M2D2: A Massively Multi-Domain Language Modeling Dataset

Machel Reid[1]*, **Victor Zhong**[2], **Suchin Gururangan**[2], **Luke Zettlemoyer**[2]
[1]The University of Tokyo, [2]University of Washington
machelreid@google.com, {vzhong,sg01,lsz}@cs.washington.edu

Abstract

We present M2D2, a fine-grained, massively multi-domain corpus for studying domain adaptation in language models (LMs). M2D2 consists of 8.5B tokens and spans 145 domains extracted from Wikipedia and Semantic Scholar. Using ontologies derived from Wikipedia and ArXiv categories, we organize the domains in each data source into 22 groups. This two-level hierarchy enables the study of relationships between domains and their effects on in- and out-of-domain performance after adaptation. We also present a number of insights into the nature of effective domain adaptation in LMs, as examples of the new types of studies M2D2 enables. To improve in-domain performance, we show the benefits of adapting the LM along a domain hierarchy; adapting to smaller amounts of fine-grained domain-specific data can lead to larger in-domain performance gains than larger amounts of weakly relevant data. We further demonstrate a trade-off between in-domain specialization and out-of-domain generalization within and across ontologies, as well as a strong correlation between out-of-domain performance and lexical overlap between domains.[1]

1 Introduction

Even though they can contain a wide variety of different types of domains, the texts that make up the corpora used to train and evaluate language models (LMs) are often treated as if they are all the same. This makes it challenging to characterize LM performance under diverse data distributions and understand how to effectively adapt LMs to new ones. To address these challenges, we develop M2D2, a **M**assively **M**ulti-**D**omain **D**ataset, with 145 subdomains and a human-curated hierarchy for studying fine-grained domain adaptation.

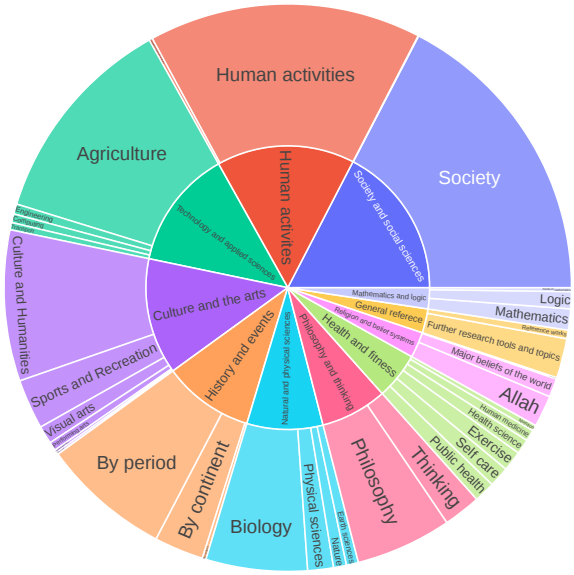

Figure 1: Visualization of the two-level fine-grained domain hierarchy in the Wikipedia portion of M2D2.

Prior work on domain transfer focuses on a small number of broad domains (typically 4-20; Gururangan et al., 2020; Gao et al., 2021; Gururangan et al., 2021). In contrast, domains in M2D2 are fine-grained and organized into a hierarchy derived from human-curated ontologies in Wikipedia (Figure 1) and Semantic Scholar (Figure 2). Unlike prior work, the fine granularity of M2D2 enables the study of transfer to naturally occurring data-scarce domains recognized by human curators (e.g. Philosophy, Public Health, Transport). This hierarchy enables the study of domain transfer at varying levels of topic granularity. For instance, how should we combine widely available internet text (entire corpus), text on computer science (coarse domain), and scarce corpus on machine learning (fine domain) to improve performance in the machine learning domain? To the best of our knowledge, M2D2 is the first dataset that combines fine domain granularity with human-curated domain hierarchy in a massively multi-domain setting.

*Currently at Google Research
[1]We release our dataset publicly at https://github.com/machelreid/m2d2.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 964–975
December 7-11, 2022

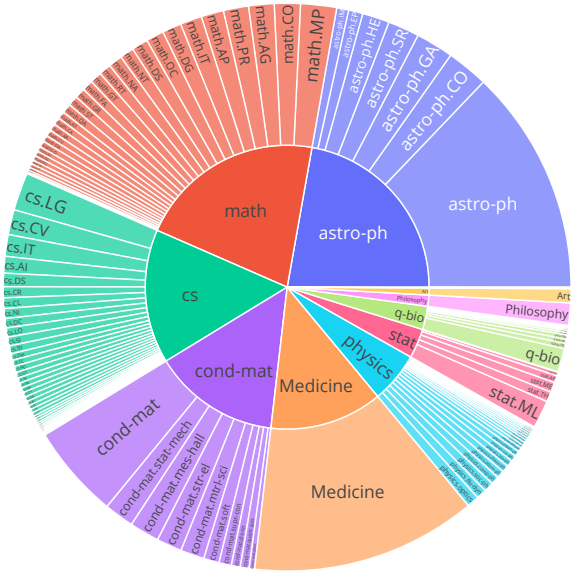

Figure 2: Visualization of the hierarchies contained within the S2ORC portion of M2D2.

Using M2D2, we investigate the following questions, as examples of the broad classes of new questions that can be asked: (1) how well do coarse and fine domains transfer to each other across the hierarchy? (2) which features and aspects of a domain are important for transfer? (3) how important is domain specificity versus breadth? We perform preliminary experiments analyzing transfer between similar domains, disparate domains, and hierarchically related domains. Moreover, we explore how to select source domains to improve transfer performance.

We present baseline experiments using a GPT2 (Radford et al., 2019) language model. We find that (1) more specific data is often more important for performance than larger, less-specific data, shown by our comparison of coarse-grained, fine-grained and coarse-to-fine adaptation comparison (in which coarse-to-fine performed best) , (2) vocabulary overlap is a surprising good indicator for transfer, and (3) data source provenance information is often a better predictor than ontology when predicting transferability, perhaps indicating a more multi-faceted definition of *domain* could be developed in future work.

Given the importance of fine granularity domains in language modeling, we hope that M2D2 will encourage the community to further study domain transfer: how do we identify hierarchical fine-grained domains in naturally occurring text, and how do we leverage this fine-grained domain hierarchy to improve domain transfer.

2 M2D2

M2D2 consists of a large quantity of fine-grain domains. Unlike prior work that defines the domain of a corpus using its source (e.g. the web text domain; Chronopoulou et al., 2021), we derive domains from a human-curated Wikipedia and arXiv ontologies. In this section, we describe how M2D2 is collected and organized.

2.1 Domain Organization

One of the unique properties of M2D2 is its hierarchical nature, enabling the study of transfer at different levels of domain granularity. We assume a particular corpus to have $L_0, \ldots, L_K$ levels of hierarchy, where L_0 refers to the lowest or most coarse-grained/broad level (i.e. the whole dataset), and L_K refers to the highest or most fine-grained/specific level. A given level of hierarchy L_i contains N_i domains $\mathcal{D}^i_{N_i}$,

$$L_i = [\mathcal{D}^i_0, \ldots, \mathcal{D}^i_j, \ldots, \mathcal{D}^i_{N_i}] \quad (1)$$

$\mathcal{D}^i_j$ is composed of multiple subdomains $\{\mathcal{D}^{i+1}_0, \ldots, \mathcal{D}^{i+1}_{N_{i+1}}\}$, which are represented in the next level of the hierarchy L_{i+1}. Similarly, we assume that a given subdomain is contained within a larger domain.

For the rest of the paper, we use L1 and L2 to represent the two levels of a K level hierarchy that we consider in this paper.

2.2 Dataset Collection

We collect M2D2 from two resources, Wikipedia and Semantic Scholar. This allows us to explore domain adaptation in a massively multi-domain setting among domains of varying granularity, while also allowing us to test whether our findings hold across different data sources.

Semantic Scholar We use the S2ORC corpus (Lo et al., 2020), a large corpus of English academic papers annotated with extensive metadata. Using this corpus, which is already categorized into L1-domains representing broader fields of academic research (e.g. Computer Science, Physics), we extract L2-domains by finding a given paper's respective arXiv[2] category (e.g. "Computation and Language" $\in$ Computer Science).

[2] https://arxiv.org

L1 (Abbrv)	Size	#L2	#Tokens	Examples of L2 domains
Health and fitness (HEAL)	761.2MB	7	116M	Exercise, Health Science
History and events (HIST)	1.4GB	4	226M	Regions, Periods
Society and social sciences (SOCI)	2.3GB	3	379M	Society, social sciences
Technology and applied sciences (TECH)	1.9GB	5	297M	Agriculture, Computing
Culture and the arts (CULT)	2.0GB	8	289M	Games and Toys, The arts and entertainment
Natural and physical sciences (NATU)	1.2GB	5	189M	Physical sciences, Earth sciences
Human activites (HUMA)	2.1GB	3	343M	Impact of human activity
Mathematics and logic (MATH)	332.3MB	4	52M	Mathematics, Logic
General reference (GENE)	385.3MB	3	60M	Research tools and topics, Reference works
Religion and belief systems (RELI)	428.0MB	4	64M	Major beliefs of the world, Belief systems
Philosophy and thinking (PHIL)	1.0GB	3	165M	Philosophy, Thinking
Mathematics (math)	4.5GB	26	1.4B	Topology, Number Theory
Quantitative Biology(q-bio)	1.9GB	3	336M	Biomolecules, Cell Behavior
Physics	4.1GB	12	737M	General Physics, Biological Physics
Nonlinear Sciences (nlin)	730MB	5	134M	Self-Organizing Systems, Chaotic Dynamics
Condensed Matter (cm)	3.8GB	10	688M	Materials Science, Quantum Gases
Economics (econ)	67MB	3	11M	Econometrics, General Econometrics, Theory
Computer Science (cs)	4.5GB	23	1.1B	Machine Learning, Databases, Graphics
Statistics (stat)	2.4GB	4	450M	Applications, Methodology
Astrophysics (astro-ph)	4.0GB	7	728M	Earth/Planetary, Cosmology
Art†	575MB	1	98M	—
Philosophy†(phil)	919MB	1	156M	—
Average$_{\pm \text{s.d.}}$	1.9G$_{\pm 1.7G}$	6.6$_{\pm 6.2}$	373M$_{\pm 347M}$	—
Total	41GB	145	8.5B	—

Table 1: Dataset statistics for M2D2. We list L1 domains, with their corresponding sizes, number of L2 domains, number of tokens, and examples of L2 domains. †These domains did not have any subdomains in the arXiv ontology.

Wikipedia We crawl the Wikipedia ontology,[3] which lists major categories contained within Wikipedia. Within these major categories or L1-domains, we then proceed to look up the category pages within a given L1-domain, and gather respective L2-domains. This procedure yields a hierarchy of domains contained within Wikipedia. We then download the Wikipedia data dump, which we clean using `wikiextractor`[4] and assign a given page to its respective domain.

2.3 Unique Properties

M2D2 has the following major unique properties when compared to previous domain adaptation datasets. First, it is massively multi-domain: we have 145 L2 domains grouped into 22 L1 domains, which allows us to test domain adaptation for language modeling on a variety of axes (such as hierarchy, subject matter, and ontology) that would be more difficult with more coarse-grained datasets. Second, M2D2 is hierarchical: this allows us to also test the performance of domain specificity versus domain breadth in more flexible adaptation settings.

We describe dataset statistics in Table 1, including dataset size (measured in MB/GB), token count (measured by whitespace tokenization), and the number of L2 domains within each L1 domain. M2D2 contains 8.5B tokens, with an average of 373 million tokens per L1 domain. Demonstrating the hierarchical nature of M2D2, we also list examples of L2 domains contained within the L1 domains (e.g. Computing $\in$ Technology and Applied Sciences, Topology $\in$ Mathematics) which are also graphically shown in Figures 1 and 2).

2.4 Dataset Splits

We split each domain into the respective train, validation, and test sets. To prevent data leakage between the domains when pages belong to two or more domains, we construct validation and test sets from pages that are not contained within any other domains on the same level of hierarchy. For example, the page for "Biotechnology" overlaps in domain with both *Biology* $\in$ *Natural and Physical Sciences* and *Engineering* $\in$ *Technology and*

[3] https://en.wikipedia.org/wiki/Wikipedia:Contents/Categories

[4] https://github.com/attardi/wikiextractor

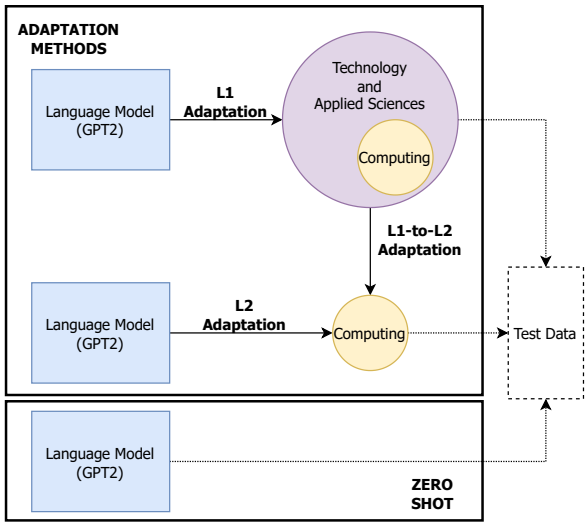

Figure 3: The types of domain adaptation that we consider in this work: L1, L2, and L1-to-L2 adaptation. Here, we use "Technology and Applied Sciences" to illustrate our L1 domain and "Computing" to illustrate our L2 domain. Bold arrows refer to adaptation steps, and dotted lines refer to an evaluation phase.

Applied Sciences so this would not be included in any evaluation set due to the potential for direct leakage. However, the page for "Computer" is only in *Computing* $\in$ *Technology and Applied Sciences* and therefore could be included in an evaluation set. We include at least 1 million tokens in the validation and test sets, respectively. This enables us to have a precise evaluation set of texts that only belong to a single fine-grained domain.

3 Experiments

As examples of the types of new studies M2D2 enables, we explore a number of key questions about the nature of effective domain adaptation in language models. For example, how does one best specialize a language model to a domain, given an ontology? How well can adapted models be applied out-of-domain, within and across ontologies? What features of target domains are predictive of out-of-domain transfer?

In this section, we present a set of experiments that begin to answer these questions. First, we study the impact of adapting to the L1 and L2 domains of our dataset on in-domain (§3.2) and out-of-domain (§3.3) language modeling performance. Then, we perform an analysis of lexical features in domains that are predictive of out-of-domain performance (§3.4).

3.1 Experimental setup

In all experiments, we use the 112M GPT2 model (Radford et al., 2019) as the baseline model. Our implementation is based on HuggingFace Transformers (Wolf et al., 2020) and PyTorch (Paszke et al., 2019). All adaptation techniques are performed using Adam (Kingma and Ba, 2015), dropout value of 0.2 (Srivastava et al., 2014), using a learning rate of 5e-5 and a batch size of 64000 tokens. We train all models for a maximum of 1 million iterations and perform early stopping over the validation set. All experiments are run on 8 NVIDIA V100 GPUs.

When adapting our GPT2 model to domains in M2D2, we use one of three settings:

L1 Adaptation We continue training on a given L1 domain (e.g. Computer Science).

L2 Adaptation We continue training on a given L2 domain (e.g. Machine Learning).

L1-to-L2 Adaptation Given a L2 domain (e.g. Machine Learning), we first perform L1 adaptation on its corresponding L1 domain (e.g. Computer Science), and then we further perform L2 adaptation. This setting similar to multi-stage adaptive pretraining approaches used for supervised tasks (Gururangan et al., 2020).

For all techniques, we evaluate test perplexity on L2 domains validation sets. Due to the large quantity of L2 domains, we aggregate L2 results by their corresponding L1. For each ontology, we report the average and standard deviation ($\text{average}_{\text{s.d.}}$) of perplexities across L2 domains in each L1.

3.2 In-Domain Results

The first set of experiments in this study considers the impact of adapting the language model to different levels of the M2D2 ontologies. We only consider in-domain perplexity, or the perplexity of model on the domain it is adapted to.

Adaptation improves in-domain performance despite pretraining. Table 2 shows test-set perplexities on L2 domains, averaged across each L1 domain, after performing each adaptation technique (see Appendix on full results). First, we observe that all proposed adaptation techniques improve performance over the base GPT-2 model. This highlights the effectiveness of adaptation in improving in-domain performance, even when considering domains that the language model has likely

Wiki	HEAL	HIST	SOCI	TECH	CULT	HUMA	MATH	GENE	RELI	PHIL	NATU	Avg
GPT2	23.1	27.5	24.5	27.8	27.5	28.9	26.6	25.9	26.3	26.2	26.7	26.5
L1	$18.1_{2.5}$	$20.9_{0.5}$	$19.7_{0.8}$	$22.3_{0.8}$	$21.2_{2.3}$	$23.0_{1.4}$	$18.3_{5.2}$	$21.6_{0.8}$	$19.8_{0.5}$	$21.8_{0.6}$	$20.8_{3.2}$	20.7
L2	$17.5_{2.7}$	$17.8_{1.9}$	$17.5_{0.7}$	$21.8_{1.0}$	$21.7_{2.6}$	$22.4_{0.9}$	$17.8_{5.2}$	$20.8_{1.0}$	$18.3_{0.4}$	$21.0_{0.4}$	$21.7_{1.6}$	19.8
L1-to-L2	$\mathbf{16.8}_{2.7}$	$\mathbf{16.7}_{2.1}$	$\mathbf{15.4}_{0.5}$	$\mathbf{21.4}_{0.9}$	$\mathbf{20.6}_{2.6}$	$\mathbf{22.0}_{0.8}$	$\mathbf{17.1}_{5.0}$	$\mathbf{19.6}_{1.1}$	$\mathbf{16.9}_{0.5}$	$\mathbf{20.5}_{0.4}$	$\mathbf{20.3}_{1.5}$	18.8
S2ORC	**Math**	**Econ**	**CS**	**CM**	**Physics**	**Art**	**Phil**	**Stat**	**Q-Bio**	**Nlin**	**Astro-Ph**	Avg
GPT2	$26.1_{2.8}$	$28.2_{2.7}$	$26.8_{2.9}$	$29.7_{1.2}$	$32.7_{2.1}$	$35.1_{0.0}$	$32.9_{0.0}$	$22.7_{7.3}$	$30.1_{1.3}$	$25.5_{1.4}$	$31.6_{1.5}$	29.2
L1	$9.2_{3.4}$	$15.9_{2.2}$	$15.4_{4.0}$	$12.5_{1.0}$	$17.1_{1.7}$	$27.7_{0.0}$	$24.4_{0.0}$	$11.0_{3.5}$	$22.6_{2.2}$	$9.8_{2.4}$	$15.5_{3.2}$	16.5
L2	$8.0_{3.2}$	$13.4_{2.1}$	$15.1_{6.7}$	$12.0_{1.3}$	$16.5_{1.3}$	$27.7_{0.0}$	$24.4_{0.0}$	$10.2_{2.5}$	$21.0_{1.3}$	$9.6_{2.1}$	$14.0_{2.3}$	15.7
L1-to-L2	$\mathbf{7.5}_{3.2}$	$\mathbf{12.5}_{2.2}$	$\mathbf{14.0}_{5.9}$	$\mathbf{11.5}_{1.0}$	$\mathbf{16.1}_{1.6}$	$\mathbf{27.7}_{0.0}$	$\mathbf{24.4}_{0.0}$	$\mathbf{9.3}_{3.3}$	$\mathbf{20.3}_{1.0}$	$\mathbf{9.2}_{2.1}$	$\mathbf{12.9}_{2.3}$	**15.0**

Table 2: In-domain test perplexities, aggregated to each L1 domain. We look at the impact of L1 vs L2 vs L1-to-L2 finetuning settings when compared to simply finetuning on L1. L2 Adaptation is usually more effective than L1 Adaptation, emphasizing the importance of fine-grained domains, with a coarse-to-fine setup using L1-to-L2 Adaptation is most effective. This finding is statistically significant ($p < 0.05$; measured using the Kolmogorov-Smirnov test).

Wiki	HEAL	HIST	SOCI	TECH	CULT	HUMA	MATH	GENE	RELI	PHIL	NATU	Avg
L1	$23.6_{3.5}$	$23.2_{2.0}$	$22.4_{2.2}$	$22.4_{2.3}$	$22.3_{2.2}$	$22.7_{2.0}$	$25.1_{3.4}$	$24.2_{2.3}$	$24.7_{2.8}$	$23.6_{2.7}$	$23.3_{3.2}$	23.3
L2	$26.1_{3.8}$	$26.1_{3.9}$	$25.7_{2.7}$	$26.1_{3.5}$	$27.0_{3.7}$	$25.6_{3.6}$	$28.9_{6.9}$	$25.1_{2.4}$	$26.3_{2.9}$	$24.1_{2.6}$	$26.3_{3.7}$	26.1
L1-to-L2	$25.5_{3.8}$	$25.9_{3.8}$	$25.2_{2.6}$	$26.0_{3.3}$	$27.0_{3.7}$	$25.1_{3.6}$	$28.5_{7.0}$	$24.5_{2.4}$	$26.2_{2.9}$	$23.2_{2.6}$	$25.2_{3.7}$	25.7
S2ORC	**Math**	**Econ**	**CS**	**CM**	**Physics**	**Art**	**Phil**	**Stat**	**Q-Bio**	**Nlin**	**Astro-Ph**	Avg
L1	$32.0_{17.2}$	$28.8_{10.9}$	$23.1_{10.1}$	$24.9_{14.0}$	$22.8_{10.6}$	$26.8_{3.3}$	$25.7_{3.9}$	$23.4_{11.5}$	$23.2_{11.3}$	$23.8_{12.9}$	$26.2_{12.8}$	25.5
L2	$36.0_{21.9}$	$33.4_{11.1}$	$32.1_{18.7}$	$32.7_{17.3}$	$25.4_{12.1}$	$26.8_{3.3}$	$25.7_{3.9}$	$32.7_{24.7}$	$33.2_{19.6}$	$34.8_{22.4}$	$27.2_{11.4}$	30.9
L1-to-L2	$36.8_{24.8}$	$31.9_{12.6}$	$31.0_{22.0}$	$30.2_{18.2}$	$24.2_{11.4}$	$26.8_{3.3}$	$25.7_{3.9}$	$30.4_{23.0}$	$32.1_{23.4}$	$36.5_{30.8}$	$27.5_{15.1}$	30.3

Table 3: Out-of-domain test perplexities, aggregated to each L1 domain. We look at the impact of L1 vs L2 vs L1-to-L2 finetuning settings when compared to simply finetuning on L1. We can see that L2 Adaptation and L1-to-L2 Adaptation are generally less performant in out-of-domain settings that L1 Adapted models, given their in-domain specification. The comparison between L1 versus L2 is statistically significant $p < 0.01$.

been exposed to during pretraining (as is the case with Wikipedia; L1 adaptation results in a 5.8 decrease in perplexity). For domains which the language model is less likely to have been exposed to during pretraining, this is more pronounced (as is the case with S2ORC; L1 adaptation results in a 12.7 decease in perplexity).

Specificity and hierarchy is more important than broad coverage in adaptation. Next, we observe that in most cases, adapting to L2 domains is more beneficial to in-domain performance than adapting to L1 domains. Adaptation to finer-grained domains better specializes a language model, even though these domains are much smaller than their L1 counterparts. Finally, we observe that using L1-to-L2 adaptation further benefits in-domain performance over L2 adaptation in all cases. Our results suggest that adapting to smaller amounts of domain-specific data leads to more effective in-domain specialization than adapting to large quantities of data that may be more weakly domain-relevant. Moreover, the best results may be achieved by organizing the target domain into subsets of broader and fine-grained data, and adapting along this hierarchy. However, this approach has increased memory and computational requirements relative to solely relying on L1 Adaptation.

3.3 Out-of-Domain Results

We also study the effects of our adaptation techniques on out-of-domain performance, by performing zero-shot inference with adapted models on domains (e.g. Art) *other* than the ones they are adapted to (e.g. Machine Learning). We first transfer models between domains in the same ontology (e.g. Wikipedia $\rightarrow$ Wikipedia), and then across ontologies (e.g. Wikipedia $\rightarrow$ S2ORC).

L2 Adaptation decreases out-of-domain performance. We show out-of-domain performance for each adaptation technique in Table 3. We show that conversely to L2 and L1-to-L2 adaptation which significantly improved in-domain performance, this comes with the tradeoff at less performance in both

Domain	NATU	TECH	SOCI	HEAL	HIST	RELI	CULT	GENE	MATH	HUMA	PHIL	Avg
NATU	—	25.5	22.1	20.0	24.5	23.5	25.7	23.7	21.0	25.6	23.2	23.3
TECH	**23.6**	—	20.8	**19.4**	23.1	22.7	23.5	22.6	21.7	24.5	22.4	22.5
SOCI	23.8	24.2	—	19.8	22.3	21.7	**23.4**	**22.0**	22.6	24.1	**22.0**	**22.4**
HEAL	24.3	25.6	21.6	—	24.4	23.7	25.2	24.0	25.0	26.2	23.9	23.9
HIST	24.7	25.3	20.7	21.8	—	21.4	24.2	22.8	24.0	**23.9**	22.6	23.0
RELI	26.3	28.2	21.9	22.8	24.0	—	25.8	24.4	26.0	26.3	24.0	24.5
CULT	23.7	24.3	20.6	20.1	23.0	22.1	—	22.5	22.8	24.4	22.2	22.5
GENE	25.4	26.4	21.8	22.1	24.2	23.2	25.4	—	24.5	26.2	23.3	24.1
MATH	26.3	26.7	23.7	23.1	26.4	25.0	27.1	25.2	—	28.3	24.4	25.0
HUMA	23.9	**24.0**	**20.1**	20.7	**22.0**	**21.3**	24.1	22.3	23.2	—	22.3	22.5
PHIL	25.1	25.7	21.8	21.3	24.4	22.9	24.7	23.5	**20.9**	26.0	—	23.5
Avg	24.4	25.3	21.3	**20.8**	23.6	22.5	24.6	23.1	22.7	25.3	22.9	23.3

Table 4: Out-of-domain transfer performance between all L1 domains (using abbreviations from Table 1) in the Wikipedia portion of M2D2. For each domain, we use the first four letters to refer to itself. The x-axis shows evaluation domains, and the y-axis shows training domains.

Domain	math	econ	cs	cm	physics	Art	Philosophy	stat	q-bio	nlin	astro-ph	Avg
math	—	25.0	22.0	21.2	35.8	66.1	57.2	19.6	38.8	13.6	43.2	32.0
econ	18.5	—	23.8	24.6	35.1	48.1	43.9	15.3	33.7	20.0	37.5	28.8
cs	12.6	17.6	—	18.0	24.5	43.2	40.1	13.9	26.6	14.0	28.6	23.1
cm	13.6	20.7	21.6	—	**17.9**	55.9	50.0	16.2	26.4	**13.2**	25.8	24.9
physics	14.1	20.5	21.0	**14.1**	—	46.2	41.9	15.8	**24.8**	13.3	**22.1**	**22.8**
Art	22.9	25.8	25.9	27.5	31.1	—	**29.0**	21.3	29.1	22.6	31.7	26.8
Philosophy	20.8	24.7	23.4	26.2	31.2	**30.4**	—	20.3	28.1	21.5	31.4	25.7
stat	12.7	**14.0**	**18.2**	17.9	24.8	47.0	43.2	—	26.6	14.8	27.0	23.4
q-bio	13.7	18.1	19.2	14.6	20.9	48.1	42.9	**13.6**	—	14.3	26.8	23.2
nlin	**11.0**	19.7	20.7	13.3	22.3	51.7	45.8	15.7	25.9	—	25.9	23.8
astro-ph	16.6	23.9	25.2	17.1	23.6	54.4	48.1	17.8	30.9	15.2	—	26.2
Avg	**15.1**	20.5	21.5	18.8	25.8	47.1	42.4	16.4	28.5	15.7	28.7	25.5

Table 5: Out-of-domain transfer performance between all L1 domains in the S2ORC portion of M2D2. "GPT2" refers to the zero-shot performance of the LM on our dataset.

L2 and L1-to-L2 settings when compared to L1 Adaptation.

Specific adaptation transfers better to related categories across ontology. Although the two data sources in M2D2 differ considerably in style and content, their ontological categories partially overlap. For example, *Mathematics* and *Art* appear in both Wikipedia and Semantic Scholar. Is it possible to transfer between corresponding categories across ontologies?

To answer this question, we first manually align L1 domains from Wikipedia and Semantic Scholar with similar ontological categories (e.g., grouping *Mathematics* from Wikipedia and *Mathematics* from S2ORC). We then apply a model adapted to an L1 domain in a source ontology onto its corresponding L1 domain in a target ontology. We compare this cross-ontology performance with two baselines: 1) the average out-of-domain performance of other L1 adapted models in the target ontology and 2) the in-domain performance of a model adapted to the target L1 domain.

Our results are displayed in Table 6. We observe that while L1 adapted models are effective at transferring to other domains *within* an ontology, they are less effective at transferring to corresponding domains *outside* an ontology. Surprisingly, in all cases, transferring outside an ontology performs even worse than using the base GPT-2 model with no additional adaptation. Moreover, the average out-of-domain performance of L1 adapted models generally outperforms cross-ontology performance, indicating properties shared within an ontology (e.g. style) could be transferred.

Summary Our investigations into the out-of-domain performance of adapted language models

S2ORC	Mathematics	Computer Science	Art	Philosophy	Physics
S2ORC (in-domain)	9.2	15.4	27.7	24.4	17.1
Wiki (in-domain)	19.6	26.8	35.3	33.4	29.6
S2ORC (out-of-domain)	15.1	21.5	47.1	42.4	25.8
Wiki	**MATH**	**TECH**	**CULT**	**PHIL**	**NATU**
Wiki (in-domain)	18.3	22.3	21.2	21.8	20.8
S2ORC (in-domain)	29.6	29.5	26.8	27.0	31.5
Wiki (out-of-domain)	22.7	25.3	24.6	22.9	22.9

Table 6: Transfer performance between corresponding domains(Math↔Mathematics and Logic(Math), Computer Science↔Technology and Applied Sciences, Art↔Culture and the Arts, etc..) in both ontologies. It can be seen that provenance is a stronger indicator of transfer performance on M2D2 than ontological correspondence.

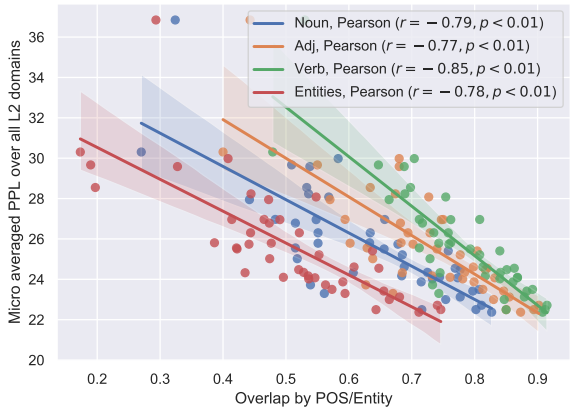

Figure 4: The relationship between overlap and transfer performance over all Wikipedia L2 domains. Entities, verbs, nouns and adjectives are all strongly correlated with performance across domains.

reveals a tradeoff between specialization and generalization. The more fine-grained the specialization of the language model, the less one can expect it to be applicable outside of the domain it was trained on. This effect size increases as we move outside the ontology: models trained on one ontology are not useful in other ontologies, despite being trained on similar categories of data. These findings lead us to believe that domain adaptation should be studied from a multi-faceted perspective to exploit specific aspects of domain (e.g. style, content). Future work may look at reducing the tradeoff between highly domain specialized models and out of domain performance, perhaps through ensembling or other approaches.

3.4 Lexical indicators of out-of-domain performance

Looking closer at the out-of-domain performance of L1 models, we see intuitive relationships between subject similarity and zero-shot out-of-domain transfer performance (Table 4). For example, *Society* and *Human Activities* domains tend to transfer well to each other, whereas *Religion* and *Mathematics* do not transfer as well. These findings suggest that out-of-domain transfer is correlated with content overlap. In this section, we present some basic lexical indicators of out-of-domain performance which support this hypothesis.

Vocabulary overlap strongly correlates with transfer regardless of part-of-speech. Figure 4 shows the correlation of vocabulary overlap a given part-of-speech tag (VERB, NOUN, ADJ) or entities and average out-of-domain performance on M2D2. We compute this by taking the top-k ($k = 1000$) most common words for a given domain which correspond to a given POS tag. For every given domain, we then calculate the intersection of shared most common words corresponding to the part-of-speech tag with the entirety of M2D2 and plot them against the L2-domain-averaged perplexity over the entire dataset. We use `spacy` (Honnibal and Montani, 2017) for both entity recognition and POS tagging. We find that vocabulary overlap is a strong predictor of transfer performance regardless of part-of-speech, perhaps indicating its relevance in transfer between fine-grained domains.

Related domains mostly transfer domain-specific tokens. We analyse domain adaptation at a token-level to characterize what different adaptation settings transfer. Specifically, we measure which tokens are most impacted in terms of per-word perplexity when we finetune on a domain-specific corpus. We do this by taking the difference between the softmax-normalized probability of pre-

Transfer	Domain-specific	General	Examples
Distant L1	25.7%	74.3%	Blockchain, Alexa
Easy L1	12.3%	87.7%	the, cache
Zero-shot	23.4%	76.6%	renewals, Markov
L1-to-L2	31.6%	68.4%	lambda DCS, Tacotron

Table 7: Average percentage of tokens transferred in-domain and out of domain. Examples are taken from Philosophy→Computer Science, Statistics→Computer Science, GPT2→Computer Science, and Computer Science→Computation and Language.

dicting a given word in a given domain when comparing two models adapted to different corpora.

We compare S2ORC adapted models in four settings: two best-transferred domains (a proxy for similar domains; easy transfers), two worst transferred L1 domains (a proxy for distant domains; difficult transfers), L1-to-L2 Adaptation (hierarchical domain transfer), and no adaptation (zero-shot performance of the base LM). We show the distribution between domain-specific (terms that appear less than 0.00001% of the time in any other domain) and non-domain-specific terms in Table 7 that appear in the top 1000 most adapted words. Finally, we show representative samples of tokens with the greatest change after adaptation. We find that the most changed tokens between easy transfers (e.g. Statistics and Computer Science) are non-domain-specific words (such as *the*) but harder transfers include words that are more domain specific (such as *Blockchain*).

Summary Our preliminary analyses suggest that simple lexical characteristics of domains are strong indicators of how well an adapted model may generalize. Developing computationally inexpensive indicators of transfer (as lexical overlap is), is important for domain transfer to find the best out of a large set of candidate corpora to perform adaptation to a target domain. This would allow one to approximately find the best corpus, without the computational overhead of adapting to all candidate corpora.

4 Related Work

Domain Adaptation Techniques (Gururangan et al., 2020) show that pretrained language models can be adapted to new domains by continued pre-training on domain-specific corpora. Chronopoulou et al. (2021); Gururangan et al. (2021) build upon this work by using hierarchically constructed domain specific adapters/experts (Houlsby et al., 2019). Another line of work in domain generalization is to simply scale the model pre-training on a corpus containing different domains (e.g. GitHub, PubMed) such as done with GPT-J (Wang and Komatsuzaki, 2021) and the Pile (Gao et al., 2021). Dery et al. (2021) also look to bridge these approaches by learning a task/domain specific mixture of tasks. Overall, however, much of this work (Daumé III, 2007; Ruder et al., 2017; Ruder and Plank, 2018; Gururangan et al., 2020; Ramponi and Plank, 2020; Gururangan et al., 2021; Chronopoulou et al., 2021) fits in a paradigm in which a base model is trained further on domain-specific corpora and then testing on tasks within that domain (e.g. abstract sentence role classification (Bird et al., 2008) for the scientific domain). M2D2 is complementary to these works in providing a testbed for fine-grained and hierarhical adaptation across a large quantity of domains.

Domain Adaptation Datasets One approach toward improved pre-trained language models includes building large-scale pre-training datasets that contain a diverse set of domains, such as the Pile (Gao et al., 2021). Overall, this emphasis has lead to improved performance in various domains, especially with large-scale pre-trained language models, such as GPT-J (Wang and Komatsuzaki, 2021). Another line of work has been in documenting large-scale web-crawled datasets, so practitioners and researchers can be more informed and mindful of the data used (Dodge et al., 2021). Our work extends this thread with a massively multi-domain corpus with a manually curated ontology that can be used to study fine-grained and hierarchical domain transfer.

5 Conclusion

We developed M2D2, a new massively multi-domain language modeling dataset for studying domain adaptation in language models. M2D2 consists of 145 fine-grained domains (curated from Wikipedia and Semantic Scholar) that are hierarchically organized using domain-specific ontologies. Using M2D2, we find that domain precision is more important than data quantity to improve in-domain performance, a tradeoff between specialization and out-of-domain generalization. We release M2D2 publicly to spur further research on building effective language models on highly heterogeneous data.

6 Limitations

In this work, we only consider adaptation techniques that assume domains are monolithic and non-overlapping. Future work may instead explore modeling the data as a mixture of domains, which may improve out-of-domain performance. In addition, M2D2 only covers two data sources (Wikipedia and Semantic Scholar). Future work could expand this corpus with ontologies from other data sources, such as Reddit, which have a fine-grained and hierarchical domains. Moreover, data sourced from the web may contain hate speech and other harmful content, which may be reproduced by language models adapted to such data. The data sources we use adhere to research-friendly data licenses, but training models on web-curated data while maintaining the rights of authors as data subjects and creators remains an open problem.

Acknowledgements

We thank Nikita Haduong, Jungo Kasai, Sophia Serrano, Wenya Wang for their feedback and proofreading comments. We thank Jesse Dodge, Alexandra Chronopoulou, and Matthew Peters for sharing code for their work on domain adaptation. We also thank Sebastian Ruder for useful discussions. MR is grateful to the Masason Foundation for their support.

References

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Alexandra Chronopoulou, Matthew E. Peters, and Jesse Dodge. 2021. Efficient hierarchical domain adaptation for pretrained language models.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

Lucio M. Dery, Paul Michel, Ameet Talwalkar, and Graham Neubig. 2021. Should we be pre-training? an argument for end-task aware training as an alternative. *arXiv preprint arXiv: Arxiv-2109.07437*.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2021. Demix layers: Disentangling domains for modular language modeling. *arXiv preprint arXiv:2108.05036*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep

learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

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

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2017. Knowledge adaptation: Teaching to adapt. *arXiv preprint arXiv: Arxiv-1702.02052*.

Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054, Melbourne, Australia. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Appendix

A.1 Hyperparameters

Computing Infrastructure	8 Volta 16GB GPUs

Hyperparameter	**Assignment**
architecture	GPT-2
tokens per sample	1024
batch size	64000
number of workers	8
learning rate	5e–5
clip norm	0.1
number of steps	1,000,000
save interval updates	1,000
validation interval	1,000
number of warmup steps	10,000
learning rate scheduler	polynomial decay
learning rate optimizer	Adam
Adam beta weights	(0.9, 0.99)
Adam epsilon	1e-6
weight decay	0.1

Table 8: Hyperparameters for finetuning in all settings.

A.2 Licenses

Our data sources have open licenses. Wikipedia has a Creative Commons Attribution-ShareAlike 3.0 Unported License and a S2ORC has a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

A.3 More examples of most transferred tokens

We give more examples of tokens transferred from the L1 S2ORC Computer Science (given its assumed familiarity to our audience) domain in the following table:

Transfer	**Example Tokens**
Computer Science→Computation and Language	lambda DCS, perplexity, Artetxe, Tacotron, Swayamdipta, Transformer, parallel, Socher, Gigaword, Lapata
Computer Science→Machine Learning	criterion, Ganchev, Ioffe, labeling, autoencoder, Hinton, hyperparameters
Computer Science→Art	Atheist, heroism, intellectuals, horrors, witchcraft, mourning, apostles
Computer Science→Technology and Applied Sciences	Sunderland, accounting, inventory, Libyan, bishop, ravaged, traffic

Table 9: More examples of most transferred tokens

A.4 All domains

We list all domains contained within dataset in Table 10.

S2ORC
cs.CE, cs.IT, cs.CG, cs.SI, cond-mat.quant-gas, math.SG, cs.SC, cs.CY, econ.GN, math.CO, cs.AR, cs.MS, cs.DC, q-bio.TO, cs.GR, physics.acc-ph, physics.geo-ph, math.RT, math.HO, cs.RO, q-bio.SC, math.QA, cs.NI, math.CA, cs.DS, astro-ph.GA, physics.atom-ph, math.CT, cs.CV, cond-mat.mtrl-sci, math.CV, math.AC, cond-mat.str-el, physics.comp-ph, cs.CC, math.FA, cond-mat.dis-nn, econ.TH, physics.gen-ph, physics.data-an, astro-ph.IM, q-bio.CB, math.LO, physics.ins-det, q-bio.BM, cs.LO, math.GR, physics.optics, cs.GT, math.AG, cs.NE, cs.SY, physics.bio-ph, physics.flu-dyn, cs.CL, math.MG, cs.AI, math.OC, nlin.CG, math.IT, stat.OT, math.OA, cond-mat.soft, Art, cs.GL, cs.PF, math.ST, physics.ao-ph, physics.plasm-ph, math.RA, physics.hist-ph, cs.PL, cs.MA, physics.chem-ph, physics.soc-ph, physics.med-ph, physics.ed-ph, stat.AP, stat.CO, math.DS, cs.DB, nlin.SI, q-bio.GN, physics.atm-clus, nlin.CD, astro-ph.CO, cs.CR, cond-mat.supr-con, cs.LG, math.KT, stat.ML, nlin.PS, q-bio.MN, cs.IR, math.GT, cs.SD, math.NA, cond-mat.other, math.NT, cs.FL, physics.pop-ph, cond-mat.stat-mech, math.GN, cs.DL, astro-ph.EP, q-bio.QM, cs.ET, q-bio.PE, cs.OH, Philosophy, physics.space-ph, econ.EM, physics.class-ph, cs.DM, cond-mat.mes-hall, stat.TH, cs.SE, astro-ph.HE, math.MP, nlin.AO, math.AP, q-bio.NC, q-bio.OT, astro-ph.SR, math.DG, math.AT, cs.MM, stat.ME, cs.OS, math.SP, physics.app-ph, cs.NA, math.PR, math.GM, cs.HC

Wikipedia
Culture and Humanities, Games and Toys, Mass media, Performing arts, Sports and Recreation, The arts and Entertainment, Visual arts, Further research tools and topics, Reference works, Exercise, Health science, Human medicine, Nutrition, Public health, Self care, By continent, By period, By region, Human activities, Impact of human activity, Fields of mathematics, Logic, Mathematics, Biology, Earth sciences, Nature, Physical sciences, Philosophy, Thinking, Allah, Belief systems, Major beliefs of the world, Social sciences, Society, Agriculture, Computing, Engineering, Transport

Table 10: All domains contained within M2D2