A Simple and Effective Framework for Strict Zero-Shot Hierarchical Classification

Rohan Bhambhoria[*1], **Lei Chen**[2], **Xiaodan Zhu**[1]
[1]Department of Electrical and Computer Engineering & Ingenuity Labs Research Institute
Queen's University, Canada
[2]Rakuten Institute of Technology (RIT)
Boston, MA
{r.bhambhoria,xiaodan.zhu}@queensu.ca
lei.a.chen@rakuten.com

Abstract

In recent years, large language models (LLMs) have achieved strong performance on benchmark tasks, especially in zero or few-shot settings. However, these benchmarks often do not adequately address the challenges posed in the real-world, such as that of hierarchical classification. In order to address this challenge, we propose refactoring conventional tasks on hierarchical datasets into a more indicative long-tail prediction task. We observe LLMs are more prone to failure in these cases. To address these limitations, we propose the use of entailment-contradiction prediction in conjunction with LLMs, which allows for strong performance in a strict zero-shot setting. Importantly, our method does not require any parameter updates, a resource-intensive process and achieves strong performance across multiple datasets.

1 Introduction

Large language models (LLMs) with parameters in the order of billions (Brown et al., 2020) have gained significant attention in recent years due to their strong performance on a wide range of natural language processing tasks. These models have achieved impressive results on benchmarks (Chowdhery et al., 2022), particularly in zero or few-shot settings, where they are able to generalize to new tasks and languages with little to no training data. There is, however a difficulty in tuning parameters of these large-scale models due to resource limitations. Additionally, the focus on benchmarks has led to the neglect of real-world challenges, such as that of hierarchical classification. As a result, the long-tail problem (Samuel et al., 2021) has been overlooked. This occurs when a vast number of rare classes occur in the presence of frequent classes for many natural language problems.

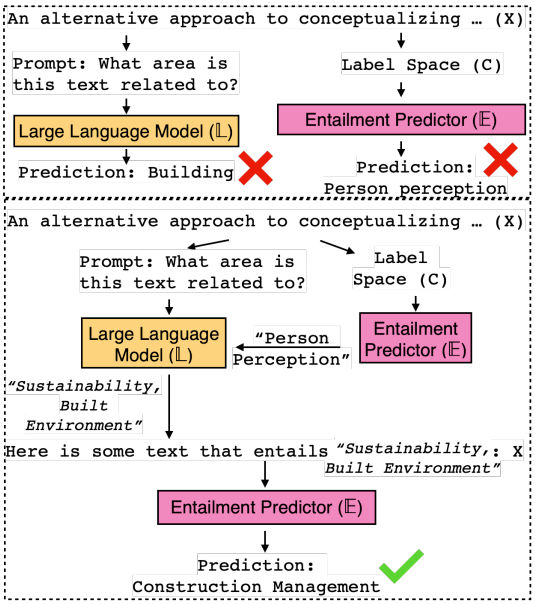

Figure 1: LLMs (𝕃) without any constraints and Entailment Predictors (𝔼) without guided knowledge (**Top**) show poor results independently. Our method (**Bottom**), combines advantages of these two systems to improve performance on strict zero-shot classification.

In many industrial real-world applications, a strong performing method for hierarchical classification can be of direct utility. New product categories are emerging in e-commerce platforms. Existing categories, on the other hand, may not be very intuitive for customers. For example, upon browsing categories such as *night creams*, we may be unable to find a product in a sibling-node category of *creams*. This is further highlighted by platforms in which a systematic structure is not created for users; parent nodes may be in place of child nodes, and vice versa (Asghar, 2016). Manually categorizing product categories can be a costly redesigning endeavour. To tackle this problem, we suggest refactoring traditional hierarchical flat-labeled prediction tasks (Liu et al., 2021) to a more indicative long-tail prediction task. This involves structuring the classification task to closely reflect the real-world long-tail distribution of classes. In

[*]This research was performed when the first author was a research intern at Rakuten Institute of Technology (RIT), Boston.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 1782–1792

doing so, we are enabled to leverage LLMs for long-tail prediction tasks in a strict zero-shot classification setting. Through a series of experiments, results in this work show that our proposed method is able to significantly improve the performance over the baseline in several datasets, and holds promise for addressing the long-tail problem in real-world applications. The contributions of this work can be summarized as follows:

- We refactor real-world hierarchical taxonomy datasets into long-tailed problems. In doing so, we create a strong testbed to evaluate "strict zero-shot classification" with LLMs.
- We explore utilizing LLMs to enhance the capabilities of entailment-contradiction predictors for long-tail classification. This results in strong capabilities of performing model inference without resource-intensive parameter updates.
- We show through quantitative empirical evidence, that our proposed method is able to overcome limitations of stand-alone large language models. Our method obtains strong performance on long-tail classification tasks.

2 Background and Related Work

Strict Zero-Shot Classification

Previous works (Liu et al., 2021; Yin et al., 2019) have explored zero-shot classification extensively under two settings—(i) zero-shot, where testing labels are unseen, i.e. no overlap with the training labels, and (ii) generalized zero-shot, testing labels are partially unseen. In both cases, the model is trained on data from the same distribution as the test data. In our proposed *strict* zero-shot setting, the model is only trained to learn the entailment relationship from natural language inference (NLI) corpora (Williams et al., 2018). The training data for this model has no overlap with the distribution or semantics of the inference set. Additionally, previous works utilizing NLI have either not examined the utility of LLMs (Ye et al., 2020; Gera et al., 2022), or transfer the capabilities of LLMs to smaller models but have failed to use them in a strict zero-shot setting for long-tail problems, only demonstrating their utility for benchmark tasks (Tam et al., 2021; Schick and Schütze, 2021). Works exploring LLMs have also limited their study to only using them independently without exploring entailment-contradiction prediction (Wei et al., 2022; Brown et al., 2020).

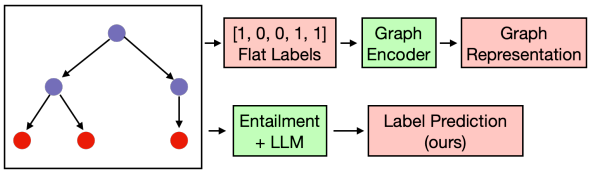

Figure 2: Previous flat label prediction and graph representation tasks (**Top**) do not make use of natural entailment relations. Our method for Label Prediction of red leaf nodes (**Bottom**) enables inference in a *strict* zero-shot setting.

Long Tail Problem

Samuel et al. (2021); Zhang et al. (2022) highlight the significance of addressing the long-tail task. Existing literature in natural language processing has focused on scenarios involving limited data availability, such as few-shot or low-resource settings. It has failed to adequately address the unique challenges presented by long-tail problems. These problems arise when a small number of classes possess a large number of samples, while a large number of classes contain very few samples. Previous works have not delved into the specific use of LLMs or entailment predictors.

Hierarchical Classification

Many real-world problems contain taxonomy data structured in a hierarchical setting. Shown in Figure 2, most previous works make use of this data as a flat-label task (Kowsari et al., 2017; Zhou et al., 2020). It is however, non-trivial to create clean training data for taxonomies, which these methods rely on. This setting also combines parent and child nodes into a multi-label task, thereby increasing the complexity of the problem as siblings amongst leaf nodes are more diverse than parent nodes. Additionally, previous works do not make use of the natural entailment relations in hierarchies. Other works extenuate this problem by opting to utilize flat labels to produce graph representations (Wang et al., 2022a,b; Jiang et al., 2022; Chen et al., 2021). For this reason, the graph representations may have limited value independently, although representations may be used to assist text classification by providing an organized label space. These works only introduce hierarchies to bring order to the label space, but overlook the original task of hierarchical taxonomy classification (Kowsari et al., 2017). For all previous works, difficult to obtain fine-tuning data is required to produce strong sig-

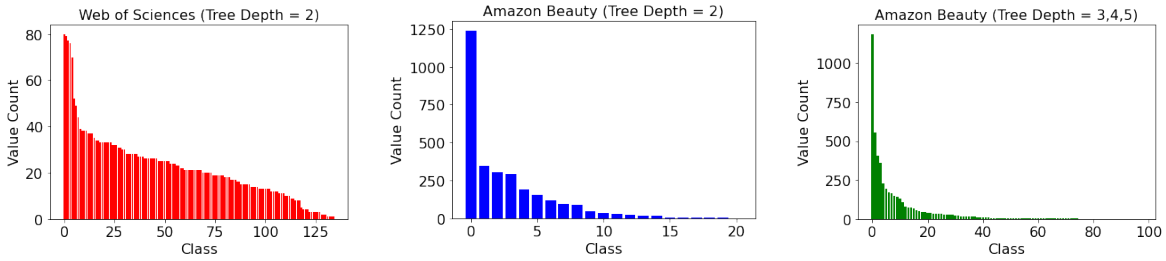

Figure 3: Web of Science (WOS) and Amazon Beauty datasets, refactored to a long-tail distribution. Maximum tree depth is shown for Amazon Beauty, which varies from 3-5. Leaf nodes are used in our method regardless of depth.

nals. In our work, we refactor this data into a leaf-node label prediction task with the help of entailment-contradiction relations and LLMs. In doing so, we enable hierarchical taxonomy prediction independent of utilizing training data for the downstream task.

3 Methodology

In this paper, we investigate the limitations of LLMs in three overlooked settings, when—(i) the model is not provided with sufficient examples due to the input text length, (ii) the label space includes tokens largely unobserved in the model's pretrained vocabulary, and (iii) there are a large number of distractors in the label space (Kojima et al., 2022; Min et al., 2022; Razeghi et al., 2022). These scenarios are common in real-world tasks, but are often overlooked in the development and evaluation of LLMs. To address these challenges, we propose the use of entailment-contradiction prediction (Yin et al., 2019), the task of determining whether a premise logically entails or contradicts a hypothesis. Through our method, we are able to leverage strong reasoning from Yin et al. (2019) with the retrieval abilities of LLMs (Wang et al., 2020) to improve overall performance in a strict zero-shot setting, where the model must classify samples from a new task without any fine-tuning or additional examples used for training from the same distribution as the inference dataset. Importantly, our proposed combined method does not require parameter updates to the LLM, a resource-intensive process that is not always feasible with increasingly large model size (Chowdhery et al., 2022).

Our simple framework is shown in Figure 1. Considering the label space, $\mathcal{C} = \{C_1, C_2, ...C_n\}$ as the set of classes for any given dataset, and a text input, X, we can utilize the entailment predictor, $\mathbb{E}$ to make a *contradiction*, or *entailment* prediction on each label. This is done by using X as the premise, and "This text is related to C_i." $\forall C_i \in \mathcal{C}$ as the hypothesis (Yin et al., 2019). In our work, the premise may be modified to include the prompt template. The prediction, $\mathbb{E}(X)$ lacks any external knowledge and is restricted to the label space, which may result in poor performance. $\mathbb{E}(X)$ can however, provide us with an implicit classification of the contradiction relation for sibling nodes. In our work, we use $\mathbb{E}(X)$ as an initializer for LLMs. We also regard it as a baseline as it shows strong performance independently. A LLM, $\mathbb{L}$ on the other hand, operates in an open space, with aforementioned shortcomings for classification tasks. For our purposes, we can regard this as a noisy knowledge graph (Wang et al., 2020), which may be utilized to predict ancestors or descendants of the target class. We consider the prediction made by the LLM as $\mathbb{L}(X)$. It is important to note that $\mathbb{L}(X)$ may or may not belong to $\mathcal{C}$. We try several prompts for this purpose, shown in Appendix A.

By combining these two components, we can create a template which utilizes the *entailment* relation explicitly, and the *contradiction* relation implicitly by constructing $\mathbb{L}(\mathbb{E}(X))$ to deseminate combined information into an entailment predictor for classification. The template we use is task-dependent, and is generally robust given an understanding of the domain. On Web of Sciences we use: "Here is some text that entails $\mathbb{E}(X)$: X. What area is this text related to?". For Amazon Beauty, we use "Here is a review that entails $\mathbb{E}(X)$: X. What product category is this review related to?". In this setting, our method still meets a barrier due to limitations of LLMs. By constructing a composite function, $\mathbb{E}(\mathbb{L}(\mathbb{E}(X))$, we are able to leverage our LLM in producing tokens which may steer the entailment predictor to correct its prediction. The template used for this composite function is "Here is some text that entails $\mathbb{L}(\mathbb{E}(X))$: X." across all datasets.

General Form: Although our results are reported combining the advantages of $\mathbb{L}$, and $\mathbb{E}$ to produce upto the composite function $\mathbb{E}(\mathbb{L}(\mathbb{E}(X))$, this can

Model	WOS (Tree Depth = 2)		Amzn Beauty (Tree Depth = 2)		Amzn Beauty (Tree Depth = 3, 4, 5)	
	Acc.	Mac.F1	Acc.	Mac.F1	Acc.	Mac.F1
T0pp	10.47	11.04	7.35	6.01	12.04	4.87
BART-MNLI (Baseline)	61.09	68.93	60.80	51.15	41.68	49.35
T0pp + BART-MNLI	20.64	24.01	37.24	24.38	23.47	18.01
BART-MNLI + T0pp	60.40	68.92	58.79	51.94	**43.98**	46.06
BART-MNLI + T0pp (Primed)	60.16	68.81	59.79	**54.04**	39.10	46.50
BART-MNLI + T0pp (Primed+)	**61.78**	**69.48**	**64.25**	52.84	40.79	**49.96**

Table 1: Baseline models (**Top**) underperform our method (**Bottom**) across all datasets for average scores of Top-1, Top-3, and Top-5 accuracy and Macro F1. Our primed and primed+ models explicitly utilize the entailment relation, with one and five predictions of $\mathbb{L}(\mathbb{E}(X))$ respectively. All models used are available on Huggingface.

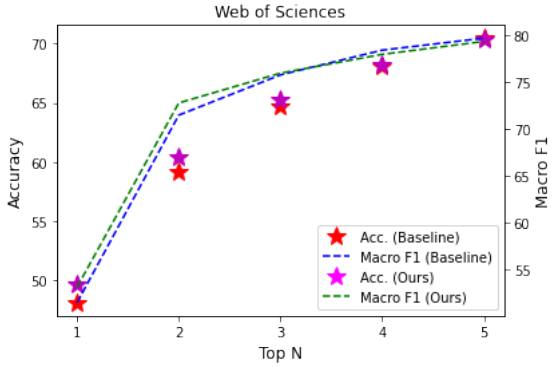

Figure 4: Results for Top-5 predictions on WOS dataset. BART-MNLI + T0pp (Primed+) (**ours**) converges with performance of the BART-MNLI (**baseline**) at Top-4.

be extended as it holds the property of being an iterative composition function to $\mathbb{E}(\mathbb{L}(\mathbb{E}(\mathbb{L}...\mathbb{E}(X))))$. Our observations show this setting having comparable, or marginal improvements with our dataset. However, this may prove to be beneficial in other tasks. We will investigate, and urge other researchers to explore this direction in future work.

4 Experiments and Results

4.1 Dataset and Experimental Settings

We refactor the widely used Web of Sciences (WOS) with Kowsari et al. (2017), and Amazon Beauty (McAuley et al., 2015) datasets to follow a class-wise long-tail distribution as shown in Figure 3. Additionally, we create two variations of the Amazon Beauty dataset, first in which it contains the same tree depth as WOS, both containing 3000 samples, and second in which all classes are included for their maximum tree depth, containing 5000 samples. We select these datasets as they challenge the shortcomings of LLMs. The input text of providing multiple abstracts in the WOS dataset surpasses the maximum input token length of most transformer-based models. This makes it difficult for models to learn the input distribution, a requirement for showing strong in-context performance (Min et al., 2022). Next, many tokens in the label space of both the WOS and Amazon Beauty datasets rarely occur in pretraining corpora, details of which are provided in the Appendix B. Additionally, both datasets contain a large number of distractors, or incorrect classes in the label space. Further details are provided in Appendix C.

All experiments are performed on a single NIVIDA Titan RTX GPU. We use BART-Large-MNLI with 407M parameters as our baseline. We use this model as it outperforms other architectures trained on MNLI for zero-shot classification. For our LLM, we opt to use T0pp (Sanh et al., 2022) with 11B parameters[1], as previous works show that it matches or exceeds performance of other LLMs such as GPT-3 (Brown et al., 2020) on benchmarks.

4.2 Results and Discussion

Results of our method are shown in Table 1. LLMs, due to their limitations, perform poorly as a standalone model for long-tail classification. These results can be improved by priming the model with an entailment predictor through the usage of a prompt. The baseline shows strong performance independent of the LLM, as it operates on a closed label space. The capabilities of the baseline can be enhanced by further explicitly priming it with a entailment relation through a LLM. Rows in which T0pp is initialized, or primed with $\mathbb{E}$ are indicated with *Primed*. Priming the model showcases improvements across all datasets for macro F1. For accuracy, priming the model shows benefit in two out of three datasets. In Figure 4, we show the results of Top-5 predictions for the WOS dataset.

[1]We observe a significant drop in performance when we utilize the 3B parameter variant of this model as $\mathbb{L}$.

All results are aggregated in Table 1. It is important to highlight that prompt variation led to stable results for our LLM. The variance upon utilizing BART-MNLI is negligible across prompts. The best results are observed upto Top-4 predictions on both accuracy and macro F1 for our method, when the entailment prompt is enhanced with a greater number of tokens corresponding to the output of $\mathbb{L}(\mathbb{E}(X))$. The variation between our method and the baseline is much greater for Top-1 predictions, but Top-5 prediction variance is negligible. Detailed results for both depth settings of Amazon Beauty are shown in Appendix C.

5 Conclusion

In this work, we utilize an LLM in the form of a noisy knowledge graph to enhance the capabilties of an entailment predictor. In doing so, we achieve strong performance in a strict zero-shot setting on several hierarchical prediction tasks. We also show the necessity of refactoring existing hierarchical tasks into long-tail problems that may be more representative of the underlying task itself. The utility in practical industry settings is also highlighted through this setting.

Limitations

In this work, we implicitly utilize the *contradiction* relation. The authors recognize explicitly including it in a prompt template leads to worse performance due to the injection of noise. Controlled template generation based on a model confidence is unexplored in this work and appears to be a promising direction. Additionally, we recognize the emergence of parameter-efficient methods for training models which are unexplored in this work, which may have utility. These methods are complimentary and may benefit the performance of models as they can be used in conjunction with training paradigms such as contrastive learning to support better representations through explicit utilization of the *contradiction* relation. In this work, we limit our study to draw attention to the importance of strict zero-shot classification settings with the emergence of LLMs.

Our study can be easily extended to recursively operate on large language models, and entailment predictors. As we observe limited performance benefits in doing so, we conduct our study to show improvements after one complete cycle, given by $\mathbb{E}(\mathbb{L}(\mathbb{E}(X))$ in Section 3.

Ethics Statement

In this work, we propose a framework which allows for the usage of entailment-contradiction predictors in conjunction with large language models. In doing so, this framework operates in a stict zero-shot setting. While it is possible to tune prompts to select optimal variants through hard/soft prompt tuning strategies, this would require additional computational resources for LLMs with billions of parameters. Our study investigates the usage of LLMs given an understanding of the domain they tend to be used for (e.g., given an understanding of Amazon Beauty containing reviews, a prompt is constructed). Further explanation of prompt templates is contained in Appendix A. Due to the lack of tuning parameters in this work, large language models are largely dependent on pre-training data. Although this can be controlled to some degree by introducing an entailment predictor with a fixed label space, the underlying model does not explicitly contain supervision signals without further training. The framework proposed for inference in this work must hence be used cautiously for sensitive areas and topics.

References

Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *CoRR*, abs/1605.05362.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021. Hierarchy-aware Label Semantics Matching Network for Hierarchical Text Classification. 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 4370–4379, Online. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts,

Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. 2022. Zero-Shot Text Classification with Self-Training. ArXiv:2210.17541 [cs].

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface Form Competition: Why the Highest Probability Answer Isn't Always Right. Technical Report arXiv:2104.08315, arXiv. ArXiv:2104.08315 [cs] type: article.

Ting Jiang, Deqing Wang, Leilei Sun, Zhongzhi Chen, Fuzhen Zhuang, and Qinghong Yang. 2022. Exploiting Global and Local Hierarchies for Hierarchical Text Classification. Technical Report arXiv:2205.02613, arXiv. ArXiv:2205.02613 [cs] type: article.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. Technical Report arXiv:2205.11916, arXiv. ArXiv:2205.11916 [cs] type: article.

Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, , Matthew S Gerber, and Laura E Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. IEEE.

Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. 2021. Improving Pretrained Models for Zero-shot Multi-label Text Classification through Reinforced Label Hierarchy Reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1051–1062, Online. Association for Computational Linguistics.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 43–52, New York, NY, USA. Association for Computing Machinery.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*.

Yasaman Razeghi, Robert L. Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of Pretraining Term Frequencies on Few-Shot Reasoning. Technical Report arXiv:2202.07206, arXiv. ArXiv:2202.07206 [cs] type: article.

Dvir Samuel, Yuval Atzmon, and Gal Chechik. 2021. From generalized zero-shot learning to long-tail with class descriptors. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 286–295, Waikoloa, HI, USA. IEEE.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. Technical Report arXiv:2110.08207, arXiv. ArXiv:2110.08207 [cs] type: article.

Timo Schick and Hinrich Schütze. 2021. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. Technical Report arXiv:2009.07118, arXiv. ArXiv:2009.07118 [cs] type: article.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and Simplifying Pattern Exploiting Training. Technical Report arXiv:2103.11955, arXiv. ArXiv:2103.11955 [cs] type: article.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *CoRR*, abs/2010.11967.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022a. Incorporating Hierarchy into Text Encoder: a Contrastive Learning Approach for Hierarchical Text Classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7109–7119, Dublin, Ireland. Association for Computational Linguistics.

Zihan Wang, Peiyi Wang, Tianyu Liu, Binghuai Lin, Yunbo Cao, Zhifang Sui, and Houfeng Wang. 2022b. Hpt: Hierarchy-aware prompt tuning for hierarchical text classification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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.

Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, SuHang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. 2020. Zero-shot text classification via reinforced self-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3014–3024, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. 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 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Chen Zhang, Lei Ren, Jingang Wang, Wei Wu, and Dawei Song. 2022. Making Pre-trained Language Models Good Long-tailed Learners. Technical Report arXiv:2205.05461, arXiv. ArXiv:2205.05461 [cs] type: article.

Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. Hierarchy-Aware Global Model for Hierarchical Text Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1106–1117, Online. Association for Computational Linguistics.

A Prompt Templates

In our work, we try various prompts for WOS and Amazon Beauty to initialize the LLM, and for the entailment predictor. These prompts are shown in Table 2. Initializing prompts for $\mathbb{L}$ may show some variance in performance when utilized independently. The prompts used for obtaining $\mathbb{L}(\mathbb{E}(X))$ are generally robust with an understanding of the domain, and show a marginal impact on outcome, upon variation. Prompts used for $\mathbb{E}(\mathbb{L}(\mathbb{E}(X)))$ have an insignificant impact on the outcome.

B Statistics

We provide some details of the distribution for Web of Science dataset are provided with the head, and tail of the distribution class names with their respective value count in Table 3. We also provide the details of class-wise distribution for Amazon Beauty (Depth=2), and Amazon Beauty (Depth=3,4,5) datasets in Table 4, and Table 5 respectively. Towards the tail-end of the distribution, we observe several tokens which may infrequently appear in most pretraining corpora, such as "Polycythemia Vera" for the WOS dataset. Updating parameters of a model on data which is heavily skewed towards the tail distribution in the presence of frequently occuring labels can be problematic for language models. Our proposed method in this work is one solution towards this challenging task.

C Detailed Results

We provide some details of results for Top-1, Top-3, and Top-5 accuracies and macro F1 scores in this section. The Web of Sciences dataset results are shown in Table 6. We observe that the accuracy is significantly higher by all of our methods over the baseline, BART-MNLI. The same trends are seen for Macro F1 scores. In predicting Top-3 labels, only our method of Primed+ shows improvement over the baseline. For macro F1, our method in the Top-3 category shows slight improvement over the baseline. For Top-5 predictions on the WOS dataset, our method shows performance marginally below the baseline. Results for Amazon Beauty (Depth=2) are shown in Table 7. There is a large improvement in accuracy using our method on this dataset for Top-1, 3, and 5. For Macro F1, there our method performs marginally worse than the baseline for Top-1 predictions. Our method strongly outperforms the baseline by a large margin for Top-3 and Top-3 prediction on Macro F1. The results for Amazon Beauty (Depth=3,4,5) are shown in Table 8. Our method improves upon the baseline for both, accuracy and macro F1 for Top-1 predictions. For Top-3, our method has a significant improvement over accuracy, with comparable performance on Macro F1. Our method has a large improvement on Top-5 scores for accuracy, and improves upon the Macro F1 score for Macro F1.

With our dataset settings, we observe the per-

Dataset	Prompt
WOS	What field is this passage related to? + X What area is this text related to? + X X + What area is this text related to? What area is this text related to? + X
Amazon Beauty	Here is a review: + X + What product category is this review related to? X + What product category is this text related to?

Table 2: Example prompts used to initialize the LLM, L.

formance of using int-8 quantization is robust and matches that of bf-16/fp-32 for inference. These settings also provide us with stable results across prompts.

Previous works have performed parameter-updates (Gera et al., 2022; Holtzman et al., 2021) to models to tackle the challenge of many distractors in the label space. This may be practically infeasible due to the requirements of compute in the case of LLMs.

Diversity between category labels is an important factor we observe which attributes to the improvement in performance. Tables 3, 4, 5 contain statistics for labels used. We observed a significant drop in Macro F1 shown in Table 1 for the Amazon Beauty Dataset (Tree Depth=2) in contrast to WOS for the same models due to the lack of diversity in several class names (e.g. "Bath" and "Bathing Accessories"). Similar trends were observed in Amazon Beauty (Tree Depth=3,4,5) for "Eau de Toilette" and "Eau de Parfum", both of which are perfumes.

Class Name	Value Count
Polymerase chain reaction	95
Northern blotting	88
Molecular biology	66
Human Metabolism	65
Genetics	62
Stealth Technology	2
Voltage law	1
Healthy Sleep	1
Kidney Health	1
Polycythemia Vera	1

Table 3: Class names and corresponding value counts for head and tail elements from for WOS dataset.

Class Name	Value Count
Face	1230
Body	344
Styling Products	298
Women's	289
Styling Tools	187
Bags & Cases	5
Hair Loss Products	5
Bath	3
Bathing Accessories	2
Makeup Remover	1

Table 4: Class names and corresponding value counts for head and tail elements from Amazon Beauty (Depth=2) dataset.

Class Name	Value Count
Lotions	1188
Eau de Toilette	553
Nail Polish	405
Eau de Parfum	363
Soaps	231
Shower Caps	1
Paraffin Baths	1
Hairpieces	1
Tote Bags	1
Curlers	1

Table 5: Class names and corresponding value counts for head and tail elements from Amazon Beauty (Depth=3,4,5) dataset.

Model	Top-1		Top-3		Top-5	
	Acc.	Mac. F1	Acc.	Mac. F1	Acc.	Mac. F1
T0pp	5.46	5.66	11.26	12.25	14.70	15.23
BART-MNLI	48.10	51.49	**64.73**	**75.77**	**70.46**	**79.69**
T0pp + BART-MNLI	12.10	13.75	22.3	26.44	27.53	31.84
BART-MNLI + T0pp	48.16	52.16	63.80	75.40	69.26	79.20
BART-MNLI + T0pp (Primed)	**48.69**	**52.78**	63.60	75.29	68.20	78.37
BART-MNLI + T0pp (Primed+)	**49.73**	**53.15**	**65.23**	**75.96**	**70.39**	**79.34**

Table 6: Accuracy and Macro F1 results for Top-1, Top-3, and Top-5 predictions for the Web of Sciences dataset.

Model	Top-1		Top-3		Top-5	
	Acc.	Mac. F1	Acc.	Mac. F1	Acc.	Mac. F1
T0pp	3.99	2.58	7.48	7.08	10.57	8.37
BART-MNLI	**34.40**	**25.10**	**68.54**	**60.15**	79.45	68.21
T0pp + BART-MNLI	19.87	8.95	39.94	26.30	51.93	37.89
BART-MNLI + T0pp	33.36	**24.84**	61.12	58.63	**81.90**	72.34
BART-MNLI + T0pp (Primed)	**41.22**	24.30	61.46	60.22	76.70	**77.59**
BART-MNLI + T0pp (Primed+)	32.32	19.91	**75.19**	**63.74**	**85.26**	**74.87**

Table 7: Accuracy and Macro F1 results for Top-1, Top-3, and Top-5 predictions for the Amazon Beauty dataset (depth = 2).

Model	Top-1		Top-3		Top-5	
	Acc.	Mac. F1	Acc.	Mac. F1	Acc.	Mac. F1
T0pp	5.22	2.32	13.80	5.54	17.12	6.76
BART-MNLI	**32.58**	**28.05**	43.73	**56.18**	48.75	63.83
T0pp + BART-MNLI	12.49	6.99	26.26	20.64	31.67	26.41
BART-MNLI + T0pp	**33.89**	23.15	**47.06**	53.02	**51.01**	62.02
BART-MNLI + T0pp (Primed)	28.18	20.22	41.89	55.15	47.24	**64.14**
BART-MNLI + T0pp (Primed+)	23.92	**29.70**	**46.43**	**56.07**	**52.02**	**64.11**

Table 8: Accuracy and Macro F1 results for Top-1, Top-3, and Top-5 predictions for the Amazon Beauty dataset (depth = 3,4,5).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations Section

☑ A2. Did you discuss any potential risks of your work?
Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
At the end of the introduction section 1, we provided the paper's main claims. The abstract and introduction summarize them.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3 and 4.

☑ B1. Did you cite the creators of artifacts you used?
Section 2, 3, 4.1

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Figure 3, Section 4.1

C ☑ Did you run computational experiments?

Section 4.1.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.1.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Not applicable. Left blank.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Yes; Fig 3; Fig 4; Table 1; Section 4.2; Appendix A, B, C,

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Table 1 Caption

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.