MSciNLI: A Diverse Benchmark for Scientific Natural Language Inference

Mobashir Sadat **Cornelia Caragea**
Computer Science
University of Illinois Chicago
msadat3@uic.edu cornelia@uic.edu

Abstract

The task of scientific Natural Language Inference (NLI) involves predicting the semantic relation between two sentences extracted from research articles. This task was recently proposed along with a new dataset called SciNLI derived from papers published in the computational linguistics domain. In this paper, we aim to introduce diversity in the scientific NLI task and present MSciNLI, a dataset containing 132,320 sentence pairs extracted from five new scientific domains. The availability of multiple domains makes it possible to study domain shift for scientific NLI. We establish strong baselines on MSciNLI by fine-tuning Pre-trained Language Models (PLMs) and prompting Large Language Models (LLMs). The highest Macro F1 scores of PLM and LLM baselines are 77.21% and 51.77%, respectively, illustrating that MSciNLI is challenging for both types of models. Furthermore, we show that domain shift degrades the performance of scientific NLI models which demonstrates the diverse characteristics of different domains in our dataset. Finally, we use both scientific NLI datasets in an intermediate task transfer learning setting and show that they can improve the performance of downstream tasks in the scientific domain. We make our dataset and code available on Github.[1]

1 Introduction

Natural Language Inference (NLI) (Bowman et al., 2015) or Textual Entailment is the task of recognizing the semantic relation between a pair of sentences where the first sentence is called premise and the second sentence is called hypothesis. Traditional NLI datasets such as SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018a), SICK (Marelli et al., 2014), and ANLI (Nie et al., 2019) classify the premise-hypothesis pairs into one of three classes indicating whether the hypothesis entails, contradicts or is neutral to the premise. These datasets have been used both as a benchmark for Natural Language Understanding (NLU) and to improve downstream tasks such as fact verification (Martín et al., 2022) and fake news detection (Sadeghi et al., 2022). In addition, they have aided in the advancement of representation learning (Conneau et al., 2017), transfer learning (Pruksachatkun et al., 2020), and multi-task learning (Liu et al., 2019a).

However, since the examples in these datasets are derived from non-specialized domains, e.g., image captions, they do not capture the unique linguistic characteristics of different specialized domains such as the scientific domain. More recently, Sadat and Caragea (2022b) introduced the task of scientific NLI along with the first dataset for this task named SciNLI, which contains 107,412 sentence pairs extracted exclusively from scientific papers related to computational linguistics published in the ACL anthology (Bird et al., 2008; Radev et al., 2009). To capture the inferences that frequently occur in scientific text, Sadat and Caragea (2022b) extended the three classes in traditional NLI to four classes for scientific NLI—ENTAILMENT, REASONING, CONTRASTING, and NEUTRAL. Since its introduction, SciNLI has gained great interest in the research community (Wang et al., 2022; Deka et al., 2022; Wu et al., 2023).

Despite introducing a challenging task and enabling the exploration of NLI with scientific text, SciNLI lacks the diversity to serve as a general purpose scientific NLI benchmark because it is limited to a single domain (ACL). Moreover, due to the unavailability of multiple domains, SciNLI is not suitable for studying domain adaptation and transfer learning on scientific NLI.

To this end, in this paper, we propose **MSciNLI**, a scientific NLI dataset containing 132,320 sentence pairs extracted from papers published in five different domains: "Hardware", "Networks", "Soft-

[1] https://github.com/msadat3/MSciNLI

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 1610–1629

ware & its Engineering", "Security & Privacy", and "NeurIPS." Similar to Sadat and Caragea (2022b), we use a distant supervision method that exploits the linking phrases between sentences in scientific papers to construct a large training set and directly use these potentially noisy sentence pairs during training. For the test and development sets, we manually annotate $4,000$ and $1,000$ examples, respectively, to create high quality evaluation data for scientific NLI.

We evaluate the difficulty of MSciNLI by experimenting with a BiLSTM based model. We then establish strong baselines on MSciNLI by a) fine-tuning four transformer based Pre-trained Language Models (PLMs): BERT (Devlin et al., 2019), SciBERT (Beltagy et al., 2019), RoBERTa (Liu et al., 2019b) and XLNet (Yang et al., 2019); and b) prompting two Large Language Models (LLMs) in both zero-shot and few-shot settings: Llama-2 (Touvron et al., 2023), and Mistral (Jiang et al., 2023). Furthermore, we provide a comprehensive investigation into the robustness of scientific NLI models by evaluating their performance under domain-shift at test time. Finally, we explore both SciNLI and MSciNLI in an intermediate task transfer learning setting (Pruksachatkun et al., 2020) to evaluate their usefulness in improving the performance of other downstream tasks.

Our key findings are: a) MSciNLI is more challenging than SciNLI; b) the best performing PLM baseline, which is based on RoBERTa, shows a Macro F1 of 77.21% on MSciNLI indicating the challenging nature of the task and a substantial headroom for improvement; c) the best performing LLM baseline with Llama-2 shows a Macro F1 of only 51.77% indicating that our dataset can be used to benchmark the NLU and complex reasoning capabilities of powerful LLMs; d) domain-shift at test time reduces the performance; and e) diversity in the scientific NLI datasets helps toimprove the performance of downstream tasks.

2 Related Work

Since the introduction of the NLI task, many datasets derived from different data sources have been made available. Datasets such as RTE (Dagan et al., 2006) and SICK (Marelli et al., 2014) were instrumental in the progress of NLI research in its earlier days. However, the training set sizes of these datasets are too small for large scale deep learning modeling. SNLI (Bowman et al., 2015) was introduced as a large dataset for NLI. SNLI contains 570K sentence pairs where premises are extracted from image captions and human crowdworkers were employed to write the hypotheses and assign the labels. While SNLI is significantly larger than all other prior datasets, due to the premises being extracted from a single source, it lacks the diversity to serve as a challenging and general purpose NLU benchmark. Consequently, Williams et al. (2018b) introduced MNLI containing 433K sentence pairs where the premises are extracted from a diverse number of sources such as face-to-face conversations, travel guides, and the 9/11 event. Apart from the premise sources, both SNLI and MNLI are constructed in a similar fashion and are the most popular NLI datasets in the recent years.

Other NLI datasets include QNLI (Wang et al., 2018)—derived from the SQuaD (Rajpurkar et al., 2016) question-answering dataset; XNLI (Conneau et al., 2018)—a cross lingual evaluation corpus derived by translating examples from MNLI; ANLI (Nie et al., 2020)—constructed in an iterative adversarial fashion to reduce spurious patterns where human annotators develop examples that can cause the model to make errors in each iteration; SciTail (Khot et al., 2018)—derived from a school level science question-answer corpus in which the sentence pairs are classified into two classes: entailment or not-entailment. These datasets have also seen wide applications both as NLU benchmarks and to improve other downstream NLP tasks. However, none of these datasets contains sentences from scientific text that is found in research articles. Moreover, the classes in these datasets are not sufficient to study the inter-sentence inferences and complexities that occur frequently in scientific text.

Thus, to capture both the particularities in scientific text and provide coverage to the frequently occurring inter-sentence semantic relations, Sadat and Caragea (2022b) introduced SciNLI. The sentence pairs in SciNLI were extracted from papers published in the ACL anthology (Radev et al., 2009) using distant supervision based on different linking phrases. Given that SciNLI was derived from a single data source (ACL), it also lacks the necessary diversity in the data. Therefore, with a similar motivation behind constructing MNLI—to extend SNLI to multiple domains, we propose MSciNLI, the first diverse benchmark for scientific NLI, to extend SciNLI to multiple domains. The availability of multiple domains in MSciNLI enables

Domain	First Sentence	Second Sentence	Class
NEURIPS	A number of psychological studies have suggested that our brains indeed perform causal inference as an ideal observer (e.g., [10, 12–14]).	~~However,~~ it has been challenging to come up with a simple and biologically plausible neural implementation for causal inference.	CONTRASTING
NETWORKS	Researchers found out that the inhomogeneity in the spatio-temporal distribution of the data traffic leads to extremely insufficient utilization of network resources.	~~Thus,~~ it is important to fundamentally understand this distribution to help us make better resource planning or introduce new management tools such as time-dependent pricing to reduce the congestion.	REASONING
HARDWARE	Scaling PCM in deep sub-micron regime faces non-negligible inter-cell thermal interference during programming, referred to as write disturbance (WD) phenomenon.	~~That is,~~ the heat generated for writing one cell may disseminate beyond this cell and disturb the resistance states of its neighboring cells.	ENTAILMENT
SECURITY & PRIVACY	Following Google's best practices for developing secure apps, the password database is saved in the app data folder, which should be accessible only to the app itself.	this defines a hierarchical relationship between domains where the bounded domain cannot have more permissions than its bounding domain (the parent).	NEUTRAL
SOFTWARE & ITS ENGINEERING	If the delete operation is complex, then it advances to the discovery mode after which it will advance to the cleanup mode.	~~On the other hand,~~ if it is simple, then it directly advances to the cleanup mode (and skips the discovery mode).	CONTRASTING

Table 1: Examples of sentence pairs from MSCINLI extracted from different domains. The linking phrases at the beginning of the second sentence (strikethrough text in the table) are deleted after extracting the sentence pairs and assigning the labels.

the evaluation of the models' generalization ability under domain shift.

Recently, LLMs have demonstrated near human performance in many NLP tasks including NLI. For example, Zhong et al. (2023) reported that ChatGPT[2] shows a zero-shot accuracy of 88% on RTE and 89.3% on the matched test set of MNLI. Thus, developing benchmark tasks and datasets which are challenging for even powerful LLMs is paramount. While the primary goal of our dataset is to introduce diversity in scientific NLI, because of the complex reasoning and inference required to predict the semantic relation between a pair of sentences from scientific text, it can serve as a challenging benchmark even for powerful LLMs.

3 MSciNLI: A Multi-Domain Scientific NLI Benchmark

In this section, we describe the data sources for MSCINLI, its construction process and statistics.

3.1 Data Sources

We derive MSCINLI from the papers published in four categories of the ACM digital library[3] — 'Hardware', 'Networks', 'Software and its Engineering', 'Security and Privacy' and the papers published in the NeurIPS[4] conference. Table 1 shows examples of sentence pairs extracted from our five domains. Further details on our data sources (e.g., publication years of the papers) are available in Appendix A.1.

[2] https://chat.openai.com/
[3] https://dl.acm.org/
[4] https://papers.nips.cc/

3.2 Data Extraction and Automatic Distant Supervision Labeling

We closely follow the data extraction and automatic labeling procedure based on distant supervision proposed by Sadat and Caragea (2022b). Specifically, we use linking phrases between sentences (e.g., "Therefore", "Thus", "In contrast", etc.) to automatically annotate a large (potentially noisy) training set with the NLI relations. The complete list of linking phrases and their mapping to the NLI relations are presented in Appendix A.2. The procedure is detailed below.

For the ENTAILMENT, CONTRASTING, and REASONING classes, we extract adjacent sentence pairs from the papers collected from our five domains such that the second sentence starts with a linking phrase. For each extracted sentence pair, the relation corresponding to the linking phrase at the beginning of the second sentence is assigned as its class label. For example, if the second sentence starts with 'Therefore' or 'Consequently', the example is labeled as REASONING. Note that the linking phrase is removed from the second sentence after assigning the label to prevent the models from predicting the label by simply learning a superficial correlation between the linking phrase and the label and without actually learning the semantic relation.

For the NEUTRAL class, we construct the sentence pairs by extracting both sentences in the pair from the same paper using three approaches as follows: a) two random sentences that do not begin with any linking phrase are paired together; b) a random sentence which does not begin with any

Domain	#Examples			#Words		'S' parser		Word	
	Train	Dev	Test	Prem.	Hyp.	Prem.	Hyp.	Overlap	Agrmt.
SciNLI (ACL)	101,412	2,000	4,000	27.38	25.93	96.8%	96.7%	30.06%	85.8%
Hardware	25,464	200	800	26.10	24.59	94.3%	94.5%	30.52%	84.6%
Networks	25,464	200	800	26.37	25.01	93.9%	93.7%	30.17%	90.5%
Software & its Engineering	25,464	200	800	25.80	24.51	93.9%	94.1%	29.83%	86.5%
Security & Privacy	25,464	200	800	26.14	24.50	94.0%	94.2%	29.91%	90.4%
NeurIPS	25,464	200	800	29.80	29.66	96.0%	95.1%	31.04%	88.5%
MSciNLI Overall	127,320	1,000	4,000	26.84	25.85	94.4%	94.3%	30.29%	88.0%

Table 2: Comparison of key statistics of MSciNLI with SciNLI.

linking phrase is chosen as the first sentence and is paired with the second sentence of a random pair that belongs to one of the other three classes; c) a random sentence which does not begin with any linking phrase is chosen as the second sentence and is paired with the first sentence of a random pair that belongs to one of the other three classes.

After extracting the sentence pairs for all four classes, we randomly split them at paper level into train, test and development sets (to ensure that the sentence pairs extracted from a certain paper end up in a single set). We directly use the automatically annotated examples for training the models. However, our use of distant supervision during the construction of the training set may introduce label noise when the relation between a pair of sentences is not accurately captured by the linking phrase. Therefore, to ensure a realistic evaluation, we employ human annotators to manually annotate the sentence pairs in the test and dev sets with one of the four scientific NLI relations as described below.

3.3 Multi-domain Scientific NLI Test and Development Set Creation

Three expert annotators (see Appendix A.3 for more details about the annotators and the instructions) are employed to annotate the test and dev sets of MSciNLI. Specifically, a random subset (balanced over the classes) of sentence pairs from the test and dev sets are given to the three annotators who are instructed to annotate their labels (the relation between the sentences) based only on the context available in the two sentences in each example. If the annotators are unable to determine the label based on the two sentences of a pair, they mark it as unclear. We assign a gold label to each example based on the majority vote from the annotators. In rare cases ($\approx 3\%$) where there is no consensus among the annotators for an example, we do not assign a gold label. The examples for which there is a match between the gold label and the automatically assigned label (based on linking phrases) are included in their respective split and the rest are filtered out.

For each domain, we continue sampling random subsets (without replacement) of examples and manually annotate them until we have at least 800 clean examples (200 from each class) in the test set and 200 clean examples (50 from each class) in the dev set. In total, we annotate $6,992$ examples (all domains combined), among which $6,153$ have an agreement between the gold label and the automatically assigned label. That is, the overall agreement rate for MSciNLI is 88.0%. Moreover, we find a Fleiss-k score of 70.51% for MSciNLI indicating substantial agreement among the annotators (Landis and Koch, 1977).

Data Balancing To ensure equal representation, the number of examples per class in each domain are downsampled to a size of 200 and 50 in the test and dev set, respectively. Consequently, we end up with a combined (over the domains) test and dev sets of 4000 and 1000 examples, respectively (balanced over the classes and domains). We balance the training set by using a similar procedure.

3.4 Data Statistics

We show a comparison of key statistics of our dataset with the SciNLI dataset in Table 2.

Dataset Size We can see that the total number of examples (<premise, hypothesis> pairs) in MSciNLI is higher than that in SciNLI, the only NLI dataset over scientific text. Moreover, each domain in MSciNLI has a large number of examples in the training set which enables exploration of NLI in-domain as well as across domains.

Sentence Parses Similar to SciNLI, we use the Stanford PCFG Parser (3.5.2) (Klein and Manning, 2003) to parse the sentences in our dataset. We can see in Table 2 that $\approx 94\%$ of the sentences in MSciNLI have an 'S' root showing that most sentences in our dataset are syntactically complete.

Token Overlap The percentage of word overlap between the premise and hypothesis in each pair in MSCINLI is also low and close to that of SCINLI as shown in Table 2. Thus, like SCINLI, our MSCINLI dataset is also less vulnerable to surface level lexical cues.

4 MSCINLI Evaluation

Our main experiments for evaluating MSCINLI consists of three stages. First, we evaluate its difficulty by experimenting with a BiLSTM model (§4.1). Next, we establish strong baselines on MSCINLI with four Pre-trained Language Models (PLMs) and two Large Language Models (LLMs), and compare them with human performance (§4.2). Finally, we analyze our best performing baseline by investigating its performance when it is fine-tuned on various subsets of the training set and its performance under domain shift (§4.3). Our implementation details are given in Appendix B. Additional experiments on the impact of dataset size and diversity in model training; performance of another LLM; spurious correlations (Gururangan et al., 2018); and class-wise performances of the baselines are shown in Appendix C.

4.1 Difficulty Evaluation

BiLSTM Model The architecture of this model (described in Appendix B) is similar to the BiLSTM model adopted by Williams et al. (2018a). We can see a comparison of the performance of this model on MSCINLI and SCINLI in Table 3. We observe the following:

MSCINLI is more challenging than SCINLI. We can see that the Macro F1 of the BiLSTM model for SCINLI is 61.12% whereas it is only 54.40% for MSCINLI (the model is trained on the combined MSCINLI training set). These results indicate that MSCINLI presents a broader range of challenges for the model compared with SCINLI, making the scientific NLI task more difficult.

4.2 Baselines

Here, we describe the baseline models for MSCINLI and discuss their performance.

4.2.1 PLM Baselines

We fine-tune the base variants of the following PLMs on the combined MSCINLI training set: **BERT** (Devlin et al., 2019); **SCIBERT** (Beltagy et al., 2019); (c) **ROBERTA** (Liu et al., 2019b); and (d) **XLNET** (Yang et al., 2019). We run each experiment with the PLM baselines three times with different random seeds and report the average and standard deviation of their domain-wise and overall Macro F1 scores in Table 4. Our findings are described below.

Dataset	F1	Acc
SCINLI (ACL)	61.12	61.32
MSCINLI		
-Hardware	53.61	53.87
-Networks	54.78	54.95
-Software & its Engineering	51.96	52.20
-Security & Privacy	52.18	52.62
-NeurIPS	59.19	59.41
-Overall	54.40	54.61

Table 3: The Macro F1 (%) and Accuracy (%) of the BiLSTM model on SCINLI and MSCINLI.

Domain specific pre-training helps improve the performance. We can see that SCIBERT shows a better performance than BERT in all domains. Note that SCIBERT does not address any weaknesses of BERT and is trained using the same procedure as BERT, except SCIBERT exclusively uses scientific text for pre-training whereas BERT is trained on the BookCorpus and Wikipedia. Thus, pre-training on scientific documents helps improve the performance of scientific NLI.

"Robust" pre-training leads to better performance. Both ROBERTA and XLNET are designed to address different weaknesses of BERT. ROBERTA focuses on optimizing the model in a more robust manner during pre-training while XLNET aims at incorporating auto-regressive nature of natural language without removing bi-directional context. Both of these models substantially outperform BERT in all domains and ROBERTA consistently outperforms XLNET. We can also observe that ROBERTA leads to even better performance compared with SciBERT in most cases.

4.2.2 LLM Baselines

We experiment with two LLMs as baselines for our dataset: (a) **LLAMA-2** (Touvron et al., 2023) and (b) **MISTRAL** (Jiang et al., 2023). More specifically, we use the *Llama-2-13b-chat-hf* and *Mistral-7B-Instruct-v0.1* variants of LLAMA-2 and MISTRAL, containing 13 billion and 7 billion parameters, respectively. Both of these models are chosen because of their success in many NLP tasks that require complex reasoning and problem solving (e.g., the MMLU benchmark (Hendrycks et al., 2021)).

MODEL	HARDWARE	NETWORKS	SWE	SECURITY	NEURIPS	OVERALL
BERT	72.89 ± 0.1	74.10 ± 1.3	71.37 ± 0.3	72.38 ± 2.5	75.46 ± 0.8	73.24 ± 0.8
SCIBERT	$\underline{75.91 \pm 0.1}$	$\mathbf{76.51 \pm 0.5}$	$\underline{75.28 \pm 1.1}$	$\underline{75.94 \pm 0.4}$	$\mathbf{78.78 \pm 0.1}$	$\underline{76.48 \pm 0.4}$
XLNET	75.59 ± 0.5	75.25 ± 0.1	73.98 ± 0.6	75.09 ± 0.8	77.64 ± 1.0	75.51 ± 0.3
ROBERTA	$\mathbf{77.79}^{\$} \pm \mathbf{0.2}$	$\underline{75.45 \pm 1.5}$	$\mathbf{77.10}^{\#} \pm \mathbf{0.7}$	$\mathbf{77.71}^{\$} \pm \mathbf{0.2}$	$\underline{78.04 \pm 0.8}$	$\mathbf{77.21}^{\#} \pm \mathbf{0.3}$

Table 4: Macro F1 scores (%) of the PLM baselines on different domains. Here, SWE: Software & its Engineering and SECURITY: Security & Privacy. $^{\#}$ and $^{\$}$ indicate statistically significant improvement by ROBERTA over XLNET and over both SCIBERT and XLNET, respectively according to a paired t-test with $p < 0.05$. Best performance is shown in **bold**, and second best is underlined.

MODEL	PROMPT	HARDWARE	NETWORKS	SWE	SECURITY	NEURIPS	OVERALL
LLAMA-2	PROMPT - 1_{zs}	20.31	21.34	19.77	21.36	18.92	20.41
	PROMPT - 2_{zs}	18.23	20.60	21.26	19.87	17.62	19.53
	PROMPT - 3_{zs}	30.27	32.64	30.49	30.16	27.58	30.36
	PROMPT - 1_{fs}	24.42	26.69	27.75	28.84	22.98	26.21
	PROMPT - 2_{fs}	37.49	38.27	34.25	36.32	35.26	36.39
	PROMPT - 3_{fs}	**53.41**	**51.38**	**50.54**	**52.75**	**50.38**	**51.77**
MISTRAL	PROMPT - 1_{zs}	21.72	21.48	19.87	22.77	21.36	21.43
	PROMPT - 2_{zs}	34.54	32.95	32.5	33.51	34.71	33.66
	PROMPT - 3_{zs}	34.64	33.68	34.14	36.00	34.78	35.00
	PROMPT - 1_{fs}	<u>48.21</u>	<u>42.50</u>	<u>45.68</u>	<u>44.40</u>	<u>45.98</u>	<u>45.49</u>
	PROMPT - 2_{fs}	39.83	38.71	35.45	36.70	36.30	37.55
	PROMPT - 3_{fs}	30.75	31.17	31.23	34.38	21.92	30.23

Table 5: Macro F1 scores (%) of the LLM baselines on different domains. Here, SWE: Software & its Engineering and SECURITY: Security & Privacy. Best performance is shown in **bold**, and second best is underlined.

We construct 3 multiple-choice question templates for the scientific NLI task to be used for prompting the LLMs:

- **PROMPT - 1**: this prompt asks the LLMs to predict the class given a sentence pair with the four class names as the choices.
- **PROMPT - 2**: to provide further context to the LLMs about the scientific NLI task, this prompt first defines the scientific NLI classes and then poses the question to predict the class with the class names as the choices.
- **PROMPT - 3**: instead of providing the definitions of the classes first and then asking a question with the class names as the choices, this prompt directly uses the class definitions as the choices.

The three prompt templates can be seen in Appendix D. We evaluate the performance of the LLMs in two settings: a) zero-shot: no input-output exemplars are shown to the model; b) few-shot: four input-human-annotated output exemplars (one for each class) are pre-pended to the prompt to evaluate the LLMs' in-context learning (Brown et al., 2020) ability for scientific NLI. The zero-shot and few-shot versions of each prompt i is denoted as PROMPT - i_{zs}, and PROMPT - i_{fs}, respectively.

We employ a greedy decoding strategy for all of our LLM based experiments and report the domain-wise and the overall Macro F1 scores of each experiment in Table 5. We find the following:

LLAMA-2 performs better than MISTRAL. We can see that LLAMA-2 with PROMPT - 3_{fs} shows the best performance among all of our LLMs with a Macro F1 of 51.77%. This is 6.28% higher than the best performance shown by MISTRAL with PROMPT - 1_{fs}. Thus, LLAMA-2 with its 13B parameters has more complex reasoning capability compared to MISTRAL with its 7B parameters.

Using class-definitions as choices in the prompt and the few-shot prompt variants improve the performance. We can see that the performance of both LLMs are generally better when we use PROMPT - 3. This indicates that using the class definitions as the potential choices in the multiple-choice question is more suitable for the models, resulting in better performance. We can also see that the few-shot variants of the prompts generally outperform their zero-shot counterparts. Thus, both LLMs are capable of in-context learning and providing few examples can boost their performance.

Scientific NLI is highly challenging for state-of-the-art LLMs Despite the promising few-shot performance, based on the results in Table 5, it is evident that the task of scientific NLI is highly challenging even for powerful LLMs. Therefore,

Method	Macro F1	Accuracy
RoBERTa	77.21 ± 0.30	77.42 ± 0.30
Llama-2	51.77 ± 0.00	51.10 ± 0.00
Human - E (est.)	89.33 ± 1.18	89.10 ± 1.10
Human - NE (est.)	79.78 ± 4.43	79.49 ± 4.84

Table 6: Comparison of *estimated* human expert and non-expert performances with RoBERTa and Llama-2 (with Prompt - 3) on the MSciNLI test set. Here, E: expert, NE: non-expert.

our dataset along with SciNLI can serve as a challenging evaluation benchmark for LLMs.

4.2.3 Human Performance

We hire three expert annotators (with relevant domain-specific background) and three non-expert annotators (with no background in any of the five domains) to evaluate the human performance on MSciNLI. Note that these expert and non-expert annotators are not involved in our dataset construction process (see Appendix A.3 for more details). Following other popular benchmarks (e.g., SuperGLUE (Wang et al., 2019)), we *estimate* the human performance by re-annotating a small randomly sampled subset of our test set. Each example in the subset is re-annotated by 3 expert and 3 non-expert annotators following the same data annotation procedure described in Section 3.3. We report the average and the standard deviation of the expert and non-expert performances (Macro F1) on this subset, and compare them with the best performing PLM baseline, RoBERTa, and the best performing LLM baseline, Llama-2 with Prompt - 3_{fs} in Table 6. Our findings are described below:

Experts outperform non-experts, and a substantial gap exists between model performance and human expert performance. As expected, expert annotators with the relevant domain-specific knowledge substantially outperform the non-expert annotators. Despite the lower performance by the non-experts (compared with experts), we can see that they still outperform our baselines. Furthermore, the performance by the experts is significantly higher than both RoBERTa and Llama-2. Therefore, there is a substantial headroom for improving the models' performance which can foster future research on scientific NLI.

4.3 Analysis

In this section, first, we diagnose the MSciNLI training set by fine-tuning separate models using different training subsets selected by performing data cartography (Swayamdipta et al., 2020) (§4.3.1). Next, we study the model behavior under domain shift at test time (§4.3.2). Finally, we perform cross-dataset experiments where we analyze the performance of models fine-tuned on SciNLI, MSciNLI, and their combination (§4.3.3). We choose our best performing baseline model, RoBERTa for these experiments.

Data Subset	Macro F1	Accuracy
100%	77.21 ± 0.30	77.42 ± 0.30
33% *easy-to-learn*	$73.71^{*} \pm 1.40$	$73.74^{*} \pm 1.43$
33% *hard-to-learn*	$34.11^{*} \pm 5.65$	$37.99^{*} \pm 1.53$
33% *ambiguous*	$75.65^{*} \pm 0.27$	$75.57^{*} \pm 0.26$
100%− top 25% *hard*	76.60 ± 0.65	76.64 ± 0.66
100%− top 5% *hard*	77.47 ± 0.23	77.44 ± 0.28

Table 7: The Macro F1 (%) and Accuracy (%) of RoBERTa fine-tuned on different subsets of MSciNLI training set. * indicates a statistically significant difference with the performance of the model trained on 100% data according to a paired t-test with $p < 0.05$.

4.3.1 Data Cartography Experiments

We perform a data cartography of MSciNLI to characterize each example in the training set using two metrics — *confidence* and *variability*. Based on this characterization, inspired by (Swayamdipta et al., 2020), first, we fine-tune three different RoBERTa models using the following subsets of the training set: 1) 33% *easy-to-learn* — examples with high confidence; 2) 33% *hard-to-learn* — examples with low confidence; 3) 33% *ambiguous* — examples with high variability (the detailed method used for selecting these subsets of training examples is available in Appendix E). In addition, to further understand the effect of *hard-to-learn* examples in model training, we fine-tune two other models using the full training set **minus** — 1) top 25% *hard-to-learn* (25% examples with lowest confidence) and 2) top 5% *hard-to-learn* examples (5% examples with lowest confidence), denoted as '100%− top 25% *hard*', and '100%− top 5% *hard*', respectively. The results are shown in Table 7. We find the following.

Ambiguous examples yield stronger models while the full training set yields better performance. We can see that the model fine-tuned on the 33% *ambiguous* examples shows the best performance among the 33% subsets. Therefore, 'ambiguousness' in the training examples helps train stronger scientific NLI models. Despite the strong performance shown by 33% *ambiguous*, its Macro F1 is still lower than the 100% of the training set. Furthermore, although 33% *hard-to-learn* shows a poor performance (34.11% in Macro F1), remov-

Train \ Test	HARDWARE	NETWORKS	SWE	SECURITY	NEURIPS	ACL
HARDWARE	74.93 ± 1.4	73.11 ± 1.2	74.24 ± 0.2	72.98 ± 2.3	73.97 ± 0.7	72.40 ± 0.8
NETWORKS	75.04 ± 1.3	73.31 ± 1.7	73.29 ± 0.5	73.44 ± 1.0	74.61 ± 1.1	72.72 ± 1.0
SWE	73.60 ± 1.1	71.25 ± 0.8	74.44 ± 0.5	73.24 ± 1.4	75.31 ± 1.4	73.06 ± 1.8
SECURITY	72.69 ± 2.5	70.94 ± 1.8	74.14 ± 1.7	74.45 ± 2.5	73.12 ± 1.6	72.85 ± 1.3
NEURIPS	73.64 ± 0.8	71.94 ± 1.1	71.76 ± 1.2	71.62 ± 0.8	76.02 ± 1.1	74.15 ± 0.8
ACL - SMALL	74.29 ± 0.1	71.81 ± 0.8	73.44 ± 1.4	73.38 ± 1.1	74.95 ± 1.9	75.30 ± 0.5

Table 8: ID and OOD Macro F1 (%) of RoBERTa models trained on different domains. ID performance is shown in gray. Here, SWE: SOFTWARE & ITS ENGINEERING and SECURITY: SECURITY & PRIVACY.

Tr \ Te	SciNLI	MSciNLI	MSciNLI+
SciNLI	78.08	75.19	76.63
MSciNLI	76.74	77.21	76.95
MSciNLI+ (S)	77.78	77.37	77.54
MSciNLI+	**79.48**	**78.07**	**78.76**

Table 9: Macro F1 scores of cross dataset experiments with RoBERTa. Here, Tr: Train, Te: Test.

ing a percentage of them (e.g., 25%, and 5% in the bottom block of Table 7) from the 100% of the training set does not result in any statistically significant difference in performance compared with 100%. Therefore, all examples in the training set are useful for training the most optimal model.

4.3.2 Out-of-domain Experiments

Here, we train RoBERTa on one domain and test it on another domain (out-of-domain) and contrast it with the RoBERTa trained and tested on the same domain (in-domain). In addition to the five domains in MSciNLI, we also experiment with the ACL domain from SciNLI. For a fair comparison with the other domains, we downsample the training set from SciNLI to the same size as that of the other domains and denote it as ACL - SMALL. Both in-domain (ID) and out-of-domain (OOD) results are shown in Table 8. Our findings are described below:

The domain shift reduces the performance. In general, for each domain, the ID model shows a higher performance than their OOD counterparts (see each column in Table 8). For example, the model fine-tuned on the NEURIPS training set shows a Macro F1 of 76.02% when it is tested on NEURIPS as well. The performance sees a decline when the models trained on other domains are tested on NEURIPS (e.g., 74.61% with the NETWORKS model). This indicates that the sentence pairs in each domain exhibit unique linguistic characteristics which are better captured by a model trained on in-domain data.

4.3.3 Cross-dataset Experiments

For the cross-dataset experiments, we train four separate RoBERTa models on: 1) SciNLI, 2) MSciNLI, 3) MSciNLI+ (S) - a combination of MSciNLI and ACL - SMALL, and 4) MSciNLI+ - a combination of MSciNLI and SciNLI. All four models are then evaluated using the separate SciNLI and MSciNLI test sets, and their combination i.e., the MSciNLI+ test set. The results are reported in Table 9. We also evaluate the models on the domain-wise test sets, and general domain NLI datasets, and report the results in Appendix F.

Diverse training data leads to robust models. The performance sees a decline for both SciNLI and MSciNLI under 'dataset-shift.' However, the model fine-tuned with SciNLI shows a higher drop in performance compared with the model fine-tuned with MSciNLI in the out-of-dataset setting. Specifically, the out-of-dataset Macro F1 of the model fine-tuned with SciNLI (when it is tested on MSciNLI) drops by 2.02% from the in-dataset performance of MSciNLI (77.21%). In contrast, the out-of-dataset Macro F1 of the model fine-tuned with MSciNLI (when it is tested on SciNLI) drops by only 1.34% from the in-dataset performance of SciNLI (78.08%). This indicates that the diversity in the data can train more robust scientific NLI models with stronger generalization capabilities.

Combining the datasets yields the best performance. The best performance for both datasets and their combination is seen when the model is fine-tuned on MSciNLI+. Therefore, fine-tuning the model on a larger training set containing diverse examples yields better performance. We can see that the models trained on MSciNLI+ (S) show a lower performance than those trained on MSciNLI+. This is because MSciNLI+ (S) is smaller in size than MSciNLI+. However, due to the additional diversity introduced by the ACL

Dataset	Intermediate training data				
	None	MSciNLI+ (MLM)	MNLI	SciNLI	MSciNLI+
SciHTC	52.59	48.95	51.83	51.83	**53.47**
Paper Field	73.66	73.46	73.64	73.61	**74.09**
ACL-Arc	69.57	63.95	59.73	68.52	**73.04**

Table 10: Macro F1 (%) of RoBERTa with intermediate task transfer using different NLI datasets.

domain, MSciNLI+ (S) consistently outperforms MSciNLI. Thus, the benefit of combining the datasets holds for MSciNLI+ (S) as well.

5 Scientific NLI as an Intermediate Task

Research (Martín et al., 2022; Sadeghi et al., 2022) has shown that traditional NLI datasets (e.g., SNLI, MNLI) can aid in improving the performance of downstream NLP tasks. While the SciNLI dataset has already been used to improve sentence representation (Deka et al., 2022), it was used in conjunction with the traditional NLI datasets. In this section, we investigate whether the scientific NLI datasets by themselves can aid in improving the performance of downstream tasks in an intermediate task transfer setting (Pruksachatkun et al., 2020).

To this end, first, a RoBERTa model (out-of-the-box pre-trained with a dynamic MLM objective) is fine-tuned on the downstream tasks. Next, we perform intermediate training of four separate out-of-the-box RoBERTa models with the following approaches before fine-tuning them on the downstream tasks: **1)** with a self-supervised dynamic MLM objective (with no information of the NLI classes) on MSciNLI+; **2)** with a supervised NLI objective using MNLI; **3)** with a supervised scientific NLI objective using SciNLI; **4)** with a supervised scientific NLI objective using MSciNLI+.

We experiment with the following downstream tasks: **SciHTC** (Sadat and Caragea, 2022a), **Paper Field** (Beltagy et al., 2019), and **ACL-Arc** (Jurgens et al., 2018). SciHTC and Paper Field are topic classification datasets for scientific papers and ACL-Arc is a citation intent classification dataset. Details on these tasks and their labels are in Appendix G. The results for each downstream task are presented in Table 10. We find that:

Scientific NLI can aid in improving the performance of downstream tasks. As we can see from the table, intermediate training with an unsupervised MLM objective on MSciNLI+ (MSciNLI+ (MLM) in the table) fails to improve the performance of the downstream tasks over the models which are fine-tuned without any intermediate training. In contrast, supervised intermediate training on MSciNLI+ improves the performance of all datasets over all other models. This indicates that training a model further on the scientific NLI task can learn better and more relevant representations for the downstream tasks in the scientific domain. We can also see that supervised intermediate training on MNLI fails to show improvement for any of the downstream tasks. This illustrates the need for NLI datasets capturing the unique linguistic properties of scientific text (e.g., SciNLI and MSciNLI) in order to improve the performance of downstream tasks in this domain. Furthermore, we observe that intermediate training with a scientific NLI objective only using SciNLI fails to improve the performance of the downstream tasks. Therefore, while intermediate training with a scientific NLI objective can aid in improving the performance of downstream tasks, the diversity in the data is essential.

6 Conclusion & Future Directions

We introduce a diverse scientific NLI benchmark, MSciNLI derived from five scientific domains. We show that MSciNLI is more difficult to classify than the only other related dataset, SciNLI. We establish strong baselines on MSciNLI and find that our dataset is challenging for both PLMs and powerful LLMs. Furthermore, we provide a comprehensive investigation into the performance of scientific NLI models under domain-shift at test time and their usage in downstream NLP tasks. In the future, we will develop methods to improve the construction of prompts that enable better reasoning and inference capabilities of LLMs.

Acknowledgements

This research is supported by NSF CAREER award 1802358 and NSF IIS award 2107518. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of NSF. We thank our anonymous reviewers for their constructive feedback, which helped improve the quality of our paper.

Limitations

From our experiments, we can see that the performance of the LLMs is low (best performing Macro F1 is 51.77%) on MSciNLI, which shows a lot of room for future improvement. The design of the prompts have a high impact on the performance as we can see from the results, thus, further exploration of other prompting strategies can potentially improve the performance further. In the future, we will focus on the design of other prompts to boost the performance of LLMs in scientific NLI.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Steven Bird, Robert Dale, Bonnie J. Dorr, Bryan Gibson, Mark T. Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir R. Radev, and Yee Fan Tan. 2008. The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In *Proc. of the 6th International Conference on Language Resources and Evaluation Conference (LREC'08)*, pages 1755–1759.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Pritam Deka, Anna Jurek-Loughrey, et al. 2022. Evidence extraction to validate medical claims in fake news detection. In *International Conference on Health Information Science*, pages 3–15. Springer.

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

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *AAAI*, volume 17, pages 41–42.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

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

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Alejandro Martín, Javier Huertas-Tato, Álvaro Huertas-García, Guillermo Villar-Rodríguez, and David Camacho. 2022. Facter-check: Semi-automated fact-checking through semantic similarity and natural language inference. *Knowledge-Based Systems*, 251:109265.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247, Online. Association for Computational Linguistics.

Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. 2009. The ACL Anthology network. In *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries (NLPIR4DL)*, pages 54–61, Suntec City, Singapore. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Mobashir Sadat and Cornelia Caragea. 2022a. Hierarchical multi-label classification of scientific documents. In *Proceedings of The 2022 Conference on Empirical Methods in Natural Language Processing (Volume 1: Long Papers)*, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mobashir Sadat and Cornelia Caragea. 2022b. SciNLI: A corpus for natural language inference on scientific text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7399–7409, Dublin, Ireland. Association for Computational Linguistics.

Fariba Sadeghi, Amir Jalaly Bidgoly, and Hossein Amirkhani. 2022. Fake news detection on social media using a natural language inference approach. *Multimedia Tools and Applications*, pages 1–21.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Chenglin Wang, Yucheng Zhou, Guodong Long, Xiaodong Wang, and Xiaowei Xu. 2022. Unsu-

pervised knowledge graph construction and event-centric knowledge infusion for scientific nli. *arXiv preprint arXiv:2210.15248*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018a. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018b. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Jinxuan Wu, Wenhan Chao, Xian Zhou, and Zhunchen Luo. 2023. Characterizing and verifying scientific claims: Qualitative causal structure is all you need. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13428–13439, Singapore. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint arXiv:2302.10198*.

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

Class	Linking Phrases
CONTRASTING	'However', 'On the other hand', 'In contrast', 'On the contrary'
REASONING	'Therefore', 'Thus', 'Consequently', 'As a result', 'As a consequence', 'From here, we can infer'
ENTAILMENT	'Specifically', 'Precisely', 'In particular', 'Particularly', 'That is', 'In other words'

Table 11: Linking phrases used to extract sentence pairs and their corresponding classes.

A Additional Dataset Details

A.1 More Details about Data Sources

To construct our dataset, for all five domains, we choose papers published after the year 2000. In particular, the sentence pairs for the training set of NEURIPS are extracted from papers published between 2000 and 2018 and the test and development sets are derived from the papers published in 2019. The training sets for the four ACM domains—HARDWARE, NETWORKS, SOFTWARE & ITS ENGINEERING, and SECURITY & PRIVACY are constructed from the papers published between 2000 and 2014. The sentence pairs extracted from the papers published between 2015 and 2017 are used to create the test and development sets for each domain.

A.2 List of Linking Phrases

To construct MSCINLI, we use the same list of linking phrases and their corresponding classes as SCINLI. Table 11 shows the linking phrases and their classes.

A.3 Details about Annotators and Annotation Instructions

In this section, we provide the details about the annotators we hired for constructing the test and development sets of MSCINLI (§A.3.1), and for evaluating the human performance (§A.3.2).

A.3.1 Annotators for constructing the test and development sets.

For constructing the MSCINLI development and test sets (in Section 3.3), we hired 7 computer science undergraduate students as research interns at our institution who were compensated in an hourly basis by $15/hour. Each annotator was trained with several pilot iterations before they started the final annotations for constructing the dataset. Moreover, out of the 7 students that we initially hired, only 3 were selected as the final annotators based on their performance during training to ensure a high quality of labels in our dataset.

The training phase of the students consists of 3 iterations. At each iteration, all 7 students were given a pilot batch and were instructed to predict the label based on the two sentences in each sample. We provide feedback to all students at the end of each iteration. In addition to the hired students, an author of this paper also annotated the examples in the third training iteration. 3 students were then selected as the final annotators who have the top three agreement rates with the author (79.3%, 78.6%, 75.8%). Once the annotators are trained, they start the final annotations to create the benchmark evaluation set of MSCINLI.

Note that the annotators are instructed to label each pair of sentences based on the four scientific NLI relations and not based on what could be a possibly good linking phrase between them. This annotation instruction ensures that the scientific NLI task formulation remains the same as the traditional NLI task—predicting the semantic relation between a pair of sentences.

A.3.2 Annotators for evaluating human performance.

For evaluating the human performance on MSCINLI (in Section 4.2.3), we hire expert as well as non-expert annotators via a crowd-sourcing platform called COGITO.[5] We ensured that none of the annotators for evaluating the human performance is involved with the construction of MSCINLI at any capacity. We distinguish between the expert and non-expert annotators based on whether they have the relevant background on the scientific domains in MSCINLI. Both sets of annotators are trained in the same fashion as the annotators who helped construct the test and development sets (described in the previous paragraphs). Both expert and non-expert annotators are paid at a rate of $0.6/sample.

A.4 Class-wise Agreement Rate

The total number of annotated examples while constructing the test and development sets (in Section 3.3) for each class and the agreement rate between the gold label and the automatically assigned label based on linking phrases can be seen in Table 12.

[5] https://www.cogitotech.com/

Class	#Annotated	Agreement
Contrasting	1748	92.9%
Reasoning	1748	83.1%
Entailment	1748	79.2%
Neutral	1748	96.7%
Overall	6992	88.0%

Table 12: Number of manually annotated examples and the agreement rate between the gold labels and automatically assigned labels for each class.

	HW	NW	SWE	SEC	NIPS	ACL
HW	1					
NW	0.94	1				
SWE	0.95	0.95	1			
SEC	0.93	0.96	0.97	1		
NIPS	0.70	0.63	0.63	0.61	1	
ACL	0.76	0.69	0.71	0.68	0.81	1

Table 13: Pair-wise cosine similarities of the probability distributions of the vocabulary of RoBERTa-base over domain-wise training sets. Here, HW:HARDWARE, NW:NETWORKS, SWE: SOFTWARE & ITS ENGINEERING., SEC: SECURITY & PRIVACY, NIPS: NEURIPS, and ACL: data from SCINLI.

A.5 Difference/Closeness of the Domains

We quantify the differences/closeness of the domains in MSCINLI and the computational linguistic domain from SCINLI as the pairwise cosine similarities of the probability distributions of the RoBERTa-base[6] vocabulary over each domain. The cosine similarities are reported in Table 13. We can see that the first four domains in the Table show a high similarity among them. Recall that the sentence pairs for all of these four domains are extracted from papers published in the ACM digital library. The high cosine similarities illustrate that the writing style and the vocabulary used in these domains are similar. In contrast, the cosine similarity of NEURIPS is the lowest with all other four domains in MSCINLI. Therefore, the vocabulary and the writing style in the papers published in NEURIPS differs substantially from the other four domains. Furthermore, it can be seen that the similarity between ACL and the five domains in MSCINLI is low, which illustrates that our dataset indeed diversifies the task of scientific NLI.

B Implementation Details

All of our experiments are implemented using PyTorch.[7] The details are provided below.

BILSTM baseline Two separate BiLSTM layers are used to get the sentence level representations of the two sentences in each pair. The token embeddings of each sentence are sent through the respective BiLSTM layer and then the output hidden states are averaged to get the sentence level representations. The context vector S_c is derived by concatenating the sentence level representations, their element-wise multiplication and difference. S_c is projected with a weight matrix $\mathbf{W} \in \mathbb{R}^{d \times 4}$ by using a linear layer with softmax to predict the class.

Each BiLSTM layer is equipped with 300D Glove (Pennington et al., 2014) embeddings which are allowed to be updated during training. The hidden state size for both BiLSTM layers is set at 300. The models are trained for 30 epochs with early stopping where we set the patience to be 10. The Macro F1 of the development score in every epoch is used as the stopping criteria. We use a cross-entropy loss and Adam optimizer (Kingma and Ba, 2014) to optimize the model parameters. The min-batch size and learning rate are set at 64 and 0.001, respectively.

PLM baselines The details of our pre-trained models are described as follows: (a) **BERT** (Devlin et al., 2019) - pre-trained by masked language modeling (MLM) and Next Sentence Prediction (NSP) objectives on BookCorpus (Zhu et al., 2015) and Wikipedia; (b) **SCIBERT** (Beltagy et al., 2019) - pre-trained using the same objectives as BERT but using scientific text exclusively as the pre-training data; (c) **ROBERTA** (Liu et al., 2019b) - an extension of BERT which uses a variation of MLM where different words are masked in each epoch dynamically (unlike static masking in standard MLM). It is also trained on larger amount of text, larger mini-batch size and larger number of epochs compared to BERT; and (d) **XLNET** (Yang et al., 2019) - pre-trained with a "Permutation Language Modeling" objective instead of MLM to provide bi-directional context to the model while being auto-regressive.

For these PLM baselines, the two sentences in each example are concatenated with a `[SEP]` token between them to be used as the input and the hidden representation embedded in the `[CLS]` token is then projected with a weight matrix $\mathbf{W} \in \mathbb{R}^{d \times 4}$. Finally, we use softmax on the projected representation to get the probability distribution over the four classes. The class with the maximum

[6] https://huggingface.co/roberta-base
[7] https://pytorch.org/

	HARDWARE	NETWORKS	SWE	SECURITY	NEURIPS	OVERALL
DOMAIN-WISE						
BERT	68.63 ± 1.4	67.69 ± 1.3	65.75 ± 0.8	67.04 ± 2.9	70.48 ± 1.6	-
SCIBERT	72.76 ± 0.9	72.97 ± 1.4	72.43 ± 0.8	72.45 ± 1.4	76.14 ± 0.5	-
XLNET	72.88 ± 0.6	68.85 ± 2.2	71.52 ± 2.1	71.60 ± 0.8	72.96 ± 0.8	-
ROBERTA	74.93 ± 1.4	73.31 ± 1.7	$74.44^{*} \pm 0.5$	74.45 ± 2.2	$76.02^{\#} \pm 1.1$	-
MERGED - SMALL						
BERT	69.36 ± 0.5	68.08 ± 1.3	66.61 ± 0.1	67.66 ± 1.4	71.62 ± 0.1	68.67 ± 0.3
SCIBERT	72.95 ± 0.3	72.88 ± 1.1	72.66 ± 0.2	72.37 ± 1.5	74.96 ± 1.6	73.17 ± 0.7
XLNET	72.87 ± 1.7	71.03 ± 1.8	72.21 ± 1.7	70.90 ± 0.8	73.45 ± 0.7	71.96 ± 1.2
ROBERTA	$75.06^{*} \pm 0.7$	73.20 ± 1.1	$74.49^{*} \pm 0.4$	$73.73^{\#} \pm 1.1$	75.75 ± 1.6	74.47 ± 0.8
MERGED - LARGE						
BERT	72.89 ± 0.1	74.10 ± 1.3	71.37 ± 0.3	72.38 ± 2.5	75.46 ± 0.8	73.24 ± 0.8
SCIBERT	75.91 ± 0.1	76.51 ± 0.5	75.28 ± 1.1	75.94 ± 0.4	78.78 ± 0.1	76.48 ± 0.4
XLNET	75.59 ± 0.5	75.25 ± 0.1	73.98 ± 0.6	75.09 ± 0.8	77.64 ± 1.0	75.51 ± 0.3
ROBERTA	$77.79^{\$} \pm 0.2$	75.45 ± 1.5	$77.10^{\#} \pm 0.7$	$77.71^{\$} \pm 0.2$	78.04 ± 0.8	$77.21^{\#} \pm 0.3$

Table 14: Macro F1 scores (%) of our PLM baselines on different domains trained in different settings. Here, SWE: Software & its Engineering and SECURITY: Security & Privacy. All MERGED - SMALL scores are statistically indistinguishable from their DOMAIN-WISE counterparts according to a paired t-test with $p < 0.05$. All MERGED-LARGE scores show statistically significant improvement over MERGED-SMALL. *, $^{\#}$, $^{\$}$ indicate statistically significant improvement by ROBERTA over SCIBERT, XLNET, and both SCIBERT and XLNET, respectively.

probability is predicted as the label for each input pair.

Each PLM baseline is fine-tuned for 10 epochs with early stopping using the huggingface[8] library. The patience for early stopping is set at 2. The learning rate and the mini-batch size is set at $2e-5$, and 64, respectively. We use a cross-entropy loss and Adam optimizer (Kingma and Ba, 2014) to optimize the model parameters.

LLM baselines We make use of the prompt templates described in Section 4.2.2 to construct the inputs to the LLM baselines. Similar to the PLM baselines, we conduct our experiments for LLM baselines using the huggingface library. We employ a greedy decoding strategy with a maximum generated token count to be 40. Generally, instead of only providing the answer to our multiple-choice question, the LLMs generates a more verbose response with the answer contained in it. We manually examine the responses for each prompt by each LLM and develop scripts to extract the correct answer with rule-based approaches.

Computational Cost The BiLSTM and PLM experiments are conducted on a single NVIDIA RTX A5000 GPU. The BiLSTM model was trained in $\approx$ 30 minutes. The time needed to fine-tune each PLM baseline on the full MSCINLI training set using a single GPU is $\approx$ 2 hours. The inference by the LLM baselines is conducted using two NVIDIA RTX A5000 GPUs and it took $\approx$ 3 hours on average for each experiment.

[8] https://huggingface.co

C Additional Results

C.1 Domain-wise vs Merged

In addition to fine-tuning on the combined MSCINLI training set (127,320 examples) in Section 4.2.1, we experiment with the PLM baselines in two other settings: DOMAIN-WISE and MERGED-SMALL (see the description of these settings below) and compare their performance with the model fine-tuned on the combined MSCINLI training set denoted as MERGED-LARGE. The motivation behind these experiments is two-fold: a) to understand the impact of diversity of examples in model training (DOMAIN-WISE vs MERGED-SMALL) (when the models are trained on data from a single domain vs data from diverse domains—but all being trained on the same training set size); and b) to understand the impact of training set size (MERGED-SMALL vs MERGED-LARGE). In the DOMAIN-WISE setting, we train and evaluate separate models for each domain using the data from the respective domain. For the MERGED-SMALL setting, we randomly down-sample the training set of each domain to 5092 examples (class-balanced) before combining them to ensure that the total size of the merged set MERGED-SMALL is similar to the DOMAIN-WISE training set size ($\approx$ 25,464). We combine the downsampled data from all domains and train *a single model* using the merged

Model	Contrasting	Reasoning	Entailment	Neutral	Macro Ave.
Precision					
RoBERTa	74.24	74.61	76.38	85.46	77.67
Llama-2	56.35	36.19	49.76	71.96	53.57
Recall					
RoBERTa	87.93	67.36	74.63	79.36	77.32
Llama-2	54.50	50.10	42.30	57.50	51.10
F1					
RoBERTa	80.43	70.76	75.44	82.22	77.21
Llama-2	55.41	42.03	45.72	63.92	51.77

Table 15: Class-wise precision (%), recall (%), F1 (%), and their macro averages (%) of our best performing PLM and LLM baselines on MSciNLI.

data. This model is then evaluated on the test set of each domain and the combined MSciNLI test set. The Merged-large setting corresponds to the combined training set of MSciNLI of 127, 320 examples.

We run each experiment three times and report the average and standard deviation of the Macro F1 score of the models in the three settings in Table 14. We find the following:

Training models on diverse data is more optimal. We can see that each model trained in the Merged - small setting shows similar performance as their Domain-wise counterparts. Moreover, in some cases (e.g., for BERT for all domains, XLNet for NeurIPS), the Merged - small models outperform the Domain-wise models. Recall that the size of the Merged-small training set is the same as each of the Domain-wise training sets. Since we train separate models for each domain in the Domain-wise setting, it is five times computationally more expensive than the Merged - small setting where a single model is trained for all domains. Therefore, training a single model on diverse data can reduce the computational cost without compromising model performance resulting in a more optimal approach.

More data leads to better performance. Next we compare the performance of the model fine-tuned on Merged-small with its Merged-large counterpart. The results show that Merged - large models consistently outperform the Merged - small models by a substantial margin. Therefore, the performance on our dataset improves with the the increase of dataset size.

C.2 Class-wise Performances

We evaluate the class-wise performance of our best performing PLM baseline—RoBERTa trained on the combined MSciNLI training set, and our best performing LLM baseline—Llama-2 with Prompt - 3 in the few-shot setting. The results are reported in Table 15.

As we can see, both models show a better performance for the Contrasting and the Neutral classes, and they struggle more for the Reasoning, and Entailment classes. However, even the Contrasting and the Neutral classes are still challenging for the models with substantial headroom for improvement.

C.3 Another LLM Baseline - GPT-NeoX

In addition to the LLMs that we explore in Section 4.2, we also experiment with the GPT-NeoXT-Chat-Base-20B[9] variant of the GPT-NeoX model. However, despite being much larger in size than the Llama-2 and Mistral baselines (20 billion parameters vs 13B and 7B, respectively), GPT-NeoX failed to show any promising performance (for the same three prompts used in the paper). We report the performance of these baselines in Table 16. We can see that the best performance for GPT-NeoX is shown by Prompt - 1_{zs} with an overall Macro F1 of only 22.14%. Moreover, none of the few-shot versions of the prompts shows any meaningful performance for this model (10.00% in Macro F1 with four labels in total means that the model always predicts the same label). In our future work, we will focus on the designing of other prompts that can improve the performance of the LLMs.

C.4 Only-Second-Sentence Baseline

To evaluate the degree of spurious correlations (Gururangan et al., 2018) that may exist in MSciNLI, we experiment with *only-second-sentence* models. Specifically, we fine-tune both RoBERTa

[9] https://huggingface.co/togethercomputer/GPT-NeoXT-Chat-Base-20B

	HARDWARE	NETWORKS	SWE	SECURITY	NEURIPS	OVERALL
GPT-NEOXT-CHAT						
PROMPT - 1_{zs}	17.84	19.17	20.19	18.04	16.99	18.49
PROMPT - 2_{zs}	20.48	18.28	20.67	21.62	27.56	22.14
PROMPT - 3_{zs}	12.69	15.30	13.63	13.90	14.66	14.12
PROMPT - 1_{fs}	10.00	10.00	10.00	10.00	10.00	10.00
PROMPT - 2_{fs}	10.00	10.00	10.00	10.00	10.00	10.00
PROMPT - 3_{fs}	10.00	10.00	10.00	10.00	10.00	10.00

Table 16: Macro F1 scores (%) of our GPT-NEOX baseline on different domains. Here, SWE: Software & its Engineering and SECURITY: Security & Privacy.

Model		SCINLI F1	SCINLI Acc
RoBERTa	BOTH SENTENCES	77.21	77.20
	ONLY 2^{nd} SENTENCE	52.55	53.55
SciBERT	BOTH SENTENCES	76.48	76.46
	ONLY 2^{nd} SENTENCE	53.14	53.65

Table 17: Performance comparison on MSCINLI when both sentences are concatenated vs. when only second sentence is used as the input.

and SCIBERT where only the second sentence is used as the input. A comparison between the *only-second-sentence* models and the models using both sentences can be seen in Table 17. The results show that the performance decreases by a large margin when only the second sentence is used as the input. Therefore, the amount of spurious correlation in MSCINLI is smaller compared with other existing NLI datasets (e.g., SNLI (Bowman et al., 2015)) and the models need to learn the semantic relation between the sentences in each pair in order to perform well.

However, given that the performance of the *only-second-sentence* models are much higher than chance (25%), we believe there are still some degree of spurious patterns in MSCINLI. In our future work, we will explore methods to identify and reduce the degree of spurious patterns in scientific NLI.

D Prompts for LLMs

The zero-shot versions of the three prompt templates that we construct for LLMs can be seen in Table 18. For the few-shot versions of the prompts, we pre-pend four input-human annotated output exemplars (one for each class) to each prompt. Note that the <*human*> and <*bot*> tags in the prompts in the Table are replaced with the relevant tags for each LLM (e.g., `[INST]`).

E Training Dynamics Based Data Selection

The *easy/hard/ambiguous* subsets of the training data are selected based on their training dynamics (Swayamdipta et al., 2020). Specifically, the training dynamics of each example is defined in the form of three metrics—*confidence*, *variability*, and *correctness* during training a classifier. These metrics are used to plot the examples in a data map to perform a visual analysis. The aforementioned three subsets of the training set are then selected based on *confidence* and *variability*. In this section, we define these metrics, perform a data cartography of MSCINLI, and describe the method to select the subsets used in Section 4.3.1.

E.1 Metrics Definitions

The *confidence* of each example is defined as the average of the probability predicted by a classifier for its label over the training epochs. That is, for a training example X_i and its label y_i, the confidence c_i is calculated as follows:

$$c_i = \frac{1}{E}\sum_{e=1}^{E} p(y_i|X_i, \theta^e) \quad (1)$$

Here, E is the number of training epochs, θ_e is the model at epoch e and p is the probability of the label given X_i and θ_e. The *variability* of each example is defined as the standard deviation of the predicted probability for its label over the training epochs. More formally, the variability, v_i of an example X_i is calculated as:

$$v_i = \sqrt{\frac{\sum_{e=1}^{E}(p(y_i|X_i, \theta^e) - c_i)^2}{E}} \quad (2)$$

Finally, the fraction of the training epochs where the classifier predicts the label of an example correctly is defined as its *correctness*.

PROMPT - 1	<human>: Consider the following two sentences: Sentence1: <sentence1> Sentence2: <sentence2> What is the semantic relation between Sentence1 and Sentence2? Choose from the following options: 1. Entailment, 2. Reasoning, 3. Contrasting, 4. Neutral. <bot>:
PROMPT - 2	<human>: Consider the following class definitions of four semantic relations between a pair of sentences. Entailment: <definition of entailment> Contrasting: <definition of contrasting> Reasoning: <definition of reasoning> Neutral: <definition of neutral> Now consider the following two sentences: Sentence1: <sentence1> Sentence2: <sentence2> Based on only the information available in these two sentences and the class definitions, answer the following: What is the semantic relation between Sentence1 and Sentence2? Choose from the following options: 1. Entailment, 2. Reasoning, 3. Contrasting, 4. Neutral. <bot>:
PROMPT - 3	<human>: Consider the following two sentences: Sentence1: <sentence1> Sentence2: <sentence2> Based on only the information available in these two sentences, which of the following options is true? a. Sentence1 generalizes, specifies or has an equivalent meaning with Sentence2. b. Sentence1 presents the reason, cause, or condition for the result or conclusion made Sentence2. c. Sentence2 mentions a comparison, criticism, juxtaposition, or a limitation of something said in Sentence1. d. Sentence1 and Sentence2 are independent. <bot>:

Table 18: Prompt templates used for our experiments with LLMs. Here, <X> indicates a placeholder X which is replaced in the actual prompts.

E.2 Data Plot

For creating the data plot, we fine-tune a RoBERTa classifier on the combined MSciNLI training set. While training, we record the probability distributions predicted by the classifier for the training examples over the four labels in each epoch. We then calculate the confidence, variability, and correctness of each example using the recorded probability distributions and plot them in the data map based on these calculated values. The data plot can be seen in Figure 1.

We can see that the model shows a high *correctness* for the examples in the high *confidence* region. Therefore, the examples in this region are *easy-to-learn* for the model. On the other hand, the plot shows that the *correctness* of the model's predictions is very low in the low *confidence* region of the map. Thus, the examples in this region are *hard-to-learn* for the model. Since, by definition, the probability predicted by the model shows a high fluctuation for the examples in the high *variability* region, they can be denoted as *ambiguous* examples.

Based on these observations from the data map, we select the various subsets from the full training set as follows.

E.3 Data Subset Selection

We rank the full training set based on *confidence* in a *descending* order and then select the top 33% examples as the 33% *easy-to-learn* subset. Similarly, we rank the full training set based on *confidence* in an *ascending* order and then select the top 33% examples as the 33% *hard-to-learn* subset. The top 33% examples from a ranking based on the *variability* in a *descending* order is chosen as the top 33% *ambiguous* subset. For the '$100\%-$ top 25% *hard*' and '$100\%-$ top 5% *hard*' subsets, we remove top 5% and 25% examples from the ranking based on *confidence* in an *ascending* order, respectively from the full training set.

F Additional Cross-dataset Experiments

F.1 Domain-wise Performance by Cross-dataset Models

We can see the domain-wise performance of the cross-dataset models described in Section 4.3.3 in

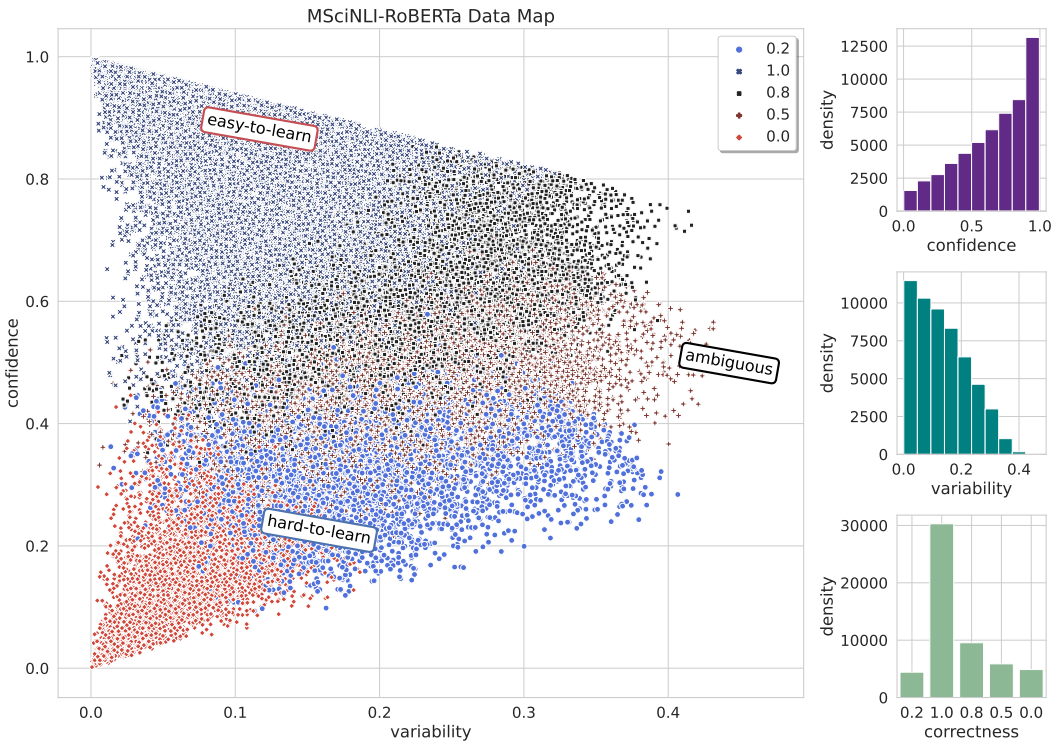

Figure 1: Data cartography of MSciNLI. The colors and shapes indicate the correctness of each example.

Train \ Test	Hardware	Networks	SWE	Security	NeurIPS	ACL
SciNLI	75.60 ± 0.8	72.71 ± 0.5	74.36 ± 0.3	75.00 ± 0.3	78.36 ± 1.0	78.08 ± 0.4
MSciNLI	77.79 ± 0.2	75.45 ± 1.5	$\mathbf{77.10 \pm 0.7}$	77.71 ± 0.2	78.04 ± 0.8	76.74 ± 0.5
MSciNLI+	$\mathbf{77.99 \pm 0.4}$	$\mathbf{77.48 \pm 0.4}$	76.78 ± 1.1	$\mathbf{78.08 \pm 1.4}$	$\mathbf{80.02 \pm 1.4}$	$\mathbf{79.48 \pm 0.4}$

Table 19: Macro F1 scores (%) of the cross-dataset models based on RoBERTa on different domains. Here, SWE: Software & its Engineering and Security: Security & Privacy. Best scores are in bold.

Tr \ Te	SciNLI	MSciNLI	SciTail	MNLI
SciNLI	86.03	81.43	51.62	53.63
MSciNLI	83.18	82.56	55.66	58.72
SciTail	48.64	48.86	91.19	73.42
MNLI	45.40	47.57	78.18	91.31

Table 20: Cross dataset performances (Macro F1 (%)) of RoBERTa on different datasets in a 2-class setting.

Table 19. The results show that MSciNLI+ shows a better performance on domain-level as well.

F.2 Out-of-dataset Performance on Regular NLI datasets

To understand the effect of data diversity in the performance of scientific NLI models on regular NLI datasets, we perform a set of experiments with RoBERTa where we train models on SciNLI, MSciNLI, SciTail (Khot et al., 2018), and MNLI (Williams et al., 2018a) and evaluate them on each of the test sets of these datasets. Note that since the test set of MNLI is not publicly available, we use the development set as the test set and a randomly sampled set of size $10,000$ as the development set. Given that the NLI classes differ in these datasets, we convert SciNLI, MSciNLI and MNLI into 2-class datasets. Specifically, we update the labels of all non-entailment classes, i.e., contradiction and neutral for MNLI and contrasting, reasoning, neutral for SciNLI and MSciNLI to a class named Not-Entailment. We do not change any labels in SciTail because it is already in a 2-class setting using Entailment and Not-Entailment as the classes. The Macro F1 from these experiments are in Table 20.

We can see that the model trained on MSciNLI shows a substantially higher performance on both MNLI, and SciTail compared to the model

Dataset	Classes
SciHTC	'General and reference', 'Hardware', 'Computer systems organization', 'Networks', 'Software and its engineering', 'Theory of computation', 'Mathematics of computing', 'Information systems', 'Security and privacy', 'Human-centered computing', 'Computing methodologies', 'Applied computing', 'Social and professional topics'
Paper Field	'Geography', 'Politics', 'Economics', 'Business', 'Sociology', 'Medicine', 'Psychology'
ACL-Arc	'Background', 'Extends', 'Uses', 'Motivation', 'Compare/Contrast', 'Future work'

Table 21: Downstream task datasets and their classes.

Dataset	#Train	#Test	#Dev
SciHTC	148,928	18,616	18,616
Paper Field	84,000	22,399	5,599
ACL-Arc	1,688	139	114

Table 22: Number of examples in downstream tasks.

trained with SciNLI. Therefore, training the models on diverse examples improves their reasoning capabilities which results in a better performance even for traditional NLI datasets. In our future work, we will investigate how the models trained on scientific NLI datasets behave when they are tested on *easy*, *ambiguous* and *hard-to-learn* examples of the traditional NLI datasets.

G Details on Intermediate Task Transfer

G.1 Downstream Tasks - Dataset Details

The categories/class labels and the number of examples in each dataset for the downstream tasks in our intermediate task transfer experiments can be seen in Tables 21 and 22, respectively.

The details of each the downstream tasks that we experiment with are as follows.

SciHTC (Sadat and Caragea, 2022a) A hierarchical multi-label scientific topic classification dataset containing 186K papers. While each paper in SciHTC is assigned multiple labels from different levels of the hierarchy tree), we only consider the level 1 flat categories which are 13 in total (see Table 21) and train the model in a multi-class (single label for each paper) setting.

Paper Field (Beltagy et al., 2019) A paper classification dataset containing 112K papers where each paper is classified to different scientific fields. The total number of paper classes in this dataset is 7 (see Table 21).

ACL-ARC (Jurgens et al., 2018) A citation intent classification dataset where the intent behind a citation made in a sentence in a scientific paper needs to be predicted. The 6 classes in this dataset can be seen in Table 21.

G.2 Experimental Details of Intermediate Task Transfer Learning

In the intermediate task transfer setting, the RoBERTa model is trained on the NLI datasets for a single epoch (unlike the baselines). For the unsupervised intermediate training with MLM, 15% tokens are randomly masked and the model is also trained for a single epoch. During the fine-tuning step, only the RoBERTa layer is initialized from the model from the intermediate training step. The parameters for the output linear layer with softmax activation is randomly initialized. The model is then fine-tuned for the downstream tasks for multiple epochs. Specifically, the models for SciHTC, and Paper-field are trained for 10 epochs. The models for ACL-Arc are fine-tuned for a maximum of 20 epochs due to its small size. Similar to our baselines, we employ early stopping with patience 2 and Macro F1 score of the development set as the stopping criteria.