



Explain Less, Understand More: Data-Efficient Personalization of Reader-Dependent Jargons

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 retraining. 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 systematically explores efficient, low-resource personalization of jargon detection using open-source language models, offering a practical path toward scalable, user-adaptive NLP systems.

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., 2024a) target personalization tasks, they do not directly address the broader challenge of personalized language understanding in scholarly contexts.

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 and generalization. In this paper, we provide

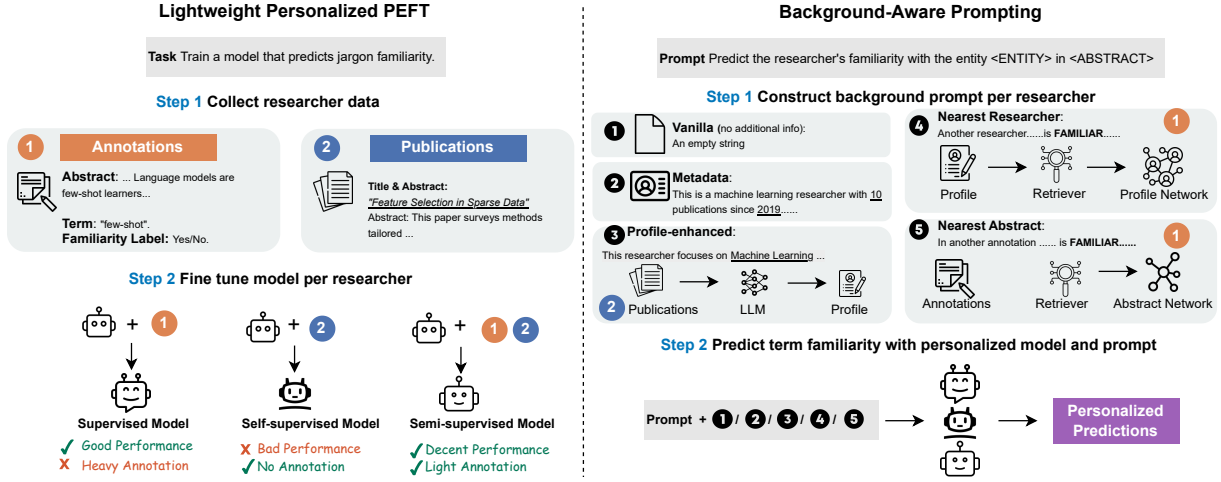


Figure 1: To enable personalized jargon detection, we first fine tune a LoRA-based PEFT model using supervised, self-supervised, and semi-supervised training strategies that reflect real world scenarios with varying levels of annotation availability. Next, we enhance the contextual understanding of the 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.

a 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. 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 pre-

liminary evaluations across several state-of-the-art LLMs. We evaluate model performance using the effective F1 score, which adjusts for output format errors by penalizing predictions with a high mismatch rate. Details of hyperparameters and model selection are in Appendix B and Appendix C.

2.2 Lightweight Personalized PEFT

To ensure consistency across different personalization settings, we adopt the Alpaca format (Taori et al., 2023). 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 Appendix 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 annotators with ≤ 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 Appendix 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.

Models	F1 ↑	Recall	Precision
Best results from (Guo et al., 2024)			
Oracle	71.9±1.7	76.0±2.1	68.2±2.1
GPT-4	64.2±1.5	98.7±0.5	47.6±1.6
Supervised			
Vanilla	77.9±1.2	77.8±2.2	78.0±0.5
Metadata	76.8±1.1	76.1±2.9	77.7±1.0
Profile	76.6±1.0	73.7±2.0	79.9±1.6
Nearest Annotator	72.1±3.0	70.3±4.1	74.5±2.7
Nearest Abstract	77.8±1.1	78.3±1.7	77.5±2.5
Self-supervised			
Published Papers	54.6±5.1	77.5±7.6	45.0±0.8
Semi-supervised			
1% Sup	53.5±2.8	56.0±4.5	51.5±3.2
10% Sup	63.6±2.9	59.8±5.1	69.0±1.7
Self + 10% Sup	77.0±1.1	78.9±2.6	75.4±0.7
Leave-one-annotator-out (no personalization)			
Sup baseline	64.7±0.8	63.9±1.1	65.5±0.6
+ Metadata	64.3±0.9	62.7±1.5	66.1±0.3
+ Profile	64.9±0.8	64.8±0.8	65.0±0.7
Self + 10% Sup	53.4±0.1	47.8±0.2	60.6±0.1
+ Metadata	58.6±0.2	55.2±0.3	62.5±0.2
+ Profile	60.4±0.4	64.2±0.3	57.0±0.4

Table 1: Test-set performance. Unless specified, results use vanilla prompting. Oracle and GPT-4 results follow Guo et al. (2024); ± denotes standard deviation across three runs.

3 Results

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 fa-

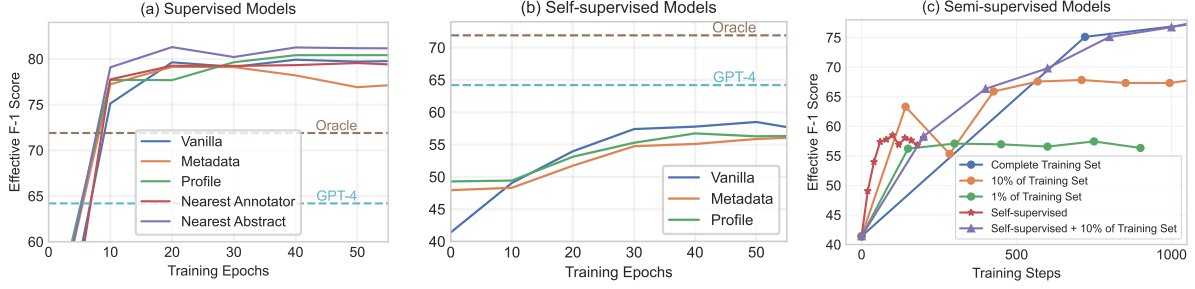


Figure 2: Validation performance of supervised (a), self-supervised (b), and semi-supervised (c) models for jargon familiarity detection. Each point is a personalized model fine-tuned for one of 11 annotators with different prompting strategies. GPT-4 and Oracle baselines are best results from (Guo et 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.

miliarity 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 (77.9) 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 while preserving personalization quality, improving the scalability and accessibility of personalized NLP systems in settings where manual annotation is costly or infeasible.

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 Discussion

Familiarity judgments are inherently reader-dependent rather than interchangeable annotation noise. The leave-one-annotator-out gap shows that a population model cannot recover an individual researcher’s familiarity boundary, even after learning the general task from other researchers. At the same time, publication histories and generated profiles remain imperfect proxies: they help under cold start but cannot reliably replace direct feedback. This motivates personalization systems that request limited calibration data, permit user correction, and avoid treating inferred profiles as authoritative representations of knowledge. Additional transfer, annotator-specific, and domain-specific analyses are provided in Appendix D.

5 Conclusion

In this work, we present a practical and cost-effective framework for personalizing jargon detection in scholarly contexts. By fine-tuning lightweight language models using LoRA, our approach achieves significant performance gains over prior high-resource benchmarks. Our systematic investigation reveals that personalized models, grounded in a researcher’s unique background, can match or exceed the performance of proprietary models like GPT-4 using only 10% of the available annotated data. This underscores the practicality of our method in resource-constrained settings where large-scale human annotation is often infeasible.

Limitations

The benchmark includes only 11 computer-science researchers and 100 abstracts, so the magnitude of the personalization gap may not generalize to other populations or disciplines. The dataset contains candidate terms rather than evaluating end-to-end term extraction. Reader-specific models may also become stale as expertise changes. Broader longitudinal evaluation and explicit uncertainty calibration are needed before deployment.

Ethical Considerations

Professional histories and familiarity judgments can reveal sensitive information about individuals. Systems should use such data with consent, minimize retention, support deletion and correction, and avoid presenting inferred profiles as authoritative. Personalized explanations may also create filter bubbles or uneven access if deployed without user control.

GenAI Disclosure

An AI assistant (ChatGPT) was used solely to improve the grammar of the manuscript. The authors reviewed and edited the output as needed and take full responsibility for the content of this publication.

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. 2024a. [Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling](#). *Computation and Language Repository*.
- Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijun Liu, Song Han, and Jiaya Jia. 2024b. Longlora: Efficient fine-tuning of long-context large language models. In *The Twelfth International Conference on Learning Representations*.
- Bernard C. K. Choi and Anita W. P. Pak. 2007. [Multidisciplinarity, interdisciplinarity, and transdisciplinarity in health research, services, education and policy: 2. promoters, 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 Pepper. 2022. Challenges facing interdisciplinary researchers: Findings from a professional development workshop. *Plos one*, 17(4):e0267234.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 10088–10115.
- 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 *NAACL:2024:long*, 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.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for nlp](#). In *International conference on machine learning*, pages 2790–2799. PMLR.
- 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.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.
- 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 *NLP:2024:1*, pages 144–158, Mexico City, Mexico. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohata, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. [Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning](#). *Advances in Neural Information Processing Systems*, 35:1950–1965.
- 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.
- Taiki Papandreou, Jan Bakker, and Jaap Kamps. 2025. [Medical text simplification: From jargon detection to jargon-aware prompting](#). In *Proceedings of the Fourth Workshop on Text Simplification, Accessibility and Readability (TSAR 2025)*, pages 36–46, Suzhou, China. Association for Computational Linguistics.
- Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yimeng Bai, Wenjie Wang, Hong Cheng, Fuli Feng, and Tat-Seng Chua. 2025. [Measuring what makes you unique: Difference-aware user modeling for enhancing LLM personalization](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21258–21277. Association for Computational Linguistics.
- 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 *NLP4MUSA:2024:1*, pages 81–86, Oakland, USA. Association for Computational Linguistics.
- Alireza Salemi, Sheshera Mysore, Michael Bender-sky, and Hamed Zamani. 2024. [LaMP: When large language models meet personalization](#). In *ACL:2024:long*, 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 *EMNLP:2024:main*, 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 *EMNLP:2024:main*, 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 Full Related Work

A.1 Jargon Detection and Simplification

Traditional jargon identification typically treats "familiarity" as an objective property of a term, utilizing frequency-based metrics or supervised classifiers trained on general-domain proxies like Wikipedia (Gardner and Davies, 2013; Guo et al., 2021). While these methods aid general audiences, they fail to account for the specialized expertise of scholars, where the boundary of "jargon" is highly subjective. Recent advancements have moved toward identifying "technicality" via contextual embeddings (Guo et al., 2022), but these remain static and non-personalized. This lack of flexibility is underscored by a recent study on medical jargon detection, which found that transfer learning across technical datasets yields only modest gains due to divergent labeling objectives, confirming the inherent difficulty of a one-size-fits-all approach to scholarly terminology (Papandreou et al., 2025). Our work, however, shifts the fundamental focus from "what is jargon" to "what is jargon *for this specific user*," acknowledging that linguistic accessibility in science is user-dependent (Gooding and Tragut, 2022).

A.2 Parameter-Efficient Fine-Tuning (PEFT)

As Large Language Models (LLMs) scale, full-parameter fine-tuning becomes computationally prohibitive. PEFT methods address this by updating a minimal subset of parameters while keeping pre-trained weights frozen. Early approaches like *adapter tuning* (Houlsby et al., 2019) introduced learnable layers within feedforward networks, while *prefix* and *prompt tuning* (Li and Liang, 2021; Lester et al., 2021) optimized continuous embeddings to steer model behavior. More recent techniques like (IA)³ (Liu et al., 2022) scale activations within the attention mechanism.

In this work, we utilize *Low-Rank Adaptation* (LoRA) (Hu et al., 2022), which approximates weight updates through low-rank decomposition. The LoRA framework has been extended to handle even more extreme resource and length constraints; for instance, QLoRA (Dettmers et al., 2024) utilizes 4-bit quantization to enable fine-tuning on consumer-grade hardware, while LongLoRA (Chen et al., 2024b) extends the effective context length for adapting to long-form documents. We prioritize standard LoRA over these variants to maintain optimal training efficiency

and performance stability, as our target model and abstract-level context lengths fit within standard GPU memory constraints without the added computational overhead of quantization or context-extension mechanisms.

A.3 Personalization and User Modeling

Personalization has gained traction across several NLP benchmarks, most notably the LaMP benchmark (Salemi et al., 2024). Recent frameworks like *One PEFT Per User* (OPPU) (Tan et al., 2024b) utilize personalized PEFT modules to store user-specific preferences, while *Personalized Pieces* (Per-Pcs) (Tan et al., 2024a) introduces a modular, collaborative approach to enhance scalability. Building on this, Qiu et al. (2025) recently proposed Difference-aware Personalization Learning (DPL), which emphasizes distilling the unique preference information that distinguishes one individual from another by comparing historical user data.

Despite these advances, existing methods primarily rely on high-resource supervised fine-tuning (SFT), which limits scalability in low-resource scholarly settings. Furthermore, while systems like HLLM (Chen et al., 2024a) explore hierarchical personalization, they do not address the specific semantic barriers of technical jargon. We extend this line of research by investigating how semi-supervised and self-supervised PEFT can leverage a researcher’s publication history—an implicit, non-annotated data source—to achieve high-quality personalization without the need for extensive manual labeling.

B Full Methodological Details

To personalize jargon familiarity, we investigate three fine-tuning settings: supervised, self-supervised, and semi-supervised (§B.2), and incorporate contextual prompts (§B.3). To isolate the impact of personalization, we additionally evaluate a leave-one-annotator-out setting (§B.4).

B.1 Experimental Setup

We utilize the personalized jargon detection dataset from Guo et al. (2024), which formalizes the task as an interaction within a **Personalized Knowledge Space** $\mathcal{K} = \mathcal{J} \times \mathcal{R}$. Here, $\mathcal{J}$ represents the set of technical jargons and $\mathcal{R}$ denotes the set of individual researchers. The dataset contains 11k binary familiarity annotations provided by 11 computer science researchers for terms extracted from 100 paper abstracts.

Each interaction $(j, r) \in \mathcal{K}$ is augmented by a context function $\mathcal{C}(\cdot)$, which projects the entities into a high-dimensional attribute space. We define the augmented representation of an interaction as:

$$\mathbf{x}_{j,r} = [\mathcal{C}(j) \oplus \mathcal{C}(r)] \quad (1)$$

where $\mathcal{C}(j)$ encapsulates jargon-specific metadata (e.g., domain, source abstract) and $\mathcal{C}(r)$ represents the researcher’s professional background (e.g., publication history, citation counts). The operator $\oplus$ denotes a concatenation of these contextual features.

In this work, our objective is to learn a mapping $f : \mathcal{K} \rightarrow \mathcal{Y}$ where $\mathcal{Y} = \{“0”, “1”\}$ is the set of target string tokens. Specifically, the model must output the label y such that:

$$y = f(\mathbf{x}_{j,r}; \Theta),$$

$$y \in \begin{cases} “0” & \text{if } r \text{ is familiar with } j \\ “1” & \text{if } r \text{ is unfamiliar with } j \end{cases} \quad (2)$$

where Θ represents the parameters of the personalized models. This formulation treats personalization as a direct text-to-text generation task conditioned on the joint knowledge space.

We follow the data split established by Guo et al. (2024), dividing the interaction pairs into 60/20/20 for the train, validation, and test sets. For our base model, we selected Llama-3.1 8B Instruct 4bit (Dubey et al., 2024). This choice was informed by preliminary evaluations across several state-of-the-art LLMs, where Llama-3.1 8B Instruct demonstrated the strongest instruction-following ability and the most competitive baseline performance, which is discussed in Appendix C.

Evaluation Metrics Our primary objective is to predict the discrete familiarity label y for a given jargon-researcher pair. While we report the standard F-1 score, we also address the practical challenge of model reliability. During our initial baseline selection, we observed that certain models failed to strictly adhere to the binary output format. To account for this, we utilize the **Effective F-1 Score**, which scales the classification performance by the model’s reliability:

$$\text{eff. F-1} = (1 - \text{Mis. Rate}) \times \text{F-1} \quad (3)$$

where the “Mismatch Rate” represents the proportion of model outputs that did not conform to the expected format.

B.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 ≤ 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.

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 $\text{epoch_size} * 10 // 8$ steps (approximately every 10 epochs).

B.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.

B.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.

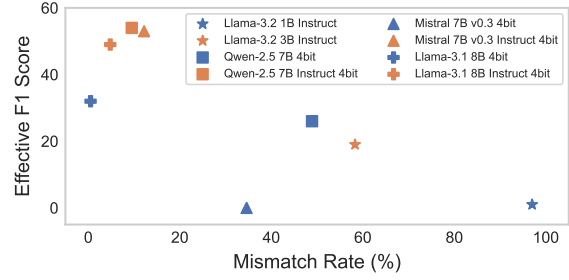


Figure 3: Evaluation results for model selection. Here the inference is done with listed entities to ensure whether the model understands the question. The evaluations are based on the entire dataset.

C 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 3) 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.

D Full Discussion and Analysis

While our primary results demonstrate the quantitative superiority of personalized PEFT, a robust understanding of user-adaptive systems requires a deeper investigation into their qualitative behavior and generalization capabilities. In this section, we move beyond aggregate metrics to analyze our

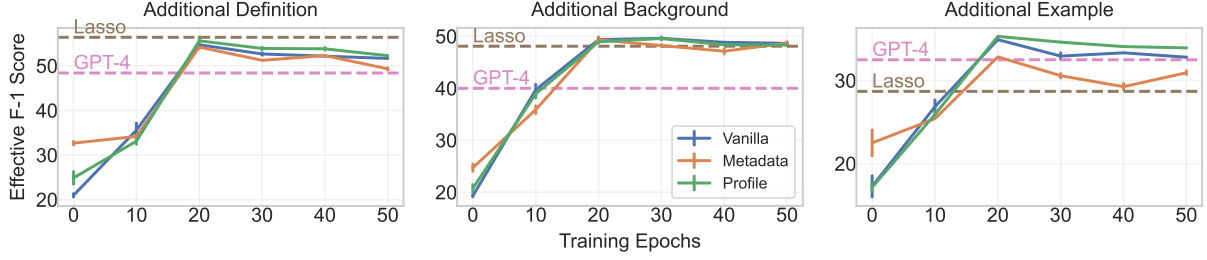


Figure 4: Prediction of additional information needs via various **Supervised** models fine-tuned on the familiarity annotation data. The results of all three sub-figures are evaluated on the validation set. Here, "Lasso" and "GPT-4" denotes the prediction performance of Lasso regression model and GPT-4,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

framework from three critical perspectives. First, we evaluate the robustness of the learned expertise profiles by testing their generalization to unseen downstream information tasks (§D.1). Second, we conduct a granular error analysis of individual researcher profiles to identify the linguistic patterns where personalization offers the most significant gains over baseline models (§D.2). Finally, we examine the stability of the model’s familiarity boundary across disparate technical domains, highlighting the trade-offs between contextual grounding and semantic noise (§D.3).

D.1 Generalization to Downstream Information Needs

To determine if our fine-tuned models have internalized a robust representation of researcher expertise, we evaluate their performance on a structurally distinct generalization task. Beyond binary familiarity, the dataset includes annotations for *Information Requirements*—specific types of auxiliary content a researcher r requires to fully comprehend a jargon j .

We formalize this as a multi-label classification task over the same interaction space $\mathcal{K}$. Let $\mathcal{I} = \{i_{def}, i_{bg}, i_{ex}\}$ represent the set of information needs, corresponding to definitions, background knowledge, and illustrative examples, respectively. Using the previously defined augmented representation $\mathbf{x}_{j,r}$, we evaluate whether the parameters Θ learned during familiarity training can generalize to the mapping g :

$$\mathbf{y}_{info} = g(\mathbf{x}_{j,r}; \Theta), \quad \mathbf{y}_{info} \in \{0, 1\}^{|\mathcal{I}|} \quad (4)$$

where each element in $\mathbf{y}_{info}$ indicates the necessity of a specific information type.

Figure 4 illustrates the performance of our models in this zero-shot or low-resource transfer setting. Notably, our fine-tuned LoRA models achieve performance on definition and background knowledge

tasks comparable to specialized Lasso regression baselines. Furthermore, our approach significantly outperforms all baselines in predicting the need for illustrative examples (i_{ex}).

These results suggest that the mapping f learned in our primary experiments is not merely a distribution-matching artifact. Instead, the model effectively constructs a **Latent Expertise Profile** that remains consistent across different modalities of personalized assistance. This demonstrates that by optimizing for $P(y | j, r)$, the model captures the underlying semantic boundary between a researcher’s current knowledge and the technical requirements of the text.

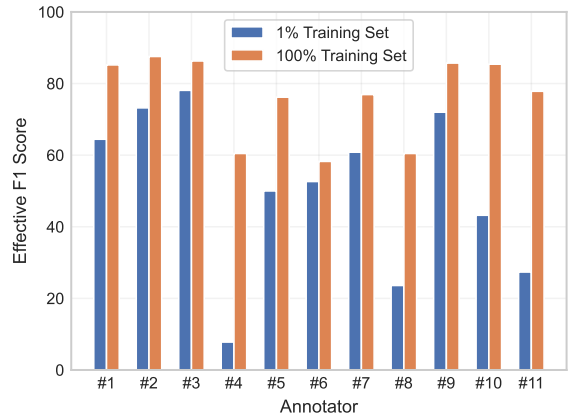


Figure 5: Personalized model Performance for individual annotators.

D.2 Model Improvement Analysis in Terms of Individual Annotators

To characterize the evolution of the predictive mapping $f(\mathbf{x}_{j,r})$ during fine-tuning, we perform a qualitative error analysis focusing on Annotator #4—a researcher with a multi-disciplinary background. We compare the baseline model (trained on 1% of $\mathcal{D}$) against our optimized LoRA-tuned model Θ .

D.2.1 Baseline Failure Modes

The baseline model primarily struggles with regions of the jargon space $\mathcal{J}$ where semantic ambiguity is high. We categorize these failures into four distinct linguistic patterns:

1. **Polysemous Common Words:** Terms that are brief and composed of general vocabulary but carry high specificity in a technical context (e.g., *Radial curves* in Materials Science, *Domains* in Geology, and *Op-amp* in Physics). The model often fails to differentiate these from general language use.
2. **Non-Compositional Multi-word Expressions (MWEs):** Complex terms where the meaning is not a straightforward combination of individual words (e.g., *Bayesian optimal mechanism*, *Riemannian framework*, *Bose-Einstein condensate*). The baseline lacks the capacity to capture the deep semantic dependencies within these units.
3. **Opaque Acronyms:** Condensed tokens lacking explicit semantic clues (e.g., *CW-SSIM*, *MANOVA*, *ARMAX model*). These require the model to leverage the researcher’s background $\mathcal{C}(r)$ more heavily than textual features.
4. **Interdisciplinary Overlap:** Terms spanning multiple fields (e.g., *Monolayers*, *Regulatory mechanisms*), which increase the complexity of the familiarity boundary.

D.2.2 The Recall-Precision Trade-off

While the improved model (Θ) demonstrates a higher F_1 score and significantly improved recall, we observe a characteristic shift toward **False Positives** ($y = 1$ predictions for familiar terms). This phenomenon occurs primarily with high-frequency methodological or statistical terms such as *Meta-analyses*, *Post-test*, and *Quantitative research*.

This suggests that as the model optimizes for the "unfamiliarity" label to capture a broader range of jargon, it becomes hypersensitive to academic-standard terminology. The model essentially conflates *scholarly frequency* with *personal difficulty*. This indicates a strategic trade-off where the system prioritizes identifying potential knowledge gaps ($y = 1$) at the cost of flagging terms that are relatively well-understood within the broader scientific community.

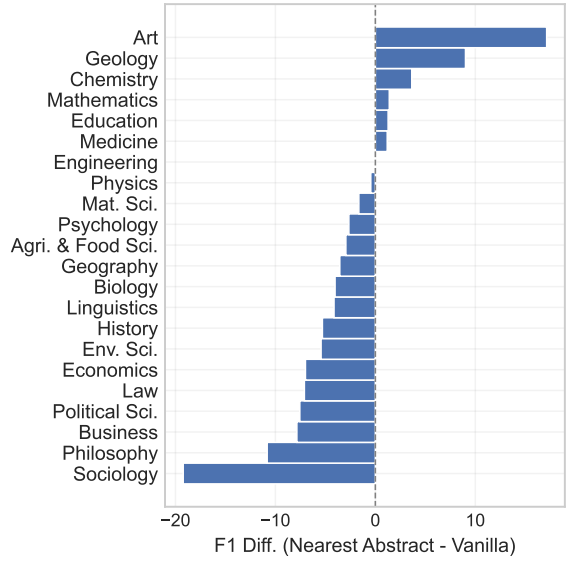


Figure 6: Supervised model performance difference for nearest abstract versus vanilla.

D.3 Model Analysis in Terms of Jargon Domain

To understand the impact of contextual information on model stability, we compare our two highest-performing configurations: **Vanilla Prompting** and **Nearest Abstract (NAb) Prompting**, both trained with 100% supervision. We analyze their performance across the Art ($\mathcal{J}_{Art} \subset \mathcal{J}$) and Philosophy ($\mathcal{J}_{Phil} \subset \mathcal{J}$) domains to evaluate how the context function $\mathcal{C}(j)$ influences the familiarity boundary.

D.3.1 The Art Domain: Mitigation of Over-Sensitivity

In $\mathcal{J}_{Art}$, the Vanilla model exhibits a significantly higher false positive rate, incorrectly labeling high-frequency terms such as *Reactions* and *Stylistic* as jargon ($y = 1$). In contrast, the NAb model—which incorporates the source abstract into $\mathcal{C}(j)$ —correctly identifies these as familiar terms. This suggests that without contextual grounding, the model is overly sensitive to terms that possess both a technical use-case and a common-usage baseline. The inclusion of the nearest abstract provides a semantic anchor that prevents the model from misclassifying general vocabulary as domain-specific jargon.

D.3.2 The Philosophy Domain: Contextual Confusion and Recall Gaps

Conversely, in the more abstract domain of $\mathcal{J}_{Phil}$, the NAb model faces a dual challenge in precision

and recall:

1. **Precision Decay:** The NAb model demonstrates a tendency toward "Over-Identification," misclassifying foundational philosophical terms like *Structural constraints*, *Analytic philosophers*, and *Argument Facets* as unfamiliar.
2. **Recall Deficiency:** Surprisingly, the NAb model fails to detect highly specialized terms such as *Nonmonotonic inference methods*, *Super-knotty rope*, and *Dialectical quality*.

These results indicate that in highly abstract or linguistic-heavy fields like Philosophy, the context function $\mathcal{C}(j)$ may introduce noise rather than clarity. While the NAb approach excels in grounding concrete terms (as seen in Art), it struggles to model the expertise boundary for complex, nested philosophical concepts. This suggests that for certain subspaces of $\mathcal{J}$, the interaction between jargon and researcher requires a more sophisticated profile of the researcher's specific sub-domain expertise $\mathcal{C}(r)$ rather than just textual proximity to an abstract.

E Complete Prompt Configurations

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. A definition provides key information on the term independent of any context.	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. Background introduces information important for understanding the term in the context 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. An 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).