl acetate, methyl methacrylate, sulfoxide, etc., to experiments and procedures like tunneling microscopy, neutralization, enzymatic hydrolysis, etc., which is included in the supplementary materials.

Similarly to the chemistry collection process, our climate dataset was collected using a keyword-collection strategy with a collection of 56 subject matter expert-informed keywords including (but not limited to) aerosol effect, anthropogenic, atmospheric model, biogeochemistry, carbon assimilation, climate change, climate security, climate sensitivity, cloud albedo, earth system model, fossil fuels, greenhouse gases, and mesoscale convective system.

2.1.1 Deduplication

Prior research (Katherine Lee 2022) has demonstrated that duplicates in training data can significantly impact the performance of models. To address this, we performed deduplication on each of our three datasets. To detect and remove duplicates, we compared the titles of scientific articles across all data sources. We standardized the titles by converting them to lowercase (i.e. case folding) and removing punctuation, creating simplified versions. Two articles were considered duplicates if they had the same processed title. Deduplication resulted in 107.9M Chemistry documents, 21.4M Climate documents, and 294.8M for General Science.

When removing duplicates from each dataset, we followed a prioritization strategy:

1. We included sources that contained only or primarily peer-reviewed publications.
2. Sources that contained both peer-reviewed and non-peer-reviewed publications.
3. Sources that did not have peer-review requirements (e.g., arXiv, bioRxiv).

It is important to note that sources like arXiv and bioRxiv, which do not require peer review, may still include peer-reviewed research. This was evident during the deduplication process for the Climate dataset, where arXiv and bioRxiv exhibited high overlap with literature from sources such as WoS, OSTI, and DBLP. The relative scale of each source is illustrated in Figure 1 for Chemistry, Figure 2 for Climate, and Figure 3 for General Science. Color encodes whether the source contains Peer Reviewed (Blue), Mixed (Purple), or Not Reviewed (Red) publications and shades distinguish sources, used consistently across all figures.

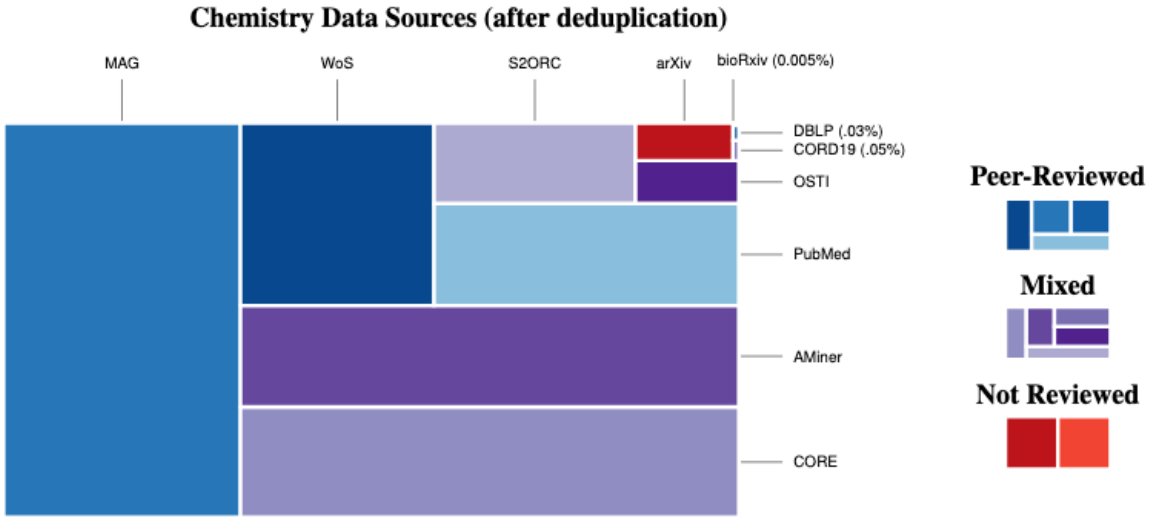

Figure 1. Overview of the Chemistry dataset where size encodes the relative scale of each source within the dataset.

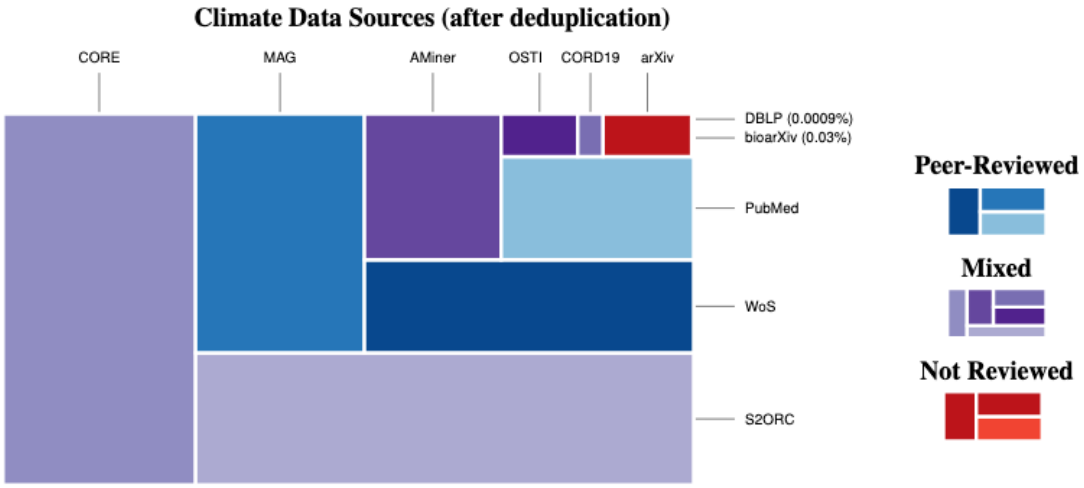

Figure 2. Climate dataset sources, size encodes the relative scale of each source.

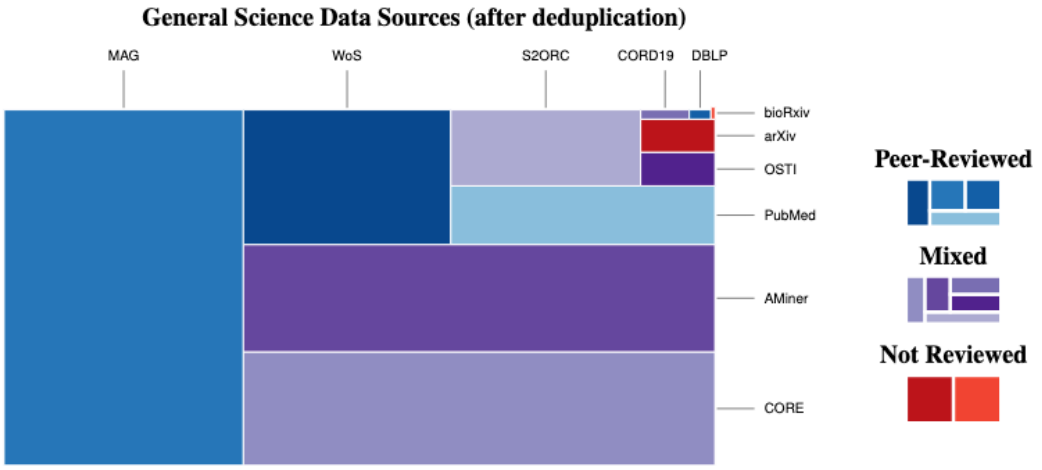

Figure 3. General Science dataset sources, size encodes the relative scale of each source.

2.1.2 Tokenization

We use the Byte Pair Encoding (BPE) algorithm (Shibata 1999) to train a tokenizer with a vocabulary size of 64K for each source, using the BPE implementation available in the Megatron (Shoeybi 2019) Python repository. Due to the massive scale of the general science corpus, we only sampled the first 1M characters in each text as the input to this training process, observing that the mean character count for abstracts (1,206) and full text (43,560) fall well below this threshold. Using a DGX A100 machine with 512Gb memory to train this tokenizer took ˜4.25 hours. We used the resulting vocabularies to tokenize each sample (General Science, Chemistry, Climate). This resulted in 200.6B tokens for Chemistry, 65.7B tokens for Climate, and 451.9B tokens for General Science.

These in-domain vocabularies provide meaningful tokens that are useful in the training (i.e., pre-training from scratch or continual pre-training) of large-scale foundation models. For example, dimethylnitroxide was tokenized into #dimethyl, #nitr, #oxide using our BPE-tokenized in-domain vocabulary and into #dim, #ethyl, #nit, #rox, #ide using the standard GPT-2 vocabulary.

2.2 Evaluation Datasets

We evaluate out-of-domain performance across science use cases using 9 commonly used LLM benchmarks: BoolQ, Commitment Bank, MathQA, PIQA, PubMedQA, WIC, WSC, Lambada, and WikiText.

BoolQ (Christopher Clark 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

Commitment Bank (CB) (Marie-Catherine De Marneffe 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.

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

Physical Interactions: Question Answering (PIQA) (Yonatan Bisk 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 dataset (Qiao Jin 2019) is a collection of 273.5k biomedical research questions and related PubMed articles with yes/no/maybe answers.

Word-in-Context dataset (WIC) (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.

Winograd Schema Challenge (WSC) (Hector J. Levesque 2012) dataset is a collection of 804 sentences in which the task is to resolve coreferences.

Lambada (Denis Paperno 2016) contains passages and target sentences from 5,325 novels collected from Book Corpus (Yukun Zhu 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 (Wikitext-2) The Wikitext benchmark (Stephen Merity 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.

2.2.1 Chemistry-Specific Evaluation Datasets

There are five in-domain chemistry benchmarks that were used throughout the efforts of Chemistry-focused modelling:

- **HendrycksTest-Chemistry** The Hendrycks Test (Dan Hendrycks 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** (Peter Clark 2018) contains 7,787 genuine grade-school level, science multiple-choice questions 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 (Johannes Welbl 2017) contains 13,679 crowdsourced multiple-choice science exam questions about Physics, Chemistry and Biology, among others.

- **OpenBookQA** The OpenBookQA (Todor Mihaylov 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 (Leo Gao 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 reported word level perplexity.

In addition, we introduced five new tasks for evaluation in the second year (described in Section 3.3) that included evaluations based on CHEMDNER (Krallinger 2015) and PubChem (Kim 2019) datasets.

- **CHEMDNER** tasks include Chemical Entity Extraction (CEE) and Chemical Entity Recognition (CER). Each text in the CHEMDNER datasets contained one or more chemical named entities from one of seven classes of chemical entities (Trivial, Family, Systematic, Formula, Abbreviation, Multiple, Identifier). For the CEE task, a model has to identify all entities present in the text for a specific entity class. For the CER task, a model has to identify all entity classes for the entities present in the provided text. We developed custom metrics for the CEE and CER tasks based on the methodology described in (Chinchor 1993) which allowed us to consider partial matches.

- **PubChem Molecular Property Tasks** We derived three new tasks using the molecular properties reflected in the PubChem (Kim 2019) database: Molecular Formula Generation (MFG), Isomeric SELFIE String Generation (ISG), and Molecular Weight Estimation (MWE). Each task is structured such that given an IUPAC name, a model is asked to generate the respective property (molecular formula (MFG), SELFIE representation of the molecule (ISG), and an estimated value for the molecular weight (MWE).

2.2.2 Climate-Specific Evaluation Datasets

The out of domain evaluation for climate used the same set as the project used for chemistry (BoolQ, CB, MathQA, PiQA, SciQ, WIC, WSC, LaMBADA, and WikiText). There were three climate-focused in domain evaluation datasets. Two of which were collected from prior research

or open-sourced work within the community (Climate-FEVER and SciDCC) and a third constructed by the project team (cleanetQA).

- **Climate-FEVER** (Diggelmann et al., 2020) was a claim verification benchmark comprising 1,535 climate-focused claims with five annotated evidence sentences per claim, individually annotated as supports, refutes, or not enough information. Claims themselves were labeled similarly, with the addition of "disputed" for claims with both supporting and refuting evidence.

- **SciDCC** (Mishra and Mittal) was a 20-class text classification dataset comprising 11,539 climate change-related news articles that were collected from the *Earth Climate* (http://www.sciencedaily.com/news/earth_climate/) and *Plant Animals* (http://www.sciencedaily.com/news/plants_animals/) topics in the environmental science section of the Science Daily website.

  SciDCC classes comprised: Earthquakes, Pollution, Genetically Modified, Hurricanes/Cyclones, Agriculture Food, Animals, Weather, Endangered Animals, Climate, Ozone Holes, Biology, New Species, Environment, Biotechnology, Geography, Microbes, Extinction, Zoology, Geology, Global Warming.

- **cleanetQA** was a climate/energy literacy question-answering dataset that the Mega AI project team constructed from the climate literacy and energy literacy quizzes from cleanet.org: https://cleanet.org/clean/literacy/climate/quiz.html and https://cleanet.org/clean/literacy/energyquiz.html.

2.3 GPT-NeoX-based Chemistry Models

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, Lo and Cohan 2019) was pretrained 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. In our FY22 experiments, we used 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 at the time.

We adapted Open-AI's GPT-2 transformer decoder architecture (Radford, et al. 2019) to train our autoregressive language models for Chemistry. To understand the impact of model size, 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 4. Our experiments leveraged the GPT-NeoX Python library (Andonian, et al. 2021) developed with Megatron (Shoeybi, et al. 2019) and DeepSpeed (Rasley, et al. 2020). We optimized the autoregressive log-likelihood (i.e., cross-entropy loss) averaged over a 2048-token context; set the micro batch size per GPU as 4, the learning rate to 2 x 10^{-4}; relied on the cosine decay; used 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.

Table 3: Our model configurations, comparing across GPT-NeoX and GPT-2.

Size	Model	# of decoder layers	hidden size	# of attention heads	#Params (B)
S	GPT-NeoX	12	768	12	0.18
	GPT-2	12	768	12	
M	GPT-NeoX	24	1024	16	0.40
	GPT-2	24	1024	16	
L	GPT-NeoX	24	1536	16	0.80
	GPT-2	36	1280	20	
XL	GPT-NeoX	24	2048	16	1.47
	GPT-2	48	1600	25	

To reduce memory and increase training throughput, we used 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 used 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 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 were pretrained from scratch for a total of 320K steps. The original GPT-2 models are fine-tuned for 150K steps.

We pretrained models with individual datasets (AMiner, CORE, MAG, PubMed, S2ORC, WOS) and combined abstracts and full texts. Goals of these experiments were to systematically study data biases in the model performance when pretraining models with individual datasets. We used 4 GPUs for the models pretrained with individual datasets and 8 GPUs for the combined models. This is to control the number of tokens seen during model pretraining (320,000 steps * 4 GPUs * 4 micro batch size * 2,048 context size = 10B tokens) relative to the maximum number of tokens available in the respective datasets which varied in scale. We also trained one XL (4x) model with 4x larger batch size than what used in XL model to evaluate the impact of the number of training tokens.

2.3.1 Key Highlights

Novel findings using the GPT-NeoX models developed in FY22 demonstrated that (1) model size significantly contributes to the task performance 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 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.

Our evaluations across the five in-domain chemistry benchmarks show that one or more configuration of our models outperforms baseline GPT-2 models in two chemistry tasks, general science QA (SciQ), and science-focused generative text language modeling. For the remaining tasks, such as ARC and OpenBookQA, our models perform within 1-4% of the GPT-2 baselines. Evaluations on the out-of-domain benchmarks commonly used for LLM evaluations illustrate that our chemistry-domain trained models outperform baseline GPT-2 performance for CB,WIC,and WSC and match best accuracy for BoolQ but that for the remaining tasks particularly Lambada and WikiText – the two general language modeling tasks – baseline GPT-2 models have stronger performance.

Temporal Training Analysis. Scientific knowledge evolves over time reflecting new research ideas, innovations, and findings. We tested how continual pretraining on temporal-aligned scientific publications impacts downstream performance. For these experiments, we maintained 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.

There were two key findings. First, SciQ and ARC-E zero-shot task performances improve over time with the models trained with temporally ordered scientific texts. For example, SciQ accuracy improved 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. When the model was pretrained with random-ordered data subsets, we observe only a slight (<1%) performance increase. However, there were 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, Lewkowycz and Dyer 2021).

2.4 MOLJET

A collaboration with the University of Washington (UW) in FY22 focused on addressing key challenges in the domain of multi-property constrained optimization of molecules through generative de novo design models. Recognizing the gap between the reported performance in literature and practical utility in real-world scenarios, and the inaccessibility of such models to chemists without a computer science background, we developed and assessed a generative foundation model named the Multimodal Joint Embedding Transformer (MOLJET).

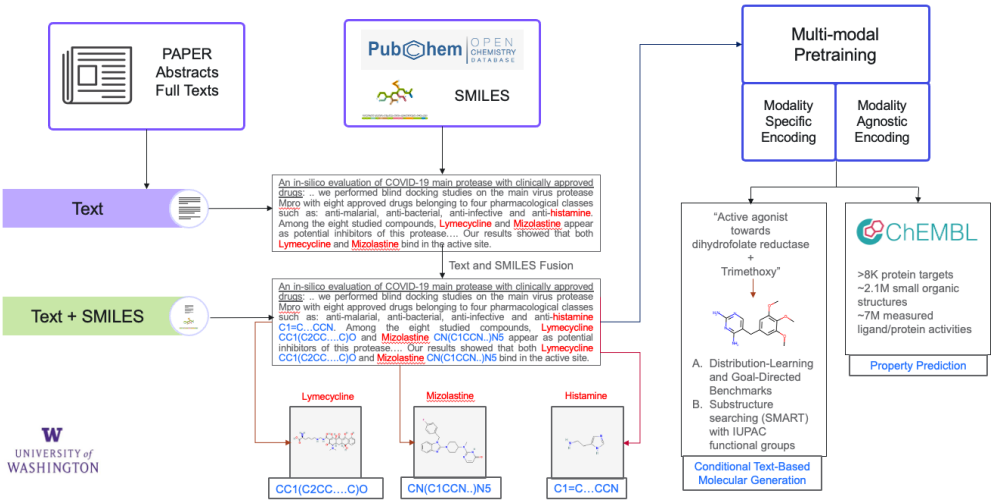

Figure 4. Outline of Multimodal Text+SMILES model development completed in collaboration with University of Washington, under a subcontract.

The MOLJET development focused on conditional generation of desired molecular distributions based on text-based chemistry prompts in a zero-shot setting. Evaluation leveraged standard benchmarks from the GuacaMol and MIMOSA frameworks, including structure-based sampling tasks and multi-property optimization tasks (designing drug-like molecules under realistic property constraints). Experiment results demonstrated that with self-supervised pretraining, MOLJET outperformed 80% of task-optimized models using zero-shot inferences and was able to surpass all baselines after minimal supervision.

Benchmark Category	Best of Data Set	SMILES LSTM	SMILES GA	Graph GA	MOLJET-GUAC (Zero-shot)	MOLJET-GUAC + Graph GA
MPOs	0.698	0.778	0.717	0.868	0.838	**0.878**
Rediscovery	0.613	**1.000**	0.523	0.945	**1.000**	**1.000**
Similarity	0.546	**1.000**	0.771	0.977	**1.000**	**1.000**
Substructure	0.643	0.973	0.769	**0.985**	0.817	**0.985**
Isomers	0.716	0.912	0.745	0.954	**1.000**	**1.000**
Median	0.371	0.403	0.362	0.417	0.409	**0.447**
Total	0.623	0.850	0.671	0.877	0.857	**0.900**

Figure 5. Summary of MOLJET model performance.

2.5 AISLE Chemistry Models

In our FY23 experiments we compare two core model architectures using their HuggingFace (Thomas Wolf 2020)[1] implementations or architectures: the Generative Pre-trained Transformer Model (GPT-2) ((Alec Radford 2019) and the BigScience Large Open-science Open-access Multilingual Language Model (BLOOM) (Teven Le Scao 2022). We collected and/or trained the following configurations for each model architecture:

[1] https://huggingface.co/

- the off-the-shelf baseline model,
- an AISLE model pre-trained from scratch using our Chemistry Pretraining dataset,
- a baseline model with instruction fine-tuning,
- and an AISLE pre-trained from scratch model with instruction fine-tuning.

Baselines. We used the GPT2-XL model with 1.5B parameters and the BLOOM-3B model with 3B parameters. We used the pre-trained weights and standard GPT-2 or BLOOM tokenizers available from HuggingFace (Thomas Wolf 2020).

AISLE Models Trained from Scratch. We leverage our aggregated scientific data to train seven models from scratch across the two architectures (GPT and BLOOM). We trained these models for three epochs each over 10B tokens from 53 million scientific documents. All of these ***AI*** from ***S***cientific ***L***iterature AISLE models were pre-trained with a 95/5 train/validation split.

Instruction Fine-tuned Models. To better adapt models for the chemistry domain, we perform instruction fine-tuning across a variety of tasks leveraging molecular database features (formula, SELFIE string, molecular weight) and chemical entity extraction/recognition using CHEMDNER data. Prompt templates are shown in Figure 5. Fine-tuning for two epochs with early stopping, we conduct experiments with both the off-the-shelf baseline models and our domain pre-trained AISLE models of each architecture and fine-tune on a combination of the training data for all five tasks we consider resulting in four fine-tuned models: Baseline GPT2 , Baseline BLOOM, $AISLE_{GPT2}$, and $AISLE_{BLOOM}$.

Task	Instruction Template
CEE	Below is an instruction that describes a task. Write a response that appropriately completes the request. ### Instruction: Identify all <*ENTITY TYPE*> entities in the given text as written. ### Text: <*INPUT TEXT*> ### Response: <*LIST OF ENTITIES*>
CER	Below is an instruction that describes a task. Write a response that appropriately completes the request. ### Instruction: What are the types of entities in the given text? ### Text: <*INPUT TEXT*> ### Response: <*LIST OF ENTITY CLASSES*>
MFG	Below is an instruction that describes a task. Write a response that appropriately completes the request. ### Instruction: Give the molecular formula for <*IUPAC NAME*>. ### Response: <*MOLECULAR FORMULA*>
ISG	Below is an instruction that describes a task. Write a response that appropriately completes the request. ### Instruction: Give the SELFIE string for <*IUPAC NAME*>. ### Response: <*SELFIE STRING*>
MWE	Below is an instruction that describes a task. Write a response that appropriately completes the request. ### Instruction: Give the molecular weight for <*IUPAC NAME*>. ### Response: <*MOLECULAR WEIGHT*>

Figure 6. Prompts for the CHEMDNER and PubChem instruction tasks. Italicized clauses in angle brackets are replaced for each example.

2.5.1 Key Highlights

Experiments using the AISLE models and comparisons to both GPT-2 (general language) and BLOOM (general science) identified several key findings. Our results show that not only do in-domain base models perform reasonably well on in-domain tasks in a zero-shot setting but that

further adaptation using instruction fine-tuning yields impressive performance on chemistry-specific tasks such as named entity recognition and molecular formula generation.

When we evaluate the model's ability to answer chemistry exam questions at the high school (HT-HC) and college (HT-CC) level, using the chemistry tasks from the MMLU benchmark, we see that our domain-pretrained AISLE models outperform baseline models, as shown in Figure 6 that illustrates the percent improvement by our model over three baselines on high-school (HC) and college (CC) level questions, showing consistent strong performance by the AISLE GPT model. In particular on the college-level task where there were improvements of 20%+.

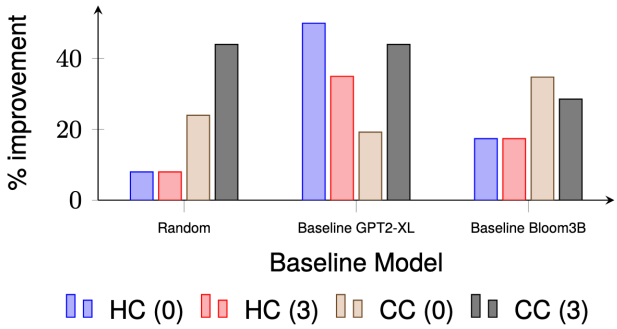

Figure 7. AISLE GPT2-XL Improvement over Baselines for Chemistry Exam Questions. Relative improvement in accuracy, ranging from 10%-50%, achieved by the AISLE GPT 2 model over baselines for zero-shot (0) and few-shot using 3 examples (3).

Expanding beyond chemistry alone, as shown in Figure 7, we find that a domain-pretrained AISLE model outperforms consistently in Chemistry-adjacent topics (ChemBioMed, Health, STEM). Interestingly, we see some higher performance compared to baselines in the Social Science (Social Sci.) focused topic. When we evaluate the models across the MMLU benchmark overall, comparing average performance across all tasks regardless of topic, we also see that our strongest AISLE model outperformed both baseline models by a small margin.

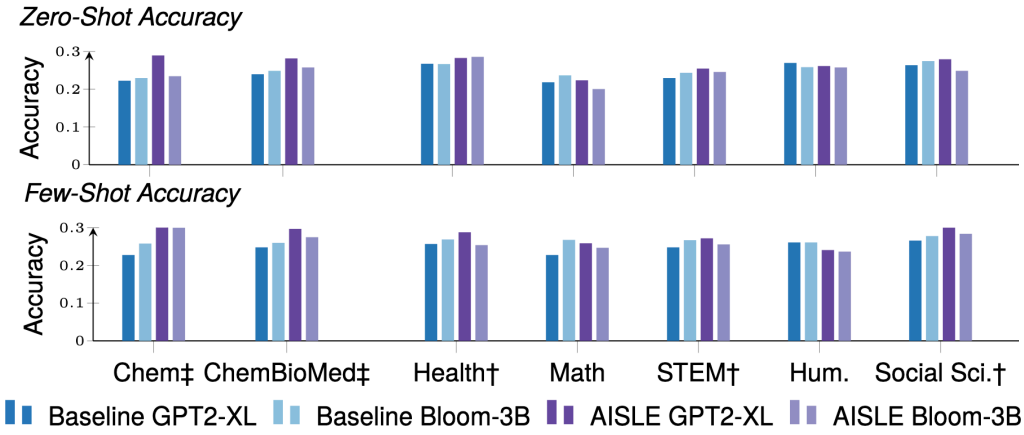

Figure 8. Model performance (accuracy) in zero- and few-shot evaluations across subsets of the MMLU benchmark. We use a ‡ to denote if both AISLE models († if one) outperform baselines.

2.6 Climate Models

The project's second science use case (FY22) was focused on climate security. Climate security encompasses the physical, economic, or societal changes associated with climate

environment changes. Given the influence of scale and emergent behavior on foundation model performance, we chose to explore the impacts of data source, pretraining scale, and pretraining objective on the ability of foundation models to perform on climate-related knowledge and reasoning tasks. Specifically, we set out to answer the following research questions:

- To what degree does fine-tuning general-language models on climate data change performance compared to climate data pretraining?
- How does downstream climate and general language performance change for the above conditions as a model's number of tokens seen during training increases?

We explored six different transformer networks, namely three each based on the popular autoregressive GPT-2 (as used in Chemistry models) and RoBERTa (Yinhan Liu 2019). GPT-2 is pretrained using an autoregressive generation task, whereas RoBERTa is pretrained using a dynamic masked language modeling (MLM) task. For each model variant, we explored three settings: evaluating an open-source baseline for each model directly on our downstream climate and general language benchmarks, fine-tuning the baseline model on our climate-specific corpus, and pretraining entirely on our climate corpus from a random initialization.

GPT-2 models were trained using the "gpt2-xl" (1.5B parameters) configuration, while RoBERTa models use the "roberta-base" (150M parameters). Due to the number of parameters in the GPT-2 models, we leverage the highly efficient and scalable GPT-NeoX python library when training or finetuning them. All RoBERTa models were trained using the huggingface library. All models were pretrained (if trained from scratch) or finetuned (if leveraging pretrained weights) on the climate corpus for 125,389 iterations with a batch size of 512. Since the models used a context window of size 1024, this is equivalent to a single full pass through the 65.74B-token training set. All models were trained using 8 40Gb GPUs from a single A100 machine.

The scale of the climate dataset for pretraining was much smaller than the volume of data used in our chemistry experiments. We found that this has a significant impact on the strength of model performance in both fine-tuned and pretraining from scratch settings. Table ?? highlights the pretrained “from scratch” and fine-tuned checkpoints that we trained using our climate corpus compared to an off-the-shelf baseline and illustrates that the domain-trained from scratch model was able to outperform on two of the three in-domain tasks but the fine-tuning alone actually showed worse performance than the off-the-shelf baseline. Findings of the significant impact of increasing scale and vocabulary size on model performance that was seen in our chemistry results underscore that there would be a need for much larger climate focused resources to realize the full benefits of pretraining from scratch.

Table 4. In-Domain Zero-Shot performance using Macro F1.
Strongest performance for each benchmark is indicated in bold.

Model	CleanNetQA	cFEVER	SciDCC
From scratch GPT-XL	0.208	**0.325**	**0.102**
Finetuned GPT-XL	0.264	0.214	0.047
Baseline GPT-XL	**0.299**	0.28	0.080

3.0 Scaling AI for Security Missions

In the final year (FY 2024), the project focused on a cyber security use case with a primary focus on the development and adaption of foundation models for cybersecurity tasks such as vulnerability assessment and annotation.

3.1 Finetuning and Evaluation Data

To create the vulnerability assessment dataset, we collected datasets for the two primary components used in Root Cause Mapping: a set of reported Common Vulnerabilities and Exposures (CVEs) and a set of Common Weakness Enumeration (CWE) labels that characterize them. Overall, we created a training set and two test sets:

- **Finetuning set**: CVEs published between 2002 and 2020, totaling 124,000 CVEs, with a subset from that same time frame reserved for validation (22,000 CVEs) stratified across training years.
- $\boldsymbol{Test_{21-23}}$: CVEs published between 2021 through 2023 (71,000 CVEs)
- $\boldsymbol{Test_{24}}$: CVEs published in 2024 (19,000 CVEs)

3.1.1 Common Vulnerabilities and Exposures (CVE) Data

Annual CVE lists were downloaded from the National Vulnerability Database (NVD) provided by the National Institute of Standards and Technology. This downloaded data included known vulnerabilities for each year, encompassing an ID field, a vulnerability description, related CWEs, and an impact report describing vulnerability severity and exploitability. Initially, we collected CVE lists from January 2002 through December 2023, followed by a second download to incorporate data from January 2024 to August 2024 for use as an additional held-out test set. The "CVE-2002" data also contained CVEs for years prior (1988-2002).

Over time, NVD's CVE list datasets are modified to correct invalid or erroneous CVEs. These entries, flagged with the keyword "REJECT," were removed from our collection sets. Each yearly list could include CVEs published in previous years due to updates in CWE labels or other content. We aligned CVEs to their publish date rather than the yearly list date. In our dataset, each CVE mapped to zero or more CWEs, with a maximum of five CWEs mapping to a given CVE and an average of one CWE per CVE. In total, this included 237,049 unique CVEs from 1988 through August 2024.

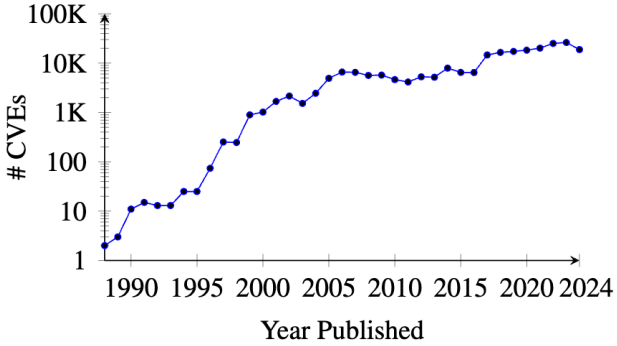

Figure 9. Volume of CVEs by original year published.

3.1.2 Common Weakness Enumerations (CWE) Data

We aggregated CWEs and corresponding metadata from the National Vulnerability Database and the Mitre Common Weakness Enumeration database, from the Research Concepts (1000) view. CWE Weaknesses have hierarchical relationships and map to one of four levels: Pillars (abstract theme), Class (1-2 dimensions reflected), Base (2-3 dimensions), and Variants (3-5 dimensions, most specific descriptions of weaknesses), as shown in Figure 10.

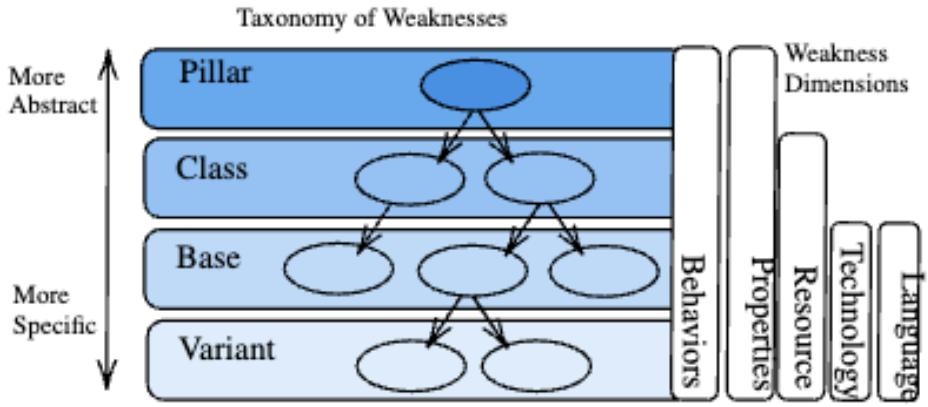

Figure 10. Taxonomy of CWE weaknesses.

After collecting our CVE dataset, we cross-referenced with the CWEs mapped to each CVE and scraped additional CWEs not collected in our initial downloads from Mitre. This resulted in a collection of 964 unique CWEs with fields including unique ID, name, description, extended description, weakness abstraction type, impact report, affected software and programming languages, as well as CWE hierarchical relationship information. Table 3 presents the top 10 most common CWEs and the number of mapping CVEs. NVD uses a subset of CWEs for mapping instead of the entire CWE list. The NVD-CWE-Other category indicates weakness types not covered by that subset.

Table 5. Table of Top 10 Most Frequent CWEs with the number of CVEs mapped to each CWE.

CWE	# CVEs with link to CWE
NVD-CWE-Other: Other	28,926
NVD-CWE-noinfo: Insufficient Information	26,781
CWE-79: Improper Neutralization of Input During Web Page Generation (Cross-site Scripting)	26,430
CWE-89: Improper Neutralization of Special Elements used in an SQL Command (SQL Injection)	11,388
CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer	11,192
CWE-787: Out-of-bounds Write	9,020
CWE-20: Improper Input Validation	8,810
CWE-200: Exposure of Sensitive Information to an Unauthorized Actor	6,617
CWE-22: Improper Limitation of a Pathname to a Restricted Directory (Path Traversal)	5,647
CWE-125: Out-of-bounds Read	5,303

3.2 Experiments

In our cyber-focused experiments, we focused on the development of models targeting vulnerability assessment and characterization – i.e., a generative approach to annotating CVE (Common Vulnerabilities and Exposures) descriptions with their related CWEs (Common Weakness Enumerations). The outputs for this task are either (a) a list of relevant CWEs or (b) a response indicating there are no relevant CWEs ("none"), or that there is insufficient information to determine relevant CWE(s) ("no-info").

Every vulnerability (CVE) is related to zero or more weaknesses (CWE). Therefore, each model's response can contain a list of many CWEs. Since CWEs have a hierarchical structure, simply judging the results of the models by checking against the ground truth does not fully convey the nuances of how well the model identifies the appropriate CWEs. Therefore, we evaluate the model output along three levels of matching to the groundtruth CWEs:

- **Exact Match**: Do the output CWE(s) exactly match the gold labelled CWE(s)?
- **Parent Match**: Are any output CWEs the parent of the gold labelled CWE(s)? This allows us to examine the models' understanding of relationships between CVEs and CWEs, even if it fails at finding the precise CWE.
- **Child Match**: Are any output CWEs the children of the gold labelled CWE(s)? As the children CWEs are more specialized than the parents, models that predict child CWEs are more specific.

For each level, we compute the precision, recall, and F1 scores to evaluate the models' performances. For each of the partial match considerations, we consider either an exact match or a partial match as correct when calculating the metrics.

This is a challenging task due to the scale of CVE and CWEs and that the list of potential CWEs is continuously evolving due to new weaknesses being added to the CWE hierarchies over time. We evaluated models in two settings: zero-shot and few-shot. The prompts for zero-shot and few-shot approaches are consistent, with the few-shot prompt including additional examples. Figure 11 provides an overview of the template used across tasks.

```
### Background:
Common Weakness Enumeration (CWE) is a formal list or dictionary of common software
    and hardware weaknesses that can occur in architecture, design, code, or
    implementation that can lead to exploitable security vulnerabilities.


Example 1: {"CVE-ID": "CVE-2009-4644", "CVE Description": "Improper Restriction of
    Operations within the Bounds of a Memory Buffer"}
Response 1: {"CWE-ID": "CWE-78", "CWE Description": "Improper Neutralization of
    Special Elements used in an OS Command ('OS Command Injection')"}
[...]


### Final Instruction:
Below is a CVE entry, consisting of a CVE ID and a CVE description. Please provide a
     list of the appropriate CWEs for the given CVE.
{"CVE-ID": "<CVE_ID>", "CVE Description": "<CVE_DESCRIPTION>"}

### Final Response:
```

Figure 11. Example prompt format. The orange text only appears in the few-shot prompts.

3.3 Model Architectures and Baselines

We used three well-performing 7 billion parameter large language models as baseline models and as initial base models for fine-tuning cyber-adapted versions of each model fine-tuned for vulnerability assessment tasking:

- Mistral (mistralai/Mistral-7B-v0.1), a fast-inference large language model that utilized grouped-query attention to handle sequences of arbitrary length.
- CodeLlama (codellama/CodeLlama-7b-hf), a large language model optimized for code generation based on Llama 2.
- WizardCoder (WizardLM/WizardCoder-Python-7B-V1.0), a code-based large language model that incorporated complex instruction fine-tuning.

Table 6. Hyperparameters and GPU resources used.

Hyper-Parameter	Value
Number of training examples	124,703
Total combined train batch size	3
Instantaneous batch size per device	128
Gradient accumulation steps	4
Total GPUs used for training	8
Total optimization steps	2,922
Memory per GPU	80GB

To fine-tune the models efficiently, we used a combination of Low-Rank Adaptation (LoRA) and Fully Sharded Data Parallel (FSDP). As CodeLlama and WizardCoder share the same base model, Llama, their trainable parameter count with LoRA was the same (29.58%), whereas Mistral's was slightly lower (28.88%).

3.4 Key Highlights

In general, the Mistral models outperform all Llama models on both test sets with roughly twice the F1-score in exact matches as well as child and parent matches. In some instances, particularly over the $Test_{2021-2023}$ set, the Llama models have stronger recall. Table 7 provides an overview of the performance across test sets and for exact, child, and parent definitions of each metric. We also investigated further which CWEs we found that the models were over- (and under-) predicting in our test sets. In Table 8, we highlight the tendency of each model to over or under represent the top 10 most commonly occurring CWEs. We find that both model types, regardless of few-shot prompting, severely under-predict *NVD-CWE-noinfo* (a class label used to denote insufficient information about the weakness or details are unknown or unspecified that prevent a CWE classification). This may be an indication of the weakness of models to attempt to produce an unreliable response of a known CWE over indicating lack of support – e.g., over-predicting a highly common CWE such as CWE-119 or CWE-200 instead.

Table 7. F1, precision (Pr), and recall (Re) results for each model under two settings: the original base model (Base) and a fine-tuned model (Fine). Results are included for exact match definitions of each metric and that allow credit for partial (child, or parent) matches in predictions. Top performance is indicated in bold.

Test 2021-23		Base						Fine-tuned					
		Llama		Mistral		Wizard		Llama		Mistral		Wizard	
		0-shot	3-shot	0-shot	3-shot	0-shot	3-shot	0-shot	3-shot	0-shot	3-shot	0-shot	3-shot
Exact	F1	0.03	0.33	0.28	0.37	0.14	0.26	0.61	0.57	**0.64**	0.61	0.61	0.52
	Pr	0.02	0.33	0.27	0.37	0.11	0.26	0.62	0.56	**0.63**	0.60	0.61	0.52
	Re	0.37	0.37	0.43	0.38	0.35	0.27	0.61	0.58	**0.66**	0.64	0.61	0.53
Child	F1	0.04	0.40	0.30	0.40	0.16	0.29	0.63	0.59	**0.66**	0.63	0.63	0.54
	Pr	0.02	0.39	0.28	0.39	0.12	0.29	0.64	0.59	**0.66**	0.63	0.63	0.54
	Re	0.48	0.44	0.48	0.41	0.41	0.30	0.63	0.60	**0.68**	0.66	0.63	0.56
Parent	F1	0.03	0.35	0.30	0.41	0.15	0.28	0.63	0.59	**0.66**	0.64	0.63	0.54
	Pr	0.02	0.35	0.28	0.41	0.12	0.28	0.64	0.59	**0.66**	0.63	0.63	0.54
	Re	0.38	0.39	0.50	0.42	0.41	0.29	0.63	0.60	**0.68**	0.66	0.63	0.56

Test 2024		Base						Fine-tuned					
		Llama		Mistral		Wizard		Llama		Mistral		Wizard	
		0-shot	3-shot	0-shot	3-shot	0-shot	3-shot	0-shot	3-shot	0-shot	3-shot	0-shot	3-shot
Exact	F1	0.02	0.23	0.19	0.25	0.10	0.22	0.37	0.31	**0.41**	0.37	0.33	0.31
	Pr	0.01	0.23	0.18	0.25	0.08	0.22	0.37	0.31	**0.41**	0.37	0.33	0.31
	Re	0.23	0.26	0.26	0.26	0.20	0.23	0.37	0.31	**0.42**	0.38	0.33	0.31
Child	F1	0.02	0.25	0.19	0.26	0.10	0.24	0.38	0.32	**0.42**	0.38	0.34	0.32
	Pr	0.01	0.24	0.18	0.26	0.09	0.23	0.38	0.32	**0.42**	0.37	0.34	0.31
	Re	0.26	0.28	0.27	0.27	0.22	0.24	0.38	0.32	**0.43**	0.38	0.34	0.32
Parent	F1	0.02	0.24	0.19	0.26	0.10	0.23	0.39	0.33	**0.43**	0.39	0.35	0.33
	Pr	0.01	0.24	0.18	0.26	0.08	0.23	0.39	0.33	**0.43**	0.39	0.35	0.32
	Re	0.24	0.27	0.28	0.27	0.21	0.24	0.39	0.34	**0.44**	0.40	0.35	0.33

Table 8. The ratio of predictions to ground truth labels made by each model per Top 10 CWE. Darker red cells indicate over-predicting, and darker blue cells indicate under-predicting. A value of 1 indicates equal predictions to ground truth. 0 denotes no predictions made for the CWE.

			CWE									
			noinfo	Other	79	89	119	787	20	200	22	125
Base	CodeLlama	0-shot	0	0	1.92	2.93	5.32	0.17	9.90	25.93	2.67	12.53
		3-shot	0.12	0	1.34	1.18	6.69	0.23	4.45	16.86	1.20	5.00
	Mistral	0-shot	0	0	1.56	2.80	19.61	0.61	16.42	37.31	1.91	5.53
		3-shot	0.10	0	1.30	1.38	6.13	0.22	7.09	12.61	1.38	1.65
	WizardCoder	0-shot	0.01	0	2.74	9.24	50.46	1.65	12.59	32.81	5.13	14.07
		3-shot	0.67	0	1.45	1.64	8.72	0.66	2.83	9.21	1.27	1.24
Fine-Tuned	CodeLlama	0-shot	0.46	0.12	1.24	1.14	0.63	1.11	3.23	2.66	1.71	1.10
		3-shot	0.08	0	1.38	1.34	2.06	1.41	4.02	4.35	2.13	1.33
	Mistral	0-shot	0.55	0.19	1.22	1.14	1.57	1.18	2.68	4.22	1.58	1.38
		3-shot	0.40	0.33	1.28	1.24	3.09	1.21	3.53	7.65	1.60	1.40
	WizardCoder	0-shot	0.15	0.02	1.24	1.16	1.12	1.17	3.50	4.22	1.60	1.33
		3-shot	0.12	0	1.58	1.28	2.98	1.27	4.33	8.36	1.72	1.39

4.0 References

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.*

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.

Alsentzer, Emily, 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.*

Andonian, Alex, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, and Josh Levy-Kramer. 2021. "GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch."

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

Black, Sid, Eric Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, and Jason Phang. 2022. "GPT-NeoX-20B: An Open-Source Autoregressive Language Model." *arXiv preprint arXiv:2204.06745.*

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, and Jared D Kaplan. 2020. "Language models are few-shot learners." *Advances in neural information processing systems* 33: 1877-1901.

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

Chinchor, Nancy, and Beth M. Sundheim. 1993. "MUC-5 Evaluation Metrics." *Fifth Message Understanding Conference (MUC-5).* Baltimore, Maryland.

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." *NAACL.* ACL.

Clark, Peter, 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.*

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

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

Gao, Leo, Stella Biderman, Sid Black, Laurence Golding, and Travis Hoppe. 2020. "The pile: An 800gb dataset of diverse text for language modeling." *arXiv preprint arXiv:2101.00027.*

Gu, Yu, 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.

Guo, Jiang, 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.*

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. "The winograd schema challenge." *KR'12.* AAAI Press.

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

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

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

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B Brown, and Benjamin Chess. 2020. "Scaling laws for neural language models." *arXiv preprint arXiv:2001.08361.*

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. "Deduplicating Training Data Makes Language Models Better ." *ACL.* ACL.

Kim, Sunghwan, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li et al. 2019. "PubChem 2019 update: improved access to chemical data." *Nucleic Acids Research.*

Krallinger, Martin, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman et al. 2015. "The CHEMDNER corpus of chemicals and drugs and its annotation principles." *Journal of Cheminformatics* 1-17.

Lee, Jinhyuk, 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* (Oxford University Press) 36 (4): 1234-1240.

Leo Gao, Stella Biderman, Sid Black, Laurence Gold- ing, Travis Hoppe, Charles Foster, Jason Phang, Ho- race He, Anish Thite, Noa Nabeshima, et al. 2020. "The pile: An 800gb dataset of diverse text for lan- guage modeling." *arXiv preprint arXiv:2101.00027.*

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

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

Lo, Kyle, Lucy Lu Wang, Mark Neumann, Rodney Michael Kinney, and Daniel S. Weld. 2020. "S2ORC: The Semantic Scholar Open Research Corpus." *ACL.*

Luu, Kelvin, 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 .*

Marie-Catherine De Marneffe, Mandy Simons, and Ju- dith Tonhauser. 2019. "The commitmentbank: Inves- tigating projection in naturally occurring discourse." *Proceedings of Sinn und Bedeutung.*

Mihaylov, Todor, 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.*

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

Naseem, Usman, 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.*

Peng, Yifan, 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.*

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

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

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.*

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

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

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

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

Shibata, Yusuxke, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. 1999. "Byte pair encoding: A text compression scheme that accelerates pattern matching."

Shin, Hoo-Chang, 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.*

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

Shoeybi, Mohammad, 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 .*

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

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

Taylor, Ross, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. "Galactica: A large language model for science." *arXiv preprint arXiv:2211.09085.*

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.*

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao. 2020. "Huggingface's transformers: State-of-the-art natural language processing."

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. "Can a suit of armor conduct elec- tricity? a new dataset for open book question answer- ing." *arXiv preprint arXiv:1809.02789.*

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.*

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

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et. al. 2020. "Piqa: Reasoning about physical commonsense in natural language." *Proceedings of the AAAI conference on artificial intelligence.*

Yuan, Zheng, 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 Salakhut- dinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books." *Proceedings of the IEEE international conference on computer vision.*

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov