Who’s a Better Scholar: Encoder *or* Decoder?

Anonymous ACL submission

Abstract

Language modeling has seen seen a tremendous development over past few years, with a considerable rise in their deployment for solving domain-specific Natural Language Processing (NLP) tasks. In recent times, the fundamental building blocks of language models are essentially composed of either an encoder-based architecture or a decoder-based architecture or a combination of both. In the scholarly domain, the majority of use cases have explored only the utilization of encoder-only models for a variety of tasks using the pre-trained model fine-tuning approach. But the same has not yet been replicated for decoder based models in spite of the recent popularity of LLMs. To address this issue, we fine-tune both encoder-based language models and decoder-based language models on an array of traditional scholarly NLP tasks. This allows us to compare the effect of learned representations in contrast to generation-based techniques on standard scholarly benchmark datasets. We conduct extensive experiments on 10 highly popular human-annotated datasets over 6 different tasks and also study the effect of domain-specific pre-training on these tasks. We achieve SOTA over two tasks using decoder-based language models.

1 Introduction

Scientific literature understanding is an important facet of Natural Language Understanding and is highly useful in the comprehension of large collections of scientific text. There has been a growing interest to explore the nuances of standard NLP tasks in the scholarly domain and, in most cases, the best results have come from fine-tuning a pre-trained language model (Beltagy et al., 2019; Lahiri et al., 2024; Sadat and Caragea, 2022).

Recently, Large Language Models (LLMs) are increasingly adopted for most NLP tasks. They contain tens to hundreds of billions of parameters, and are much larger than their predecessor Pre-trained Language Models (PLMs).

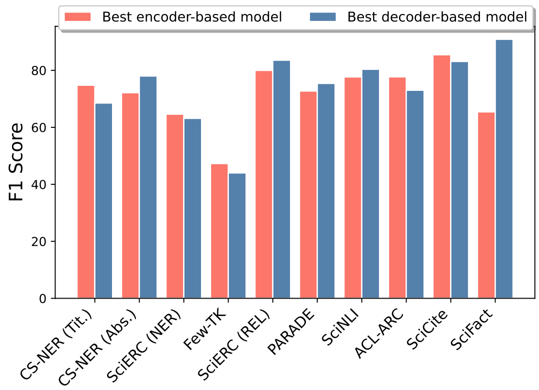

Figure 1: Comparison of the scores achieved by the *best* performing encoder-based and decoder-based LMs.

LLMs and PLMs trace their architectural roots to the original Transformer model (Vaswani et al., 2017). While LLMs like LLaMA (Touvron et al., 2023a) generally use only the decoder module of the Transformer, PLMs like BERT (Devlin et al., 2019) typically leverage only the encoder while PLMs like T5 (Raffel et al., 2020) are comprised of both the encoder and the decoder. Encoder-based models, although task-agnostic, generally need to go through fine-tuning over a limited amount of task-specific data to achieve proficiency in that particular task. LLMs possess greater emergent and reasoning capabilities (Wei et al., 2022a,b; Yao et al., 2023), yet, they are reported to be even more accomplished when fine-tuned over task-specific data (Minaee et al., 2024; Wadden et al., 2024).

Given that LLMs (that are decoder-only models) incur exorbitant computational and environmental costs, we ask if they indeed outperform the smaller PLMs which are either encoder-only or use both encoders and decoders. Inspired by the existing NLP task sets, we build a novel set of common scholarly NLP tasks with a focus on those where encoders have been applied successfully and LLMs have been hardly experimented with. We fine-tune

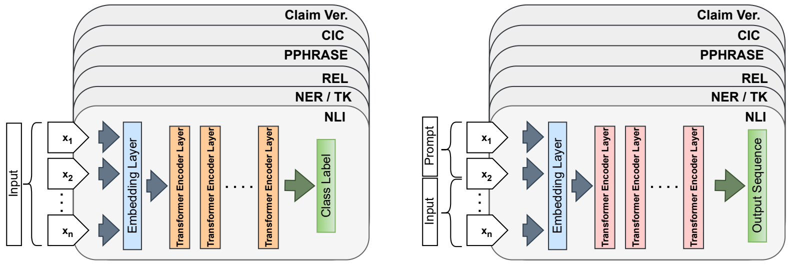

Figure 2: Fine-tuning for a Transformer encoder-based LM (left) and for a Transformer decoder-based LM (right).

all the chosen models on the corresponding datasets and evaluate on the test sets.

Figure 1 gives a sneak preview into your observations. Although very recently related studies have been conducted for word meaning understanding (Qorib et al., 2024) and multi-lingual natural language understanding (Nielsen et al., 2024), ours is the first in scholarly NLP. To this end, we have identified 6 scholarly tasks and corresponding publicly available datasets on which we fine-tune several PLMs (encoder-only and encoder-decoder models) and LLMs (decoder-only), some of which are pre-trained on scholarly datasets and some only on open-domain corpora. Note that our aim is *not* to achieve state-of-the-art (SOTA) results – though we do achieve SOTA for two tasks – but rather contrast the performance among the model types.

Contributions

⋆ We compare decoder-only, encoder-only, and encoder-decoder based models on 10 benchmark scholarly datasets over 6 different tasks. We use 2 encoder-based LMs and 6 decoder-based LMs.

⋆ Our experiments indicate that encoder-only models outperform decoder-only and hybrid models for most of the tasks. Moreover, decoder models hallucinate novel output categories even when prompted with the correct label set for classification.

⋆ We study the effect of domain-specific data in the pre-training corpus. Pre-training with in-domain data generally improves downstream performance for all encoders, encoder-decoders and decoders.

⋆ Parameter-efficient fine-tuning of LLMs takes much longer than full fine-tuning of encoder-based and hybrid models.

2 Related Work

Since the first Transformer model was proposed in 2017 (Vaswani et al., 2017), several PLMs and LLMs have been developed, many of which are specifically pre-trained or fine-tuned on domain-specific data. Models built by fine-tuning and instruction-tuning LLaMA (Touvron et al., 2023a) and LLaMA-2 (Touvron et al., 2023b) include Code LLaMA (Rozière et al., 2024), Vigogne (Huang, 2023), Tülu (Wang et al., 2023), Tülu-2 (Ivison et al., 2023) and Stable Beluga2 (Mahan et al.). Galactica (Taylor et al., 2022), DARWIN (Xie et al., 2023), SciTülu (Wadden et al., 2024) and SciLitLLM (Li et al., 2024) are some recently developed LLMs that have scientific knowledge injected into them, and they perform better than general-domain LLMs on scientific tasks.

Evaluation of PLMs and LLMs – in open-domain as well as domain-specific areas – is a critical and challenging research area. Popular NLP task benchmarks include GLUE (Wang et al., 2019b), SuperGLUE (Wang et al., 2019a) and MMLU (Hendrycks et al., 2021) – all spanning multiple domains. (AI4Science and Quantum, 2023) explores the performance of GPT-4 on a range of scientific domains including drug discovery, biology, computational chemistry, materials design, and partial differential equations. SciBench (Wang et al., 2024) and SciEval (Sun et al., 2024) are benchmarks designed for evaluating the scientific reasoning capabilities of LLMs. These studies mainly examine only the zero-shot, few-shot and chain-of-thought inferencing capabilities of LLMs to identify the best performing models. Perhaps, the closest work to ours is the SciRIFF (Wadden et al., 2024), which creates an instruction-tuning dataset for scientific literature understanding and fine-tunes the Tülu V2 checkpoint. In contrast, our work is more aligned towards the evaluation of decoder-based, encoder-based and hybrid LMs.

3 Tasks

Inspired by NLP task benchmarks like GLUE (Wang et al., 2019b), MMLU (Hendrycks et al., 2021) and datasets for multi-task learning for LLMs (Wadden et al., 2024), we have built a collection of 6 scholarly NLP tasks, each of which is briefly described below. The details of the datasets shown in Figure 3 are in the Appendix A.

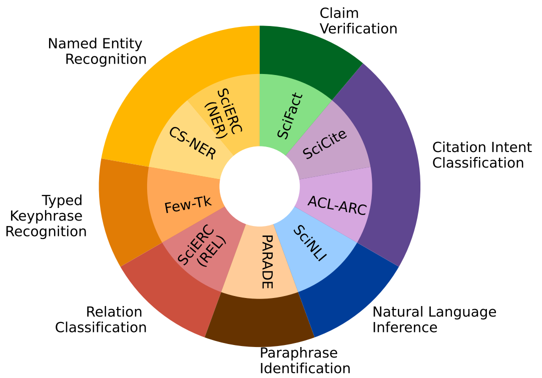

Figure 3: Tasks and Datasets

Dataset Selection Rationale: Our emphasis is on tasks where the focus is on understanding and classification problems rather than tasks that are primarily generative in nature. We started with tasks like keyphrase recognition, meta-review generation, scientific publication rating and extractive question answering, but later on, we excluded these tasks due to their generative nature. Among the tasks that we select, we include all the major datasets in that task domain and are publicly available.

3.1 NER/TK: Named Entity Recognition/ Typed Keyphrase Recognition

Named Entity Recognition (NER) is the Information Extraction (IE) task of identifying references to rigid designators (Nadeau and Sekine, 2007). Recently (Lahiri et al., 2024) presented a broader definition for this task in the scientific domain and termed it as Typed Keyphrase Recognition.
Definition: The input is a sequence of tokens $x = (x_1, x_2, ..., x_n)$, from which we derive a set $S = \{s_1, ...s_p\}$, which represents a set of semantically meaningful within-sentence contagious sequence spans each of which is assigned a label from the set $Y = \{y_1, y_2, ..., y_m\}$. The elements in set S may contain words, phrases or other syntactic units from the given text sequence x. Therefore, the final output can be construed as $Z = \{(s_i, y_j) : i \in 1, ..., p; j \in 1, ..., m; s_i \in S; y_j \in Y\}$.

3.2 REL: Relation Classification

Relation Classification is also an Information Extraction task, wherein the objective is to predict the relationship type between a given ordered pair of spans within a sentence.
Definition: The input is a sequence of tokens $x = (x_1, x_2, ..., x_n)$ and two entities (spans), $s_A = (x_i, ..., x_j)$ and $s_B = (x_u, ..., x_v)$, the expected output is a triple (s_A, s_B, r), where $r \in R$ such that R is a pre-defined set of relation labels.

3.3 PPHRASE: Paraphrase Recognition

Sentences or phrases conveying identical meaning but with the use of different wording are called paraphrases. The model's ability to demonstrate specialized domain knowledge is tested in the scholarly paraphrase identification task(He et al., 2020).
Definition: A pair of sentences (s_1, s_2) are to be classified as paraphrases or non-paraphrases.

3.4 NLI: Natural Language Inference

Natural Language Inference (NLI), also known as Textual Entailment (Bowman et al., 2015; Sadat and Caragea, 2022), is the task of identifying whether there is an entailment or a contradiction between a pair of sentences or whether they are independent of each other.
Definition: Given a pair of sentences (s_1, s_2), the task is to assign a label $y \in Y$ which indicates the semantic relatedness of the latter to the former.

3.5 CIC: Citation Intent Classification

Citations form an important part of scientific documents. The kind of purpose the citation serves in the scholarly document is known as its citation intent (Roman et al., 2021).
Definition: The input is a citation sentence x and the aim is to assign a class label $y \in Y$, where Y is the set of citation intents.

3.6 CLAIM: Claim Verification

This task intends to assess the truthfulness of a claim (Vlachos and Riedel, 2014), which is important in the scientific domain due to the possibility of a far-reaching impact of a decision taken based on some scientific misinformation. We follow the simplified setting of (Vladika and Matthes, 2024) where the model is provided with golden abstracts:
Definition: Given a claim c and an evidence abstract d (each of which is a sequences of tokens), the task is to find whether c supports or refutes the abstract d.

NER/TK Model	CS-NER (Titles)				CS-NER (Abstracts)			
	Precision	Recall	F1	H	Precision	Recall	F1	H
BERT	72.83	**76.81**	74.77	0	69.38	71.32	70.33	0
SciBERT	**72.98**	76.66	**74.78**	0	72.97	**71.35**	72.14	0
T5	30.53	8.74	12.25	0	59.22	26.60	36.62	0
SciFive	25.30	8.14	11.25	0	59.59	26.55	36.62	0
LLaMA-7B	66.00	70.38	68.12	1	83.29	68.18	74.98	0
LLaMA-13B	65.72	70.50	68.03	3	82.64	69.03	75.22	0
LLaMA-70B	66.41	70.61	68.45	3	**90.00**	62.92	74.06	0
SciLitLLM-7B	67.33	69.35	68.32	0	86.42	70.79	77.83	0
Tülu-2-dpo-7B	66.47	65.74	66.10	1	79.85	70.70	75.00	0
Tülu-2-dpo-70B	67.25	69.83	68.52	3	88.35	69.82	**78.00**	0

Table 1: Results for fine-tuning encoder-based LMs and instruction-tuning decoder-based LMs on **CS-NER (Titles)** and **CS-NER (Abstracts)** for Named Entity Recognition. H stands for Hallucinated Tags, i.e., the tags which LLMs have generated, but are not part of the dataset's annotation schema.

4 Hypothesis and Research Questions

Hypothesis: We hypothesize that, in traditional scholarly NLP tasks, encoder models perform better than the (much larger) decoder-based models. More specifically, we ask:

- RQ1: (a) Do decoder-based or encoder-decoder-based models outperform their encoder-based counterparts?

  (b) Are decoder-based LLMs lacking in sequence labeling and classification tasks?
- RQ2: Are domain-specific models better than their counterparts?
- RQ3: Which models are more computationally efficient?

5 Experimental Setup

We test our hypothesis with three categories of models: only-encoder-based models, encoder-decoder-based models and only-decoder-based models.

Encoder-based Language Models: We use the BERT-base model (Devlin et al., 2019) and the SciBERT-base model (Beltagy et al., 2019) model checkpoints as the encoder-based LMs in our experiments. More details about the models and the experimental setup are present in the Appendix B.

Encoder-Decoder-based Language Models: We use the T5-base (Raffel et al., 2020) and the SciFive-base-PMC (Phan et al., 2021) as the encoder-decoder-based models in our experiments. The details about the hyperparameters and the models are in the Appendix C.

Decoder-based models: We use the 7B, 13B and the 70B model variants of LLaMA-2 (Touvron et al., 2023b), SciLitLLM-7B[1] (Li et al., 2024) and 7B and 70B variants of Tülu-2 (Ivison et al., 2023) as the decoder-based LMs in our experiments. We instruction-tune the decoder-based LMs using QLoRA (Dettmers et al., 2023), which is an efficient approach for fine-tuning LLMs using relatively less GPU memory. QLoRA uses 4-bit NormalFloat, Double Quantization and Paged Optimizers on the Low-rank Adapter (LoRA) fine-tuning approach (Hu et al., 2022). Details about the models and the hyperparameters used can be found in Appendix D.

Prompt Creation: We follow (Taori et al., 2023) We use simple intuitive prompts that are similar to the ones used for the Alpaca project[2]. We do not focus on prompt optimization (Schulhoff et al., 2024) as we fine-tune and evaluate the LLMs with the same prompts and our target is to make a comparative performance study of the models rather than achieving SOTA results.

========INSTRUCTION FORMAT========

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

###Instruction:
[Instruction Prompt]

###Input:
[Input Text]

###Response:
[Output Text]

The instruction and input-output pair format for every task can be found in Appendix G.

[1] https://huggingface.co/Uni-SMART/SciLitLLM

[2] https://huggingface.co/datasets/tatsu-lab/alpaca

REL Model	Cmp.	Cnj.	Evl.-for	Ft.-of	Hyp.-of	Pt.-of	Used-for	F1	H
BERT	81.58	90.98	82.47	60.00	91.30	53.57	92.19	78.87	0
SciBERT	84.21	93.17	84.82	58.91	89.21	59.65	92.95	80.42	0
T5	80.00	92.43	82.84	59.46	88.73	50.47	92.77	78.10	0
SciFive	70.13	90.62	78.36	52.63	82.01	45.45	90.29	72.79	0
LLaMA-7B	87.32	94.4	87.01	**71.54**	**94.03**	**68.38**	93.67	74.54	2
LLaMA-13B	88.31	94.02	**89.73**	64.08	90	64.35	**94.34**	**83.55**	0
LLaMA-70B	**88.57**	93.02	86.34	66.67	84.93	37.97	93.66	78.74	0
SciLitLLM-7B	87.32	**94.82**	89.13	64.91	92.09	61.95	93.95	73.02	1
Tülu-2-dpo-7B	**88.57**	92.86	84.21	60.00	82.64	60	92.84	80.16	0
Tülu-2-dpo-70B	87.18	93.06	83.17	62.50	90.91	66.07	93.83	72.09	3

Table 2: Results for fine-tuning encoder-based LMs and instruction-tuning decoder-based LMs on **SCIERC** for Relation Classification. H stands for Hallucinated Tags.

NER/TK Model	P	R	F1	H
BERT	59.71	65.95	62.67	0
SciBERT	**62.24**	**67.2**	**64.62**	0
T5	51.74	24.60	32.67	0
SciFive	54.20	25.99	34.44	0
LLaMA-7B	58.57	61.83	60.16	4
LLaMA-13B	57.94	62.26	60.02	0
LLaMA-70B	61.42	64.95	63.14	4
SciLitLLM-7B	58.39	60.67	59.51	1
Tülu-2-dpo-7B	59.95	61.9	60.91	2
Tülu-2-dpo-70B	60.81	60.55	60.68	3

Table 3: Results for fine-tuning encoder-based LMs and instruction-tuning decoder-based LMs on **SCIERC** for Named Entity Recognition. H stands for Hallucinated Tags.

NER/TK Model	P	R	F1	H
BERT	40.59	45.05	42.66	0
SciBERT	**46.87**	**47.82**	**47.29**	0
T5	39.68	13.14	18.71	0
SciFive	35.63	11.92	16.93	0
LLaMA-7B	39.54	40.17	39.86	5
LLaMA-13B	40.51	46.12	43.13	8
LLaMA-70B	40.4	44.38	42.29	5
SciLitLLM-7B	41.47	44.96	43.15	16
Tülu-2-dpo-7B	38.36	41.48	39.86	15
Tülu-2-dpo-70B	42.55	45.54	43.99	5

Table 4: Results for fine-tuning encoder-based LMs and instruction-tuning decoder-based LMs on **Few-TK** for Typed Keyphrase Recognition. H stands for Hallucinated Tags.

6 Results

6.1 Named Entity Recognition

Table 1 presents the results for the CS-NER (Abstracts) and CS-NER (Abstracts) datasets (D'Souza and Auer, 2022). Table 3 shows the results obtained for SCIERC (Luan et al., 2018), another NER dataset. For the NER task, the generative decoder-based LMs, despite having the class names specified in the prompt, *hallucinate* new labels such as Objective, Scenario, Author, Profession, User, and Drug among others. We see that for CS-NER (Abstracts), none of the models hallucinate, which is perhaps due to the fact that it consists of only two classes, whereas SCIERC and CS-NER (Titles) contains six and seven classes respectively.

6.2 Typed Keyphrase Recognition

Table 4 shows the results on the Few-TK dataset (Lahiri et al., 2024). Similar to the results for NER, here too we see that SciBERT outperforms all other models, although the results are generally low for this dataset. This is due to large number of classes, which is 38, in this dataset, that is much higher than that of other datasets in this domain. This shows that simple vanilla fine-tuning or instruction-tuning may not be enough for more complex multi-label tasks such as these as they require significantly higher reasoning capabilities. We also see that due to the larger number of classes into which the keyphrases are to be divided, the number of hallucinations for this dataset are also much larger.

6.3 Relation Classification

Table 2 shows the results for relation classification on the SCIERC dataset. LLaMA-13B is found to be the best performing model for this task, which to the best of our knowledge is also the SOTA for relation classification on this dataset. Some of the hallucinated labels from generative decoder-based LMs are Induced-from, Sum-of and Weighted-sum, in the very rare cases where they hallucinate.

6.4 Paraphrase Recognition

Table 5 shows the results for the task of paraphrase recognition. Although the results achieved by each

PPHRASE Model	Paraphrase	Non-paraphrase	Accuracy	Precision	Recall	F1
BERT	72.21	73.28	72.78	72.88	72.83	72.74
SciBERT	71.77	73.63	72.59	72.54	72.55	72.54
T5	72.65	71.96	72.32	72.54	72.48	72.30
SciFive	69.74	74.20	72.20	72.41	71.98	71.97
LLaMA-7B	**73.69**	72.18	72.96	73.39	73.20	72.93
LLaMA-13B	73.13	71.24	72.22	72.72	72.49	72.19
LLaMA-70B	73.30	77.30	75.46	75.58	75.25	75.30
SciLitLLM-7B	73.15	**77.65**	**75.61**	**75.82**	**75.36**	**75.40**
Tülu-2-dpo-7B	65.93	77.27	72.73	75.20	72.02	71.60
Tülu-2-dpo-70B	63.83	76.86	71.78	74.86	70.98	70.35

Table 5: Results for fine-tuning encoder-based LMs and instruction-tuning decoder-based LMs on **PARADE** for paraphrase recognition. We report the overall precision, recall, macro F1, accuracy and the class-wise macro F1.

NLI Model	Contrasting	Reasoning	Entailment	Neutral	F1	Accuracy
BERT	77.17	71.25	74.37	74.01	74.20	74.27
SciBERT	79.69	74.35	74.35	76.46	77.68	77.67
T5	79.68	72.06	75.54	77.65	76.10	76.16
SciFive	80.86	73.88	77.34	**78.52**	77.65	77.72
LLaMA-7B	78.22	69.53	73.53	61.05	70.58	71.10
LLaMA-13B	82.92	74.93	77.60	71.71	76.79	76.98
LLaMA-70B	86.17	74.45	77.77	64.51	75.73	76.50
SciLitLLM-7B	82.54	76.52	77.06	69.77	76.47	76.80
Tülu-2-dpo-7B	79.82	71.03	74.87	63.86	72.39	72.85
Tülu-2-dpo-70B	**87.24**	**78.22**	**79.20**	76.23	**80.22**	**80.37**

Table 6: Results for fine-tuning encoder-based LMs and instruction-tuning decoder-based LMs on **SciNLI** for Natural Language Inference. We report the overall macro F1, accuracy and the class-wise macro F1.

of the models are very close to each other, decoder-based LMs hold a slight edge in performance over encoder-based LMs, with the SciLitLLM-7B being the best performing model by outperforming even the 70B models.

6.5 Natural Language Inference

Table 6 shows the results for scientific Natural Language Inference. The Tülu-2-dpo-70B model shows superior performance among the tested models and also achieves the SOTA performance on this dataset (Sadat and Caragea, 2024).

6.6 Citation Intent Classification

Table 7 and Table 8 shows the result for Citation Intent Classification on the ACL-ARC (Jurgens et al., 2018) and SciCite (Cohan et al., 2019) datasets, respectively. We see that for both the datasets SciBERT shows better performance. Only for F1 scores of two classes of the ACL-ARC dataset and the overall accuracy score, other language models are able to perform better than SciBERT. LLaMA-70B and Tülu-2-dpo-70B – both 70B LLMs clock almost about the same overall F1 score, whereas the two 7B models show some hallucinations like Repeats and Inspired.

6.7 Claim Verification

Table 9 shows the result for Claim Verification on the SciFact dataset (Wadden et al., 2020). This is the only task where we find that a large language model i.e. the Tülu-2-dpo-70B model is the best performing model on all metrics and is also separated from the encoder-based LMs by a huge margin.

7 Performance Analysis

RQ1: (a) Do decoder-based or encoder-decoder-based models outperform their encoder-based counterparts?

We find that encoder-based LMs offer stiff competition to their decoder-based counterparts even though the encoder-based LMs are quite smaller in size and trained on much less data. Decoder-based LMs perform well in those tasks where the number of labels or classification heads are less

CIC Model	Bckg.	Comp.	Extends	Future	Motiv.	Uses	Accuracy	F1	H
BERT	84.12	59.15	44.81	21.67	00.00	64.91	45.78	70.74	0
SciBERT	**87.67**	**73.76**	**73.13**	**76.26**	41.79	78.42	74.96	**77.70**	0
T5	84.80	73.62	44.44	75.56	54.55	72.29	77.94	67.54	0
SciFive	89.33	77.93	64.45	88.38	53.03	76.31	82.73	74.90	0
LLaMA-7B	84.62	60.00	61.54	50.00	**71.43**	84.44	77.70	58.86	2
LLaMA-13B	86.09	68.18	50.00	66.67	40.00	80.77	78.42	65.29	0
LLaMA-70B	84.97	63.41	72.73	80.00	26.67	79.17	76.98	67.82	0
SciLitLLM-7B	84.00	60.47	61.54	72.73	36.36	76.00	75.54	65.18	0
Tülu-2-dpo-7B	84.93	60.00	46.15	72.73	44.44	77.55	74.82	55.12	1
Tülu-2-dpo-70B	84.97	61.90	80.00	72.73	53.33	**85.11**	**79.14**	73.01	0

Table 7: Results for fine-tuning encoder-based LMs and instruction-tuning decoder-based LMs on **ACL-ARC** for Citation Intent Classification. We report the overall macro F1, accuracy and the class-wise macro F1. H stands for Hallucinated Tags.

CIC Model	Background	Method	Result	Accuracy	F1
BERT	88.28	85.28	80.6	86.17	84.72
SciBERT	**88.51**	**86.33**	**81.53**	**86.75**	**85.46**
T5	88.72	84.63	81.53	86.39	84.96
SciFive	88.46	85.62	82.56	86.69	85.54
LLaMA-7B	85.85	81.44	77.96	83.37	81.75
LLaMA-13B	85.31	80.28	77.12	82.56	80.90
LLaMA-70B	86.83	82.58	79.92	84.55	83.11
SciLitLLM-7B	86.10	81.02	79.06	83.48	82.06
Tülu-2-dpo-7B	86.54	82.41	76.73	83.80	81.89
Tülu-2-dpo-70B	86.19	83.09	80.00	84.23	83.10

Table 8: Results for fine-tuning encoder-based LMs and instruction-tuning decoder-based LMs on **SciCite** for Citation Intent Classification. We report the overall macro F1, accuracy and the class-wise macro F1.

than or equal to 3. Among the tasks considered, decoder-based LMs have been found to work well in tasks like Paraphrase Recognition, Natural Language Inference and Claim Verification.

On the bright side, our experiments on decoder-based LMs have led to achieving SOTA performance on two tasks – Relation Classification and Natural Language Inference.

> RQ1: (b) Are decoder-based LLMs lacking in sequence labeling and classification tasks?

We see that the 110M-parameter SciBERT is a better performer than most decoder-based models on most tasks. We can attribute two factors to this performance: the first is that difference in the way that encoder-based and decoder-based models are pre-trained and the second, as mentioned in our paper, is due to the hallucinations in LLMs. Encoder-based models undergo pre-training majorly using the Masked Language Modelling objective, while decoder-based models are pre-trained on the Next Token Prediction objective. Therefore, we postulate that the embedding generated by encoder-based models using bidirectional attention contains much more precise information than the unidirectional attention used by decoder-based models. Decoder-based models are only trained to see the next token, which may not be so useful in tasks like sequence labeling like NER or sentence classification tasks like NLI and others.

(Wadden et al., 2024) reports the F1 score in the SCIERC using GPT-4 to be 42.2 and using their own SCITÜLU 70B model to be 35.9. Therefore, we see that fine-tuning decoder-based LMs gives far better results than the simply prompting.

We see that many of the decoder-based LMs hallucinate when there are too many labels for classification. Hallucinations are a major reason for the overall decrease in performance of decoder-based LMs in many tasks. We postulate that the pre-training of large generative models plays a major part in such hallucinations, where in spite of the classes being mentioned in the training prompt, the model in a few exceptional cases generates data which is meaningful but does not pertain to the constrained framework of the given task.

> RQ2: Are domain-specific models better than their counterparts?

We see across all tasks that language models that have been pre-trained on scholarly data have a slight edge over those trained on general domain data. We observe this trend both in the case of encoder-based models (SciBERT) and decoder-based models (SciLitLLM and Tülu-2). But, we notice an interesting scenario in the case of Tülu-2: SCIERC (one of our NER and relation classification datasets) is included within its pre-training data and even after explicitly fine-tuning on the same data, we do not obtain an improvement in the results. Yet, although SciFact occurs in Tülu-2 pre-

CLAIM **Model**	**Support**	**Contradict**	**Accuracy**	**Precision**	**Recall**	**F1**
BERT	77.14	00.52	62.82	34.15	49.21	38.83
SciBERT	80.22	53.15	69.82	66.89	65.15	65.41
T5	75.47	52.95	67.75	64.69	63.95	64.21
SciFive	78.26	53.68	70.41	67.73	65.44	65.97
LLaMA-7B	81.87	51.89	73.67	74.64	66.20	66.88
LLaMA-13B	85.59	71.11	80.77	79.90	77.46	78.35
LLaMA-70B	90.20	79.26	86.69	87.86	83.16	84.73
SciLitLLM-7B	85.27	69.68	80.18	79.47	76.46	77.48
Tülu-2-dpo-7B	83.41	67.83	78.11	76.55	75.02	75.62
Tülu-2-dpo-70B	**93.08**	**88.72**	**91.42**	**90.25**	**91.86**	**90.9**

Table 9: Results for fine-tuning encoder-based LMs and instruction-tuning decoder-based LMs on **SciFact** for Claim Verification. We report the overall precision, recall, macro F1, accuracy and the class-wise macro F1.

training corpus, hallucinations do not occur during claim verification on SciFact. Therefore, we again conclude that hallucinations play a large role in the performance of decoder-based models.

RQ3: Which models are more computationally efficient?

The time taken by decoder models is shown in 5. Encoder-based LMs take much lower time for both training and inferencing than decoder-based LMs, which require anywhere about 4 to 26 A100 GPU hours per dataset only for the training part. Apart from this, the inferencing stage is also a time-consuming process with datasets like CS-NER which have large amounts of test data requiring more than 12 hours on an A100 GPU. In comparison, encoder-based LMs require at most 5-6 hours for the completion of both the training and inferencing stages. SciLitLLM (Li et al., 2024) takes an inordinately large amount of time for the inferencing phase in spite of its model size.

7.1 Experimental Setup Analysis

We do not opt for multi-task fine-tuning of LLMs as we have chosen a diverse range of tasks and therefore, there is a high possibility of negative transfer even though multi-task fine-tuning is a viable option sometimes while dealing with related tasks (Karimi Mahabadi et al., 2021).

We choose BERT (Devlin et al., 2019) over other variants of Transformer encoder based model variants because other architecturally similar models do not show any drastic improvement in performance over BERT and also because of the popularity of BERT on standard NLP tasks. We do not use the SciTülu (Wadden et al., 2024) checkpoints for our experiments as most of the datasets overlap with their training data and this would not have been suitable for our experiments.

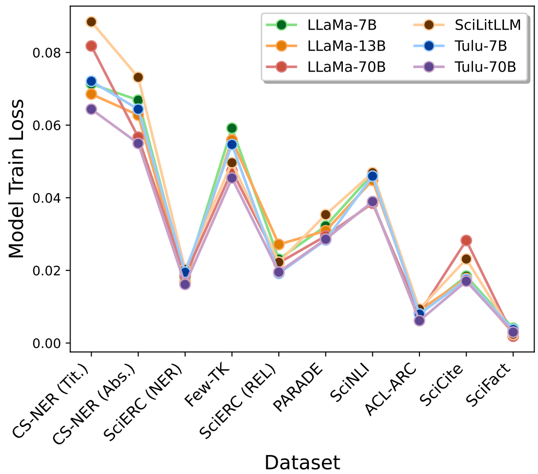

Figure 4: Train loss in decoder-based language models

Figure 4 shows the training loss of all the decoder-based language models. All the models have near about the same training loss. We see that the Tülu-2-dpo-70B model gets optimized the most in terms of loss in all the datasets.

8 Conclusion

We fine-tune and examine 2 encoder-based language models, 2 encoder-decoder based language models and 6 decoder-based language models on 10 benchmark scholarly datasets over a span of 6 tasks. In the case of decoder-based language models, we find that there is a huge dissimilarity between the performance achieved and the computational costs involved. We also report the usefulness of fine-tuning and using domain-specific large language models.

Limitations

We do not test over different prompt templates due to computational costs. More language models, including more LLMs and PLMs, can be tested for these tasks. We also do not aim for SOTA results for the tasks we considered. SOTA results sometimes use very specialized techniques that optimize the model for the task.

References

Microsoft Research AI4Science and Microsoft Azure Quantum. 2023. The impact of large language models on scientific discovery: a preliminary study using gpt-4. *Preprint*, arXiv:2311.07361.

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.

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.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. 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 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

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

Jennifer D'Souza and Sören Auer. 2022. Computer science named entity recognition in the open research knowledge graph. *arXiv preprint arXiv:2203.14579*.

Yun He, Zhuoer Wang, Yin Zhang, Ruihong Huang, and James Caverlee. 2020. PARADE: A New Dataset for Paraphrase Identification Requiring Computer Science Domain Knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7572–7582, Online. 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 *Proceedings of the International Conference on Learning Representations,*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Bofeng Huang. 2023. Vigogne: French instruction-following and chat models. `https://github.com/bofenghuang/vigogne`.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *Preprint*, arXiv:2311.10702.

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.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576, Online. Association for Computational Linguistics.

Avishek Lahiri, Pratyay Sarkar, Medha Sen, Debarshi Kumar Sanyal, and Imon Mukherjee. 2024.

Few-TK: A dataset for few-shot scientific typed keyphrase recognition. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4011–4025, Mexico City, Mexico. Association for Computational Linguistics.

Sihang Li, Jin Huang, Jiaxi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. 2024. Scilitllm: How to adapt llms for scientific literature understanding. *Preprint*, arXiv:2408.15545.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Dakota Mahan, Ryan Carlow, Louis Castricato, Nathan Cooper, and Christian Laforte. Stable beluga models.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *Preprint*, arXiv:2402.06196.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26.

Dan Saattrup Nielsen, Kenneth Enevoldsen, and Peter Schneider-Kamp. 2024. Encoder vs decoder: Comparative analysis of encoder and decoder language models on multilingual NLU tasks. *arXiv preprint arXiv:2406.13469*.

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

Muhammad Qorib, Geonsik Moon, and Hwee Tou Ng. 2024. Are decoder-only language models better than encoder-only language models in understanding word meaning? In *Findings of the Association for Computational Linguistics ACL 2024*, pages 16339–16347.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Muhammad Roman, Abdul Shahid, Shafiullah Khan, Anis Koubâa, and Lisu Yu. 2021. Citation intent classification using word embedding. *IEEE Access*, 9:9982–9995.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code llama: Open foundation models for code. *Preprint*, arXiv:2308.12950.

Mobashir Sadat and Cornelia Caragea. 2022. 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.

Mobashir Sadat and Cornelia Caragea. 2024. Co-training for low resource scientific natural language inference. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2538–2550, Bangkok, Thailand. Association for Computational Linguistics.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. 2024. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*.

Noam Shazeer. 2020. GLU variants improve transformer. *CoRR*, abs/2002.05202.

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

Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. 2024. Scieval: A multi-level large language model evaluation benchmark for scientific research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19053–19061.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA. Association for Computational Linguistics.

Juraj Vladika and Florian Matthes. 2024. Comparing knowledge sources for open-domain scientific claim verification. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2103–2114, St. Julian's, Malta. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

David Wadden, Kejian Shi, Jacob Morrison, Aakanksha Naik, Shruti Singh, Nitzan Barzilay, Kyle Lo, Tom Hope, Luca Soldaini, Shannon Zejiang Shen, Doug Downey, Hannaneh Hajishirzi, and Arman Cohan. 2024. Sciriff: A resource to enhance language model instruction-following over scientific literature. *Preprint*, arXiv:2406.07835.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 7th International Conference on Learning Representations*.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024. SciBench: Evaluating College-Level Scientific Problem-Solving Abilities of Large Language Models. In *Proceedings of the Forty-First International Conference on Machine Learning*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *Preprint*, arXiv:2306.04751.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Tong Xie, Yuwei Wan, Wei Huang, Zhenyu Yin, Yixuan Liu, Shaozhou Wang, Qingyuan Linghu, Chunyu Kit, Clara Grazian, Wenjie Zhang, Imran Razzak, and Bram Hoex. 2023. Darwin series: Domain specific large language models for natural science. *Preprint*, arXiv:2308.13565.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

A Dataset Description

A.1 Named Entity Recognition/ Typed Keyphrase Recognition

We make use of the following popular datasets for Named Entity Recognition: SCIERC (Luan et al., 2018), CS-NER (Abstracts) (D'Souza and Auer, 2022), CS-NER (Abstracts) (D'Souza and Auer, 2022). For the Typed Keyphrase Extraction task, we use FEW-TK (Lahiri et al., 2024). Almost all of these datasets are annotated on research paper abstracts or titles or both.

A.2 Relation Classification

We use SCIERC (Luan et al., 2018), which contains about $4,716$ relations over 500 scientific document abstracts.

A.3 Paraphrase Recognition

PARADE (PARAphrase identification based on Domain knowledgE) (He et al., 2020) is a dataset tailored for paraphrase identification consisting of $10,182$ pairs of definitions that describe 788 distinct entities in the Computer Science domain. Out of these, $4,778$ are paraphrases and $5,404$ are non-paraphrases.

A.4 Natural Language Inference

NLI for the scientific domain is relatively new and also quite challenging due to the difference in the vocabulary and sentence structure in comparison to the general domain. SciNLI (Sadat and Caragea, 2022) is a Natural Language Inference (NLI) dataset tailored for the scientific domain, consisting of 101,412 samples in the training set, 2,000 samples in the validation set, and 4,000 samples in the test set. In comparison to traditional datasets, this dataset contains two new classes, taking the total number of classes to four: "Contrasting", "Entailment", "Reasoning" and "Neutral".

A.5 Citation Intent Classification

Citation intents are useful in tasks like the measurement of scientific impact (Cohan et al., 2019) and the temporal study of scientific concepts (Jurgens et al., 2018).

We consider two datasets for this task: ACL-ARC (six categories) (Jurgens et al., 2018) and SciCite (three categories) (Cohan et al., 2019). SciCite consists of $11,020$ instances and is larger than ACL-ARC which contains $1,941$ data points.

A.6 Claim Verification

SCIFACT (Wadden et al., 2020) is a dataset that is made up of $1,409$ expert-written scientific claims which are verified against a corpus of $5,183$ abstracts. The claims in this dataset

B Encoder Model Checkpoints and Experimental Setup

B.1 BERT

BERT (Devlin et al., 2019) stands for Bidirectional Encoder Representations from Transformers. BERT is a multi-layer bidirectional Transformer encoder model that is pre-trained on unlabelled data from the BooksCorpus and English Wikipedia for two different tasks: the masked language modelling (MLM) task and the next sentence prediction (NSP) task. The BERT model may be fine-tuned for several downstream tasks and this fine-tuning paradigm has found success in almost all major NLP tasks.

B.2 SciBERT

SciBERT (Beltagy et al., 2019) is domain-specific variant of BERT that is pre-trained on scientific text. SciBERT retains the architecture as well as all the major characteristics of BERT except that it is pre-trained on a corpus that consists of papers from the biomedical domain and the computer science domain in a $82:18$ ratio.

The experimental details for fine-tuning encoder-based LMs are as follows:

NER/TK: We train the uncased versions of BERT and SciBERT by passing their output through a linear classifier and training using the cross-entropy loss for 20 epochs. The maximum sequence length considered is 256.

REL: This task is formulated for encoder-based LMs as a special case of text classification: the given entities are delineated with special tokens and the model learns to predict the relation between these entities (Beltagy et al., 2019).

PPHRASE: We fine-tune BERT and SciBERT by considering this task as a text classification task as was done for the original PARADE dataset (He et al., 2020). We fine-tune the backbone PLMs for 5 epochs using a learning rate of $2e-5$.

NLI: The pair of sentences provided as input are concatenated separated by a [SEP] token between them. A softmax layer is used to predict the output class from the [CLS] token embedding. Each backbone model is trained for 5 epochs and the

Corpora	Domain	Classes	Papers	Tokens	Entities
SciERC (Luan et al., 2018)	AI	5	500	60,749	8,089
CS-NER (Abstracts) (D'Souza and Auer, 2022)	AI	2	12,271	1,317,256	29,273
CS-NER (Titles) (D'Souza and Auer, 2022)	CL	7	31,044	263,143	67,270
Few-TK (Lahiri et al., 2024)	AI	38	500	115,745	20064

Table 10: Details of standard scientific-domain Named Entity Recognition datasets and Few-TK for Typed Keyphrase Recognition

maximum input length is set at 300. We use the cased versions of the BERT and SciBERT models keeping in line with the original paper (Sadat and Caragea, 2022).

CIC: It is treated as a simple text classification problem given the citation sentence, as in (Beltagy et al., 2019). Therefore, the BERT vector is given as input into a linear classification layer. The learning rate is taken as $2e-5$ and the model is trained for 5 epochs.

CLAIM: We model the claim verification task as a two-class classification problem, such that given the claim-evidence pair, the model predicts whether the claim supports or contradicts the evidence.

C Encoder-Decoder Model Checkpoints and Hyperparameters

T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format.

SciFive (Phan et al., 2021) is a Text-Text framework for biomedical language and natural language in NLP. We use the checkpoint trained on PMC.

For all the text classification datasets i.e. everything except NER and TK, we use a maximum sequence length of 512, a learning rate of 3e-4 and a batch size of 16.

For all the NER and TK datasets, we use a maximum sequence length of 256, a learning rate of 3e-4 and a batch size of 8.

D Decoder Model Checkpoints

We follow QloRA's original hyperparameter settings instead of doing a exhaustive hyperparameter search. We fix both the source length and the target length to 512 for better comprehension. The learning rate is kept at $2e-4$, and we fine-tune each model for $1,875$ steps.

D.1 LLaMA family of models

LLaMA is a family of pre-trained foundational language models that have been open-sourced by Meta in recent times. LLaMA models incorporates the following three minor architectural changes within the original Transformer architecture (Vaswani et al., 2017): (1) use of SwiGLU (Shazeer, 2020) activation function instead of ReLU, (2) use of rotary positional embeddings (Su et al., 2021) instead of absolute positional embedding, and, (3) use of RMSNorm (Zhang and Sennrich, 2019) normalizing function instead of layer-normalization.

D.2 SciLitLLM

SciLitLLM (Li et al., 2024) is a very recently released LLM designed for the task of scientific literature understanding that has been trained using both continual pre-training (CPT) and supervised fine-tuning (SFT).This strategy is used on Qwen2.5 to obtain SciLitLLM. The CPT stage uses 73,000 textbooks and 625,000 academic papers, while the SFT stage uses SciLitIns, SciRIFF (Wadden et al., 2024) and Infinity-Instruct[3]. We use the SciLitLLM 7B[4] for our experimental purposes.

D.3 Tülu family of models

Tülu (Wang et al., 2023) is a set of models that are instruction-tuned on LLaMA (Touvron et al., 2023a) using a mixture of human-generated as well as GPT-generated data. Tülu-2 (Ivison et al., 2023) is trained on LLaMA-2 over a more updated and refined data mixture, which contains even datasets from scientific literature like SciERC (Luan et al., 2018), Qasper (Dasigi et al., 2021), SciFact (Wadden et al., 2020) and SciTLDR (Cachola et al., 2020). Tülu-2 is further trained using the direct preference optimization (DPO) algorithm (Rafailov et al., 2023).

[3] https://huggingface.co/datasets/BAAI/Infinity-Instruct

[4] https://huggingface.co/Uni-SMART/SciLitLLM

E Hallucinated Labels

The following tables show the hallucinated labels in different decoder-based language models.

F Decoder Time Analysis

Model	SciERC (REL)
LLaMA-7B	COMBINATION-STRATEGY -OVER, WEIGHTED-SUM.
LLaMA-13B	-
LLaMA-70B	-
SciLitLLM-7B	INDUCED-FROM
Tulu-2-dpo-7B	-
Tulu-2-dpo-70B	FOR-FOR, SUM-OF, OUT-OF-NLP.

Table 11: Hallucinated Labels for Relation Extraction datasets

Model	ACL-ARC
LLaMA-7B	INSPIRED, TUV
LLaMA-13B	-
LLaMA-70B	-
SciLitLLM-7B	-
Tulu-2-dpo-7B	-
Tulu-2-dpo-70B	REPEATS

Table 12: Hallucinated Labels for Citation Intent Classification datasets

G Prompt Template

Table 15 shows the prompt templates used by the generative decoder-based language models.

Model	Few-TK
LLaMA-7B	‘Data Mining Information Retrieval metrics’, ‘Compute architecture’, ‘Data Mining’ ‘Information Retrieval dataset’, ‘Statistical Mathematical domain’, ‘Statistical Mathematical phenomenon’
LLaMA-13B	‘Astronomy term’, ‘Astronomy term’, ‘Astronomy term’, ‘Astronomy term’, ‘Statistical Mathematical domain’, ‘Statistical Mathematical technique’, ‘Statistical Mathematical domain’, ‘Bioinformatics algorithm tool’
LLaMA-70B	‘Garbage value: Tourism is the typed keyphrase identified from the given text.’, ‘Statistical Mathematical focus’, ‘Statistical Mathematical domain’, ‘New York City dog park’, ‘AI ML DL metrics’
SciLitLLM-7B	‘Reference’, ‘Optimization algorithm tool’, ‘Data Mining Information Retrieval dataset’, ‘AI ML DL library’, ‘Q&A site for programmers’, ‘Commercial LP solver’, ‘Data Mining Information Retrieval dataset’, ‘Miscellaneous result’, ‘Data Mining Information Retrieval strategy’, ’Statistical Mathematical focus’, ’Statistical Mathematical domain’, ‘NLP author’, ‘NLP author’, ‘Information Retrieval focus’, ‘Garbage value: 600 words of type’
Tulu-2-dpo-7B	‘Miscellaneous dataset’, ‘Miscellaneous dataset’, ‘Miscellaneous result’, ‘Statistical Mathematical focus’, ‘Statistical Mathematical focus’ , ‘Data Mining Information Retrieval dataset’, ‘Computer vision algorithm step’, ‘Financial term’, ‘Quality metrics’, ‘Statistical Mathematical focus’, ‘Statistical Mathematical discipline’, ‘author’, ‘author’, ‘Information retrieval focus’, ‘Statistical Mathematical focus
Tulu-2-dpo-70B	‘Application term’, ‘Computer Vision algorithm tool’, ‘Data Mining Information Retrieval tool’, ‘Miscellaneous dataset’, ‘NLP framework’

Table 13: Hallucinated Labels for Typed Keyphrase Recognition dataset, Few-TK

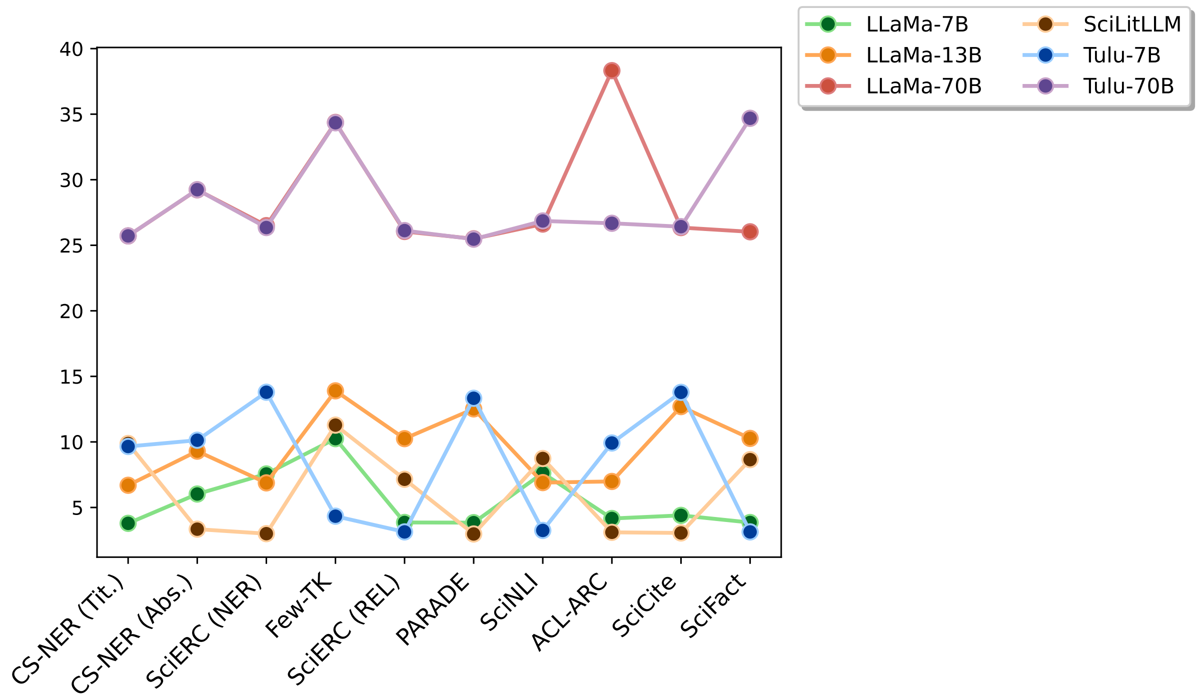

Figure 5: Time taken by decoder-based language models

Model	CS-NER (Titles)	SciERC (NER)
LLaMA-7B	AUTHOR	OBJECTIVE, SCENARIO, AUTHOR
LLaMA-13B	DATE	-
LLaMA-70B	AUTHOR, R, REGION	AUTHOR, HUMAN
SciLitLLM-7B	-	PROFESSION
Tulu-2-dpo-7B	DATE	FUNCTION, AUTHOR
Tulu-2-dpo-70B	DATE, REGION, DATE	USER, PLATFORM, DRUG

Table 14: Hallucinated Labels for Named Entity Recognition datasets

Task	Instruction	Input	Output
Named Entity Recognition	In the given sentence, find the named entity mentions and classify them among the following possible categories - Y	X	The entities s_i of type y_i are identified from the given text.
Typed Keyphrase Recognition	In the given sentence, find the typed keyphrase mentions and classify them among the following possible categories - Y	X	The typed keyphrases s_i of type y_i are identified from the given text.
Relation Extraction	In the given sentence, find and classify the relation between the mentioned pair of named entities, where the relation can be of the following types: Y	X	The relation between s_A and s_B is r.
Paraphrase Recognition	Paraphrases are sentences that express the same meaning by using different wording. Are the following pair of sentences paraphrases or non-paraphrases? SEP separates the two sentences.	(s_1, s_2)	The given pair of sentences are paraphrases/ non-paraphrases.
Natural Language Inference	Analyze the provided pair of sentences to determine their relationship. Choose one of the following categories: Y	(s_1, s_2)	$y \in Y$
Citation Intent Classification	Given a scientific text containing a citation and the citation string, classify the intent of the citation among the following categories: Y.	X	The intent of the citation falls under the following category: $y \in Y$
Claim Verification	Given a scientific claim, evaluate the evidence to determine whether it supports or refutes the claim.	(s_1, s_2)	The given evidence supports/refutes the scientific claim.

Table 15: Table showing prompts used to instruction-tune LLMs