🤫 Explain Less, Understand More: Jargon Detection via Personalized Parameter-Efficient Fine-tuning

Bohao Wu[♡] **Qingyun Wang**[◇] **Yue Guo**[♡]
[♡]University of Illinois at Urbana-Champaign [◇] William & Mary
{bohaowu,yueg}@illinois.edu qwang16@wm.edu

Abstract

Personalizing jargon detection and explanation is essential for making technical documents accessible to readers with diverse disciplinary backgrounds. However, tailoring models to individual users typically requires substantial annotation efforts and computational resources due to user-specific finetuning. To address this, we present a systematic study of personalized jargon detection, focusing on methods that are both *efficient* and *scalable* for real-world deployment. We explore two personalization strategies: (1) lightweight finetuning using Low-Rank Adaptation (LoRA) on open-source models, and (2) personalized prompting, which tailors model behavior at inference time without retaining. To reflect realistic constraints, we also investigate semi-supervised approaches that combine limited annotated data with self-supervised learning from users' publications. Our personalized LoRA model outperforms GPT-4 with contextual prompting by 21.4% in F1 score and exceeds the best performing oracle baseline by 8.3%. Remarkably, our method achieves comparable performance using only 10% of the annotated training data, demonstrating its practicality for resource-constrained settings. Our study offers the first work to systematically explore efficient, low-resource personalization of jargon detection using open-source language models, offering a practical path toward scalable, user-adaptive NLP system [1].

1 Introduction

Large Language Models (LLMs) are increasingly used to support interdisciplinary research by helping scholars navigate diverse and domain-specific texts (Leto et al., 2024; Ramoneda et al., 2024; Lu et al., 2024; Jiang et al., 2025). However, a persistent barrier to effective interdisciplinary collaboration is the prevalence of domain-specific jargon (Barnett and Doubleday, 2020; Strober, 2006). Researchers often struggle to interpret specialized terminology outside their core expertise, leading to miscommunication (Han et al., 2018; Choi and Pak, 2007), impaired knowledge integration (Lucy et al., 2023), and ultimately slow scientific discovery (Glasziou et al., 2020; Daniel et al., 2022; van Helden et al., 2024). While prior work has developed NLP methods to identify and simplify scholarly jargon using general-purpose corpora like Wikipedia as proxies for reader knowledge (Gardner and Davies, 2013; Tanaka-Ishii and Terada, 2011; Guo et al., 2022, 2021), these approaches remain limited by their lack of personalization. A researcher's background significantly influences their familiarity with domain-specific terms (Gooding and Tragut, 2022; Guo et al., 2024), suggesting that individualized models could more effectively determine which terms require explanation.

To address this challenge, we focus on the task of personalized jargon identification: automatically detecting domain-specific terms that may be unfamiliar to an individual researcher based on their background. Our goal is to make interdisciplinary content more accessible by leveraging LLMs in a personalized, data-efficient, and scalable manner.

Recent efforts in personalized language models, such as LaMP (Salemi et al., 2024), OPPU (Tan et al., 2024b), and Per-Pcs (Tan et al., 2024a), have shown promise by adapting models to user preferences via parameter-efficient fine-tuning (PEFT). However, these methods often rely on costly supervised data or explicit annotation, limiting generalizability. While systems like HLLM (Chen et al., 2024) target personalization task, they do not directly address the broader challenge of personalized language understanding in scholarly context.

Guo et al. (2024) made the first step toward personalized jargon detection by releasing a benchmark and analyzing GPT-4's capabilities. However, their approach depends on costly prompting and rich supervision, raising concerns about scalability

[1] Code can be found at: anonymous GitHub repository.

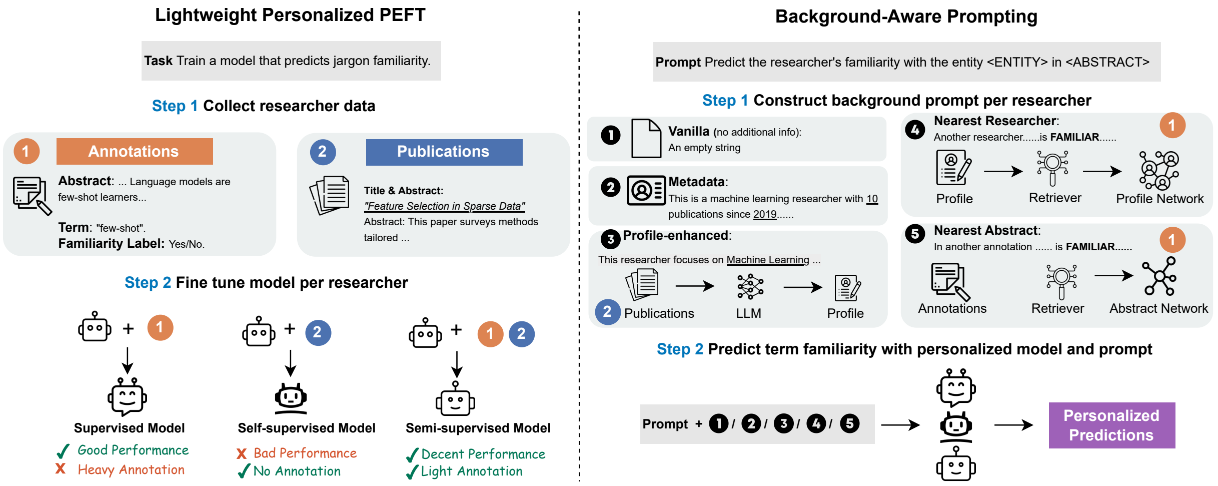
Lightweight Personalized PEFT
Task Train a model that predicts jargon familiarity.
Step 1 Collect researcher data
1 Annotations
Abstract: ... Language models are few-shot learners...
Term: "few-shot".
Familiarity Label: Yes/No.
2 Publications
Title & Abstract:
"Feature Selection in Sparse Data"
Abstract: This paper surveys methods tailored ...
Step 2 Fine tune model per researcher
+ 1
+ 2
+ 1 2
Supervised Model
✓ Good Performance
X Heavy Annotation
Self-supervised Model
X Bad Performance
✓ No Annotation
Semi-supervised Model
✓ Decent Performance
✓ Light Annotation
Background-Aware Prompting
Prompt Predict the researcher's familiarity with the entity <ENTITY> in <ABSTRACT>
Step 1 Construct background prompt per researcher
1 Vanilla (no additional info):
An empty string
2 Metadata:
This is a machine learning researcher with 10 publications since 2019......
3 Profile-enhanced:
This researcher focuses on Machine Learning ...
2 Publications
LLM
Profile
4 Nearest Researcher:
Another researcher......is FAMILIAR......
1
Profile
Retriever
Profile Network
5 Nearest Abstract:
In another annotation ...... is FAMILIAR......
1
Annotations
Retriever
Abstract Network
Step 2 Predict term familiarity with personalized model and prompt
Prompt + 1 / 2 / 3 / 4 / 5
Personalized Predictions

Figure 1: To enable personalized jargon detection, we first fine tune a LoRA-based PEFT model using supervised, self-supervised, and semi-supervised training strategies that reflect real world scenarios with varying levels of annotation availability. Next, we enhance the contextual understanding of the target researcher through a range of background aware prompting methods, including vanilla, metadata based, profile enhanced, nearest researcher, and nearest abstract), to generate personalized familiarity predictions.

and generalization. In this paper, we provide the first comprehensive and systematic study of personalized jargon detection with an emphasis on efficiency, scalability, and low-resource practicality. We show that lightweight PEFT on open-source models can surpass GPT-4 while requiring only 10% annotated data, highlighting the feasibility of scalable, user-adaptive NLP systems.

2 Method

To personalize jargon familiarity, we investigate three fine-tuning settings: supervised, self-supervised, and semi-supervised (§2.2), and incorporate contextual prompts (§2.3). To isolate the impact of personalization, we additionally evaluate a leave-one-annotator-out setting (§2.4).

2.1 Experimental Setup

We use the personalized jargon detection dataset from Guo et al. (2024), which contains 11k term familiarity (i.e., familiar or unfamiliar) annotations provided by 11 computer science researchers for terms extracted from 100 paper abstracts. To our knowledge, it is the only interdisciplinary jargon dataset with high-quality annotations, rich personal metadata, and accompanying published papers from the annotators. We follow the same data split as Guo et al. (2024), dividing the dataset into 60/20/20 for the train, validation, and test sets. We selected `Llama-3.1 8B Instruct 4bit` (Dubey et al., 2024) as our baseline model for personalized jargon detection based on its strong instruction-following ability, low mismatch rate, and competitive performance in preliminary evaluations across several state-of-the-art LLMs. We evaluate model performance using the effective F-1 Score, which adjusts for output format errors by penalizing predictions with a high mismatch rate. Details of hyperparameters are in App. §A.

2.2 Lightweight Personalized PEFT

To ensure consistency across different personalization settings, we adopt the Alpaca format (Taori et al., 2023), which is widely used in instruction fine-tuning. Each training instance is structured into three components: an `Instruction`, an `Input` (target abstract and term), and a `Response` (binary familiarity label). We use this standardized format to unify model interaction across different training strategies. Task-specific instructions and prompt examples are shown in Table 3. We evaluate three training strategies with varying supervision levels:

- *Supervised*: We adopt LoRA (Hu et al., 2022) for PEFT, following findings from Tan et al. (2024b) demonstrating its strong performance. The model is trained to predict annotator familiarity from each term and its associated abstract.
- *Self-supervised*: To simulate low-resource scenarios, we fine-tune on unlabeled titles and abstracts from each annotator's prior publications using a causal language modeling (next-token prediction) objective. For with $\leq$ 5 publications, we augment the corpus with papers from their self-defined subdomain, reflecting practical

cases where personalized models rely on domain-relevant but unlabeled content.

- *Semi-supervised*: We examine a hybrid approach that combines limited labeled data with the annotator's publication corpus, evaluating how much supervision is needed to balance annotation effort with personalization quality and to guide future annotation strategies.

2.3 Background-Aware Prompting

Building on prior work (Guo et al., 2024; Tan et al., 2024b), we design prompting strategies that add varying levels of researcher-specific context to the model input (full prompts in Table 2), differing in both type and detail of background information:

- *Metadata*: Structured features including the annotator's self-defined subfield (e.g., NLP, computer vision), publication count, average references, year of first publication, and the domain of the current abstract. They serve as lightweight indicators of expertise and familiarity.
- *Profile*: Following prior work on user modeling in personalized recommendation (Tan et al., 2024b), we use the baseline model to generate a natural language summary of the annotator's research background based on their metadata.
- *Nearest Annotator*: We use BM25 (Trotman et al., 2014) to retrieve the most similar annotator based on profile text, and use their familiarity labels for the most similar terms as proxy input.
- *Nearest Abstract*: We retrieve the most similar abstract using BM25 and use the target annotator's familiarity labels for its terms as context.

2.4 Ablation Study

To isolate the effect of personalization, we consider two ablation settings. First, we include a *vanilla prompting* setup with no additional contextual information (e.g., no metadata, profiles, or nearest-neighbor retrieval). While still personalized, since the model is fine-tuned on annotator specific data (supervised, self-supervised, or semi-supervised), this setting removes auxiliary background features. Second, we evaluate a *non-personalized baseline* using a leave-one-annotator-out scheme, where the model is trained on data from all annotators except the held-out one. For comparability with the supervised personalized model, we subsample the training data to match the same number of examples, keeping all other parameters identical.

3 Results

We compare against the best results from (Guo et al., 2024), where the oracle uses familiarity ratings from the most similar annotator and GPT-4 prompts include five prior publications. Figure 2(a) shows validation results: supervised models plateau after 20 epochs (reported at 20), while self-supervised models improve more gradually (reported at 50).

Supervised fine-tuning outperforms GPT-4 and oracle settings On the test set (Table 1), vanilla prompting with a supervised personalized model yields the highest F1 (77.9), outperforming GPT-4 with contextual prompting by 21.4% and the oracle baseline by 8.3%. Additional prompting strategies (metadata, profile, nearest annotator, nearest abstract) do not improve over vanilla prompting, suggesting that LoRA fine-tuning itself captures most of the relevant personalized information. The limited gains from these strategies may reflect either noise in background features or insufficient dataset size to reveal their benefits. Overall, these findings highlight the effectiveness of full supervision with PEFT for modeling the link between annotator background knowledge and jargon familiarity, while showing limited added value from more elaborate prompting.

Self-supervised fine-tuning without annotations shows limited effectiveness In the self-supervised setting, models were fine-tuned solely on each annotator's published papers, without familiarity annotations. Although performance improves slightly over time, overall results remain poor, confirming that publication history alone fails to capture familiarity judgments, consistent with prior findings (Haghani, 2023). We exclude the nearest annotator and nearest abstract settings from this experiment, as they require access to annotated familiarity labels and are therefore not applicable in the self-supervised scenario.

Semi-supervised personalization enables efficient adaptation with minimal supervision. Integrating self-supervised user publication data with only 10% of labeled training data yields an F1 of 77.0, nearly matching fully supervised performance (71.9) and clearly surpassing models trained on the same limited labeled data alone (63.6). This demonstrates the value of leveraging unlabeled, domain-relevant data to reduce annotation costs

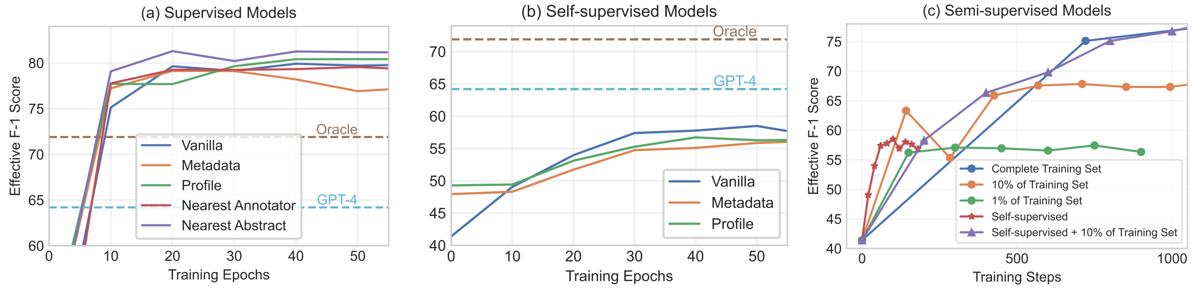

Figure 2: Validation performance of supervised (a), self-supervised (b), and semi-supervised (c) models for jargon familiarity detection. Each point is a personalized model fine-tuned for one of 11 annotators with different prompting strategies. GPT-4 and Oracle baselines are best results from (Guo et al., 2024), where the Oracle uses ratings from the most similar annotator and GPT-4 prompts include five publications. For semi-supervised models, training steps on the x-axis allow consistent comparison across dataset sizes and reflect relative computational cost.

Models	F-1 Score ↑	Recall	Precision
Best results from (Guo et al., 2024)			
Oracle	71.9$_{\pm 1.7}$	76.0$_{\pm 2.1}$	68.2$_{\pm 2.1}$
GPT-4	64.2$_{\pm 1.5}$	98.7$_{\pm 0.5}$	47.6$_{\pm 1.6}$
Supervised			
Vanilla	77.9$_{\pm 1.2}$	77.8$_{\pm 2.2}$	78.0$_{\pm 0.5}$
Metadata	76.8$_{\pm 1.1}$	76.1$_{\pm 2.9}$	77.7$_{\pm 1.0}$
Profile	76.6$_{\pm 1.0}$	73.7$_{\pm 2.0}$	79.9$_{\pm 1.6}$
Nearest Annotator	72.1$_{\pm 3.0}$	70.3$_{\pm 4.1}$	74.5$_{\pm 2.7}$
Nearest Abstract	77.8$_{\pm 1.1}$	78.3$_{\pm 1.7}$	77.5$_{\pm 2.5}$
Self-supervised			
Published Papers	54.6$_{\pm 5.1}$	77.5$_{\pm 7.6}$	45.0$_{\pm 0.8}$
Semi-supervised			
1% Sup	53.5$_{\pm 2.8}$	56.0$_{\pm 4.5}$	51.5$_{\pm 3.2}$
10% Sup	63.6$_{\pm 2.9}$	59.8$_{\pm 5.1}$	69.0$_{\pm 1.7}$
Self + 10% Sup	77.0$_{\pm 1.1}$	78.9$_{\pm 2.6}$	75.4$_{\pm 0.7}$
Leave-one-annotator-out (no personalization)			
Sup baseline	64.7$_{\pm 0.8}$	63.9$_{\pm 1.1}$	65.5$_{\pm 0.6}$
+ Metadata	64.3$_{\pm 0.9}$	62.7$_{\pm 1.5}$	66.1$_{\pm 0.3}$
+ Profile	64.9$_{\pm 0.8}$	64.8$_{\pm 0.8}$	65.0$_{\pm 0.7}$
Self + 10% Sup	53.4$_{\pm 0.1}$	47.8$_{\pm 0.2}$	60.6$_{\pm 0.1}$
+ Metadata	58.6$_{\pm 0.2}$	55.2$_{\pm 0.3}$	62.5$_{\pm 0.2}$
+ Profile	60.4$_{\pm 0.4}$	64.2$_{\pm 0.3}$	57.0$_{\pm 0.4}$

Table 1: Performance of fine-tuned models on the **test** set. Unless specified, results use vanilla prompting. Oracle setting uses the familiarity ratings from the annotator with the highest agreement on the training set for the annotator. For GPT-4, prompts include five annotator's publications. The ± values represent standard deviations across 3 repeated runs to illustrate consistency.

while preserving personalization quality, improving the scalability and accessibility of personalized NLP systems in settings where manual annotation is costly or infeasible. Additional qualitative analyses, including generalization to related tasks, annotator-specific performance, and domain-specific behavior, are provided in App. §B.

Personalization is necessary to solve the jargon detection task The personalized model achieves an F1 of 77.9, which is a 20.4% improvement over supervised leave-one-annotator-out testing (64.7) and a 45.9% improvement over the self-supervised + 10% supervised setting (53.4). While background-aware prompting with metadata or profile does not improve performance over vanilla in the personalized setting, it yields gains without personalized annotation: profile improves F1 by 13.1% (60.4 vs. 53.4) and metadata by 9.7% (58.6 vs. 53.4). These results further underscore the importance of personalization for jargon detection and highlight the effectiveness of PEFT under full supervision, with background-aware prompting offering value only when annotations are limited.

4 Conclusions

In this work, we present a practical and cost-effective approach to personalization in jargon detection. By fine-tuning lightweight language models with LoRA, our method achieves significant performance gains over prior work while maintaining computational efficiency. We further show that personalized prompts grounded in a researcher's background improve non-personalized models for familiarity prediction, providing an alternative when direct annotations are unavailable. Remarkably, our method achieves comparable performance with only 10% of annotated data, underscoring its practicality in resource-constrained settings where large-scale annotation is costly or infeasible. Together, these contributions demonstrate the effectiveness and scalability of personalized NLP, offering a path toward tools that improve accessibility and foster cross-disciplinary collaboration.

Limitations

One limitation of our current work is its reliance on a specific dataset, which is primarily focused on computer science researchers and encompasses a limited number of out-of-domain areas. While this allowed for a controlled evaluation of our personalized techniques, the generalizability of our findings to a broader range of interdisciplinary domains and diverse researcher backgrounds requires further investigation. Future work should explore the application and evaluation of our framework on more heterogeneous datasets that encompass a wider spectrum of academic disciplines and research profiles, to assess its robustness and adaptability in more varied real-world scenarios.

Ethical Considerations

In this paper, we utilized anonymized data from a pre-existing dataset, raising ethical considerations regarding the privacy and responsible use of researcher background information in future implementations. We acknowledge the potential for our personalized models to inherit or amplify biases present in pre-trained models or training data, necessitating careful evaluation across diverse user groups to ensure equitable performance. Furthermore, we recognize the importance of clarifying jargon without oversimplification and the potential for over-reliance on such tools to impact researchers' own interdisciplinary language development. Finally, we advocate for responsible development to prevent unintended consequences like the creation of echo chambers. Ongoing evaluation and community discussion are essential for navigating these ethical complexities.

References

Adrian Barnett and Zoe Doubleday. 2020. The growth of acronyms in the scientific literature. *elife*, 9:e60080.

Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. 2024. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *Computation and Language Repository*.

Bernard C. K. Choi and Anita W. P. Pak. 2007. Multidisciplinarity, interdisciplinarity, and transdisciplinarity in health research, services, education and policy: 2. promotors, barriers, and strategies of enhancement. *Clinical and investigative medicine. Medecine clinique et experimentale*, 30 6:E224–32.

Kristy L Daniel, Myra McConnell, Anita Schuchardt, and Melanie E Peffer. 2022. Challenges facing interdisciplinary researchers: Findings from a professional development workshop. *Plos one*, 17(4):e0267234.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *Computation and Language Repository*.

Dee Gardner and Mark Davies. 2013. A new academic vocabulary list. *Applied Linguistics*, 35(3):305–327.

Paul P Glasziou, Sharon Sanders, and Tammy Hoffmann. 2020. Waste in covid-19 research. *BMJ*, 369.

Sian Gooding and Manuel Tragut. 2022. One size does not fit all: The case for personalised word complexity models. *ArXiv*, abs/2205.02564.

Yue Guo, Joseph Chee Chang, Maria Antoniak, Erin Bransom, Trevor Cohen, Lucy Wang, and Tal August. 2024. Personalized jargon identification for enhanced interdisciplinary communication. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4535–4550, Mexico City, Mexico. Association for Computational Linguistics.

Yue Guo, Wei Qiu, Gondy Leroy, Sheng Wang, and Trevor A. Cohen. 2022. Cells: A parallel corpus for biomedical lay language generation. *ArXiv*, abs/2211.03818.

Yue Guo, Wei Qiu, Yizhong Wang, and Trevor Cohen. 2021. Automated lay language summarization of biomedical scientific reviews. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 160–168.

Milad Haghani. 2023. What makes an informative and publication-worthy scientometric analysis of literature: a guide for authors, reviewers and editors. *Transportation Research Interdisciplinary Perspectives*, 22:100956.

Paul KJ Han, Brian J Zikmund-Fisher, Christine W Duarte, Megan Knaus, Adam Black, Aaron M Scherer, and Angela Fagerlin. 2018. Communication of scientific uncertainty about a novel pandemic health threat: ambiguity aversion and its mechanisms. *Journal of health communication*, 23(5):435–444.

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*.

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

Xue Jiang, Weiren Wang, Shaohan Tian, Hao Wang, Turab Lookman, and Yanjing Su. 2025. Applications of natural language processing and large language models in materials discovery. *npj Computational Materials*, 11(1):79.

Alexandria Leto, Shamik Roy, Alexander Hoyle, Daniel Acuna, and Maria Leonor Pacheco. 2024. A first step towards measuring interdisciplinary engagement in scientific publications: A case study on NLP + CSS research. In *Proceedings of the Sixth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS 2024)*, pages 144–158, Mexico City, Mexico. Association for Computational Linguistics.

Yikang Lu, Alberto Aleta, Chunpeng Du, Lei Shi, and Yamir Moreno. 2024. Llms and generative agent-based models for complex systems research. *Physics of Life Reviews*.

Li Lucy, Jesse Dodge, David Bamman, and Katherine Keith. 2023. Words as gatekeepers: Measuring discipline-specific terms and meanings in scholarly publications. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6929–6947.

Pedro Ramoneda, Emila Parada-Cabaleiro, Benno Weck, and Xavier Serra. 2024. The role of large language models in musicology: Are we ready to trust the machines? In *Proceedings of the 3rd Workshop on NLP for Music and Audio (NLP4MusA)*, pages 81–86, Oakland, USA. Association for Computational Lingustics.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Myra H. Strober. 2006. Habits of the mind: Challenges for multidisciplinary engagement. *Social Epistemology*, 20:315 – 331.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024a. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6459–6475, Miami, Florida, USA. Association for Computational Linguistics.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024b. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6476–6491, Miami, Florida, USA. Association for Computational Linguistics.

Kumiko Tanaka-Ishii and Hiroshi Terada. 2011. Word familiarity and frequency. *ArXiv*, abs/1806.03431.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, ADCS '14, page 58–65, New York, NY, USA. Association for Computing Machinery.

Daniël Paul van Helden, Diane Levine, Eric Guiry, Natalie Darko, Charlotte King, Zahir Hussain, Mukund Janardhanan, Sarah Inskip, and Himanshu Kaul. 2024. Seven recommendations for scientists, universities, and funders to embrace interdisciplinarity: Practical guidelines to enabling interdisciplinarity. *EMBO reports*, 25(7):2832–2836.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

A Setup

Evaluation Metrics To evaluate the performance of our personalized jargon identification models, we focus on predicting binary familiarity labels (0 for familiar, 1 for unfamiliar) for entities extracted from research paper abstracts. Our primary evaluation metric is the F-1 score. However, during our initial baseline model selection phase, we observed that some models struggled to consistently produce the required binary label lists without additional text or nonsensical information. To account for this, we introduced the Effective F-1 Score. This metric incorporates the "Mismatch Rate", the proportion of model outputs that did not conform to the expected binary label format. The Effective F-1 Score is calculated as follows:

$$\text{eff. F-1 score} = (1 - \text{Mis. rate}) \times \text{F-1 score}.$$

Baseline Model Selection To establish a robust foundation for our personalized jargon identification task using Parameter Efficient Fine-Tuning (PEFT), we first selected a suitable open-source Large Language Model (LLM) as our baseline. We evaluated several state-of-the-art options, including Llama-2 (Touvron et al., 2023), Llama-3 (Dubey et al., 2024), Mistral v0.3 (Jiang et al., 2023), and Qwen-2.5 (Yang et al., 2024; Team, 2024), considering both their base and instruction-tuned versions in 4-bit and full precision.

For our implementation, we use the `unsloth` library with all parameters set to their default values, including `is_bfloat16_ supported`. The results of this initial evaluation (depicted in Figure 4) revealed significant performance variations in terms of both F-1 score and mismatch rate. While Qwen-2.5 7B Instruct 4bit achieved the highest Effective F-1 Score (0.54), and Llama-3.1 8B 4bit exhibited the lowest mismatch rate (0.5%), we ultimately selected Llama-3.1 8B Instruct 4bit as our baseline for subsequent fine-tuning experiments. This decision was based on its robust performance (Effective F-1 Score of 0.49) and its demonstrated ability to follow instructions with minimal mismatches, suggesting a strong potential for effective adaptation through PEFT for our personalized jargon identification task.

Implementation Details We fine-tuned the unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit model with a maximum sequence length of 2048 tokens. For parameter-efficient training, we applied LoRA with rank 16, scaling factor (α) 16, dropout 0, and targeted the projection modules (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`). The model was trained for 100 epochs with a per-device batch size of 2 and gradient accumulation of 4 steps, using a learning rate of 2e-4, weight decay of 0.01, and the AdamW (8-bit) optimizer with a linear scheduler and 5 warmup steps. Training was conducted in FP16 or BF16 precision depending on hardware support. Checkpoints were saved every `epoch_size * 10 // 8` steps (approximately every 10 epochs).

B Additional Analysis

B.1 Does the familiarity model generalize over other personalized tasks?

In this part of the experiment, we evaluate whether the finetuned models, trained on familiarity annotations, can generalize to related but unseen tasks. Specifically, we test whether the models can predict annotators' need for additional information (e.g., definitions, background, or examples), a task structurally different from the original familiarity labeling. This setup allows us to examine whether the models have truly internalized the annotators' knowledge levels, or if their performance is simply a result of alignment with the annotation distribution.

Figure 3 demonstrates the strong generalization of our fine-tuned models, achieving performance on definition and background knowledge tasks comparable to prior best Lasso regression models (without explicit fine-tuning on this data) and significantly outperforming them on predicting the need for additional examples. These results suggest that supervised LoRA fine-tuning effectively captures not just annotation patterns but also a robust semantic understanding of the annotators' domain expertise.

B.2 Model Improvement Analysis in Terms of Individual Annotators

Taking annotator #4 as the object, a qualitative analysis of the missed and falsely detected jargon reveals several interesting patterns. The baseline model (trained with 1% training set), while showing some capability in jargon detection, struggled with terms that exhibit a combination of characteristics. Firstly, it frequently failed to identify the terms as jargon that are relatively short and composed of common words but carry highly specific

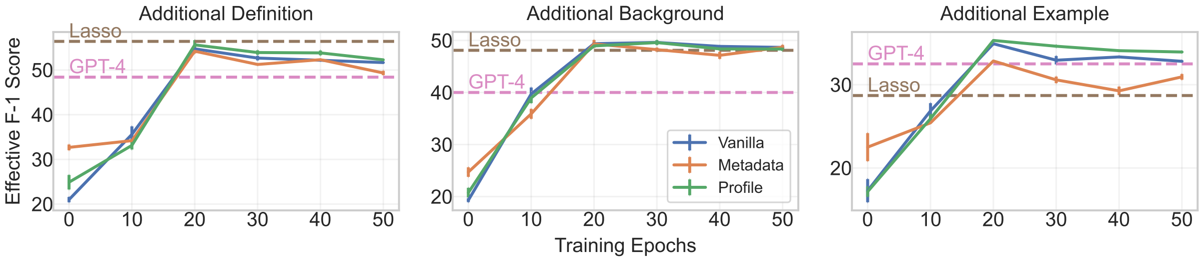

Figure 3: Prediction of additional information needs via various **Supervised** models fine-tuned on the familiarity annotation data. The results of all three sub-figures are evaluated on the validation set. Here, "Lasso" and "GPT-4" denotes the prediction performance of Lasso regression model and GPT-4, respectively. (Guo et al., 2024)

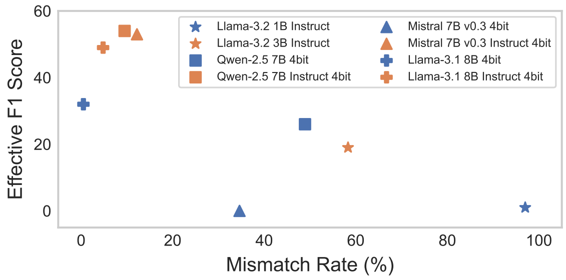

Figure 4: Evaluation results for model selection. Here the inference is done with listed entities to ensure whether the model understands the question.The evaluations are based on the entire dataset.

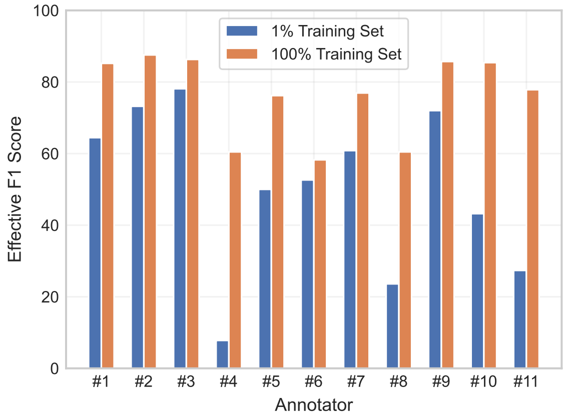

Figure 5: Personalized model Performance for individual annotators.

meanings within a particular domain. Examples include ‘Radial curves’ (Materials Science), ‘Op-amp’ (Physics), and ‘Domains’ (Geology). These terms, due to their brevity and seemingly ordinary components, may have been harder for the baseline to differentiate from general language use. Secondly, the baseline model had difficulty with multi-word terms where the meaning is not a straightforward combination of the individual words, but rather a more nuanced concept. This is evident in its failure to identify ‘Bayesian optimal mechanism’ (Economics), ‘Riemannian framework’ (Materials Science), ‘Bose-Einstein condensate’ (Physics), ‘Psychometric properties’ (Economics), ‘Dialectical quality’ (Philosophy), ‘Explanatory account’ (Linguistics), ‘Long-range ordered coupling’ (Physics, Materials Science), and ‘Qualitative spatio-temporal inferences’ (Psychology). In these cases, the model may have lacked the ability to capture the semantic relationships and contextual dependencies necessary for accurate identification. Thirdly, the baseline also missed acronyms like ‘CW-SSIM’ (Agricultural And Food Sciences), ‘MANOVA’ (Education), and ‘ARMAX model’ (Business, Engineering). Acronyms often present a challenge due to their condensed nature and lack of explicit semantic clues. Finally, there were instances where the jargon term spans multiple disciplines, such as ‘Monolayers’ (Engineering, Biology), ‘Peri-implant bone density’ (Materials Science, Medicine, Biology), and ‘Regulatory mechanisms’ (Biology, Environmental Science), which might have added to the difficulty. While the improved model demonstrated a higher F1 score, indicative of better overall performance, it exhibited a tendency to produce more false positives. These false positives included terms like ‘Savitzky-Golay (SG) filter’ (Environmental Science), ‘Meta-analyses’ (Medicine), ‘Post-test’ (Education), ‘Quantitative research’ (Education), and

'Content analysis' (Medicine). This suggests that the improved model, in its attempt to capture a broader range of jargon, may be more sensitive to terms that share some characteristics with jargon but are more commonly used or understood. This could indicate a trade-off where the improved model sacrifices some precision for increased recall, potentially overgeneralizing in certain contexts. Specifically, the improved model appears to be more prone to misclassifying statistical or methodological terms (e.g., 'Meta-analyses', 'Post-test', 'Quantitative research') as jargon, possibly due to their frequent occurrence in academic contexts, even when they are relatively well understood within the broader research community.

B.3 Model Analysis in Terms of Jargon Domain

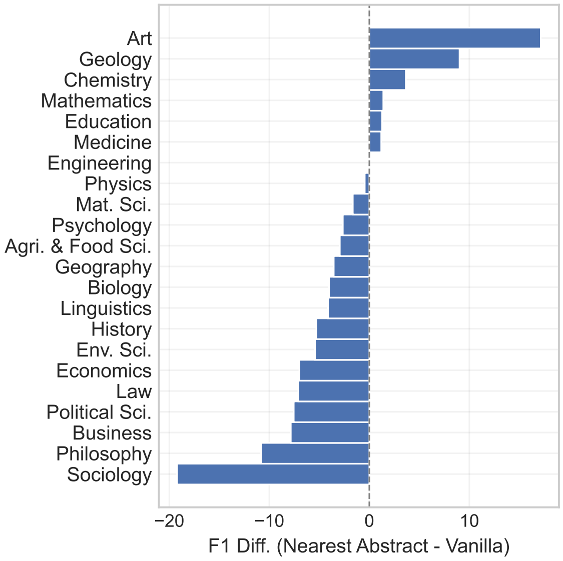

Figure 6: Supervised model performance difference for nearest abstract versus vanilla.

In this study, the two best models are selected, which are 100% TS with vanilla and nearest abstract (NAb) prompting. When comparing the two models across the 'Art' and 'Philosophy' domains, a nuanced performance profile emerges. In the 'Art' domain, the vanilla model exhibits a higher false positive rate, incorrectly identifying terms like 'Reactions' and 'Stylistic' as jargon, whereas the NAb model correctly classifies them. This suggests that vanilla model may be overly sensitive to terms that, while potentially used in art-related contexts, also have broader, common usage.

Conversely, in the 'Philosophy' domain, NAb model faces challenges in both precision and recall. It exhibits a higher false positive rate, misclassifying terms such as 'Structural constraints', 'Analytic philosophers', and 'Argument Facets'. This indicates a tendency to over-identify common philosophical terms as highly specialized jargon. Furthermore, NAb model also demonstrates lower recall in the Philosophy domain, failing to detect several jargon terms, including 'Computational argumentation', 'Corpus with 320 arguments', 'Nonmonotonic inference methods', 'Super-knotty rope', 'Super-knot', and 'Dialectical quality'. These terms represent complex philosophical concepts that the NAb model struggles to recognize as domain-specific jargon.

Strategies	Related data
Vanilla	"" (empty string)
Metadata	"Self-defined subfield of the reader is: { } Number of papers published by the reader is: { } Number of papers referenced by the reader is: { } Year of the first paper published by the reader is: { } Domain of study of the paper is: { }"
Profile-enhancement	"This reader is a domain expert in natural language processing (NLP) ..." (Machine-generated profile)
Nearest annotator	Another researcher similar to the reader has read the same abstract. For the entity list {entity_list}, this researcher provides the familiarity list as {familiarity_list}.
Nearest abstract	For another similar abstract with the entity list {entity_list}, this reader provides the familiarity list as {familiarity_list}.

Table 2: The prompting strategies for both supervised fine-tuning and inference.

Tasks	Instructions	Prompt
Familiarity classification	Your job is to estimate how much the reader knows about an entity. You will be provided with the entity, the abstract where the entity came from, and related data about either the reader or the abstract. Your answer should be a list of binary, either 0 or 1, of the same length as the entity list, with no other words.	Entity: {entity} Abstract: {abstract} Additional information: {related_data} Here's how to gauge the reader's familiarity: - 0: The reader knows this subject well and can describe it to others. - 1: The reader has either encountered this subject before but knows little about it, or has never come across it at all. Based on the information provided, determine familiarity score list corresponding to the entity list, a list of either 0 or 1:
Definition needs classification	You are tasked with predicting whether the reader might need **additional Definition/Explanation** to fully grasp the entities mentioned in a given abstract. You will be provided with the entity list, the abstract where the entities come from, and related data pertinent to the reader or the abstract. Definition of definition/explanation: provides key information on the term independent of any context (e.g., a specific scientific abstract). A definition answers the question, "What is/are [term]?"	Entity: {entity} Abstract: {abstract} Additional information: {related_data} Provide the estimation whether additional information is needed in a list in the order of the entity. The estimation should be either 0(no) or 1(yes). No need to mention the entity:
Background needs classification	You are tasked with predicting whether the reader might need **additional Background/Motivation** to fully grasp the entities mentioned in a given abstract. You will be provided with the entity list, the abstract where the entities come from, and related data pertinent to the reader or the abstract. Definition of background/motivation: introduces information that is important for understanding the term in the context of the abstract. Background can provide information about how the term relates to overall problem, significance, and motivation of the abstract.	Entity: {entity} Abstract: {abstract} Additional information: {related_data} Provide the estimation whether additional information is needed in a list in the order of the entity. The estimation should be either 0(no) or 1(yes). No need to mention the entity:
Example needs classification	You are tasked with predicting whether the reader might need **additional Example** to fully grasp the entities mentioned in a given abstract. You will be provided with the entity list, the abstract where the entities come from, and related data pertinent to the reader or the abstract. Definition of example: offers specific instances that help illustrate the practical application or usage of the term within the abstract.	Entity: {entity} Abstract: {abstract} Additional information: {related_data} Provide the estimation whether additional information is needed in a list in the order of the entity. The estimation should be either 0(no) or 1(yes). No need to mention the entity:

Table 3: The configuration of instructions and prompts for training and inference, following the prompting format from the previous work (Guo et al., 2024).