PromptLab: A Collaborative Platform for Prompt Engineering and Dataset Curation

Maged S. Al-shaibani[1] Zaid Alyafeai[2] Dania Refai[3] Nawaf Alomari[3] Ahmed Ashraf[3]
Mais Alheraki[3] Mustafa Alturki[4] Hamzah Luqman[1,3] Irfan Ahmad[1,3]
[1]SDAIA-KFUPM JRC for AI [2]KAUST [3]KFUPM [4]HUMAIN AI
{majed.alshaibani,irfan.ahmad,hluqman}@kfupm.edu.sa
zaid.alyafeai@kaust.edu.sa, malturki@thefutureai.co
{g202391270,g201931050,g202411740,g202401480}@kfupm.edu.sa

Abstract

PromptLab is a web-based platform for collaborative prompt engineering across diverse natural language processing tasks and datasets. The platform addresses primary challenges in prompt development, including template creation, collaborative review, and quality assurance through a comprehensive workflow that supports both individual researchers and team-based projects. PromptLab integrates with HuggingFace and provides AI-assisted prompt generation via OpenRouter[1], and supporting real-time validation with multiple Large Language Models (LLMs). The platform features a flexible templating system using Jinja2, role-based project management, peer review processes, and supports programmatic access through RESTful APIs. To ensure data privacy and support sensitive research environments, PromptLab includes an easy CI/CD pipeline for self-hosted deployments and institutional control. We demonstrate the platform's effectiveness through two evaluations: a controlled comparison study with six researchers across five benchmark datasets and 13 models with 90 prompts; and a comprehensive case study in instruction tuning research, where over 350 prompts across 80+ datasets have been developed and validated by multiple team members. The platform is available at https://promptlab.up.railway.app and the source code is available on GitHub at https://github.com/KFUPM-JRCAI/PromptLab.

1 Introduction

Prompt engineering is a fundamental technique for effectively utilizing large language models across diverse natural language processing tasks (Minaee et al., 2024). The practice of designing natural language instructions to guide model behavior has proven notable for achieving performance gains in zero-shot and few-shot settings. However, the process of creating, refining, and managing prompts at scale presents significant challenges, particularly for collaborative research environments where consistency, quality, and reproducibility are paramount (Schulhoff et al., 2024b; Sahoo et al., 2024a).

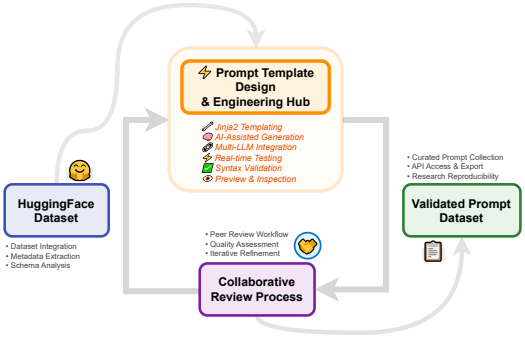

Figure 1: PromptLab General Pipeline

Recent comprehensive surveys have catalogued over 58 distinct prompting techniques (Schulhoff et al., 2024a) and established systematic taxonomies for prompt engineering methodologies (Sahoo et al., 2024b), underscoring the maturation of prompt engineering as a research discipline while revealing the complexity and diversity of approaches required for different tasks and domains. The growing sophistication of prompting techniques demands platforms that can support collaborative development, systematic evaluation, and reproducible research practices. As research teams increasingly work across institutional boundaries and language barriers, the need for a comprehensive collaborative environment becomes even more demanding.

The challenges facing prompt engineering research extend beyond individual technical difficulties to encompass broader issues of research coordination, quality control, and knowledge sharing. Current approaches to prompt development often rely on ad-hoc methodologies and informal sharing mechanisms that limit reproducibility and hinder

[1]https://openrouter.ai/

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 3: System Demonstrations, pages 225–260

scientific progress. The lack of standardized workflows for collaborative prompt development has led to duplicated efforts, inconsistent quality standards, and missed opportunities for cross-institutional research collaboration.

Existing prompt engineering tools primarily focus on individual use cases and lack comprehensive support for collaborative workflows. While platforms like PromptSource (Bach et al., 2022) have demonstrated the value of structured prompt creation environments, they often fall short in providing the project management, peer review, and real-time evaluation proven useful for large-scale research initiatives. PromptLab addresses these limitations by providing a comprehensive platform designed specifically for collaborative prompt engineering. It combines flexible prompt template development with robust project management features. The platform's design philosophy centers on three core principles. First, prompt creation should be accessible to researchers with varying technical backgrounds and different domains while maintaining the expressiveness necessary for complex tasks. This requires carefully designed interfaces that maintain a balance between simplicity and templating power. Second, collaborative prompt development requires advanced workflow management that goes beyond simple sharing mechanisms to include role-based access control, systematic review processes, and progress tracking. Third, quality assurance in prompt engineering demands automated validation and human expertise, live testing with different LLMs, and an iterative refinement workflow.

2 Background & Related Work

The development of PromptLab builds upon extensive research in prompt engineering and instruction tuning. Below, we briefly survey this literature.

2.1 Prompt Engineering and Optimization Techniques

Prompt engineering has rapidly evolved from basic few-shot learning (Brown, 2020) into advanced optimization frameworks (Schulhoff et al., 2024a; Sahoo et al., 2024b). Chain-of-Thought prompting (Wei et al., 2022b) demonstrated significant reasoning improvements, leading to advanced variants including Self-Consistency (Wang et al., 2022a), Tree-of-Thoughts (Yao et al., 2023), and Plan-and-Solve approaches (Wang et al., 2023a).

Automated prompt optimization evolved as an exciting research direction. The Automatic Prompt Engineer (APE) framework (Zhou et al., 2022) treats instruction generation as natural language synthesis, discovering superior zero-shot prompts compared to human-engineered alternatives. Optimization by PROmpting (OPRO) (Yang et al., 2023) leverages language models as optimizers, achieving up to 50% improvements on Big-Bench Hard tasks through iterative refinement. PromptAgent (Wang et al., 2023b) introduces strategic planning via Monte Carlo Tree Search, systematically exploring expert-level prompt spaces through domain knowledge integration and error feedback mechanisms. Another notable work in this direction are DSPy (Khattab et al., 2023). GreaterPrompt (Zheng et al., 2025) is another prompt optimization-based work that integrates diverse techniques under a customizable API, supporting both text feedback-based and gradient-based optimization across different model scales.

On another dimension, interactive prompt development has been explored through visual interfaces. (Strobelt et al., 2022) introduced PromptIDE for real-time experimentation and performance visualization, while (White et al., 2023) developed comprehensive prompt pattern catalogs analogous to software design patterns. Wordflow (Wang et al., 2024) introduces a social prompt engineering paradigm, enabling users to collaboratively create, share, and discover prompts within a community-driven platform. Prompterator (Sučik et al., 2023) provides a human-in-the-loop environment for iteratively refining prompts based on human feedback. These works establish precedents for user-friendly collaborative prompt development.

Another direction focusing on parameter-efficient methods includes HyperPrompt (He et al., 2022), which uses HyperNetworks(Ha et al., 2016; Chauhan et al., 2024) to generate task-conditioned prompts for multi-task learning with only 0.14% additional parameters. Complementary techniques include AutoPrompt (Shin et al., 2020) for gradient-guided prompt search, Prefix Tuning (Li and Liang, 2021), and Prompt Tuning (Lester et al., 2021) for continuous prompt optimization, alongside Prompt-OIRL (Zhang et al., 2023) applying reinforcement learning principles to query-dependent prompt generation.

2.2 Instruction Tuning Datasets and Collaborative Infrastructure

Large-scale datasets have been fundamental to advancing instruction tuning research. P3 (Public Pool of Prompts) (Bach et al., 2022) established template-based approaches with 2,000+ prompts across 270+ English datasets, directly influencing PromptLab's design. The multilingual extension xP3 (Muennighoff et al., 2022) spans 46 languages and 16 NLP tasks, enabling cross-lingual models like BLOOMZ and mT0.

Super-NaturalInstructions (Wang et al., 2022c) provides 1,616 diverse tasks with expert-written instructions across 76 task types, establishing evaluation frameworks for cross-task generalization. BIG-Bench (Srivastava et al., 2022) represents collaborative benchmark development across 450+ authors and 132 institutions, providing architectural insights for community-driven platforms while focusing on tasks beyond current model capabilities.

Instruction tuning foundations began with InstructGPT (Ouyang et al., 2022) and evolved through the Flan family (Wei et al., 2022a; Chung et al., 2024) and T0 (Sanh et al., 2021), demonstrating unified multitask approaches. Synthetic data generation through Self-Instruct (Wang et al., 2022a) and cost-effective approaches like Alpaca (Taori et al., 2023) complement human-curated datasets, while quality-focused methods (Zhou et al., 2024) emphasize careful curation over quantity.

2.3 The Landscape of Prompt Engineering Tools

The tooling ecosystem of prompt engineering evolves rapidly to meet diverse needs from research and development domains. This remains a very active area of research and development, with new tools and frameworks appearing regularly as the field matures (Wei et al., 2022b; Brown, 2020).

Current tools can be broadly categorized into two main paradigms: web-based platforms and command-line interface (CLI) tools. Web-based platforms are particularly popular among downstream LLM application developers, those building web and mobile applications that interact with LLMs through APIs and primarily work with direct API integrations. These include playground environments backed by LLMs providers like **OpenAI Playground**[2] and **Anthropic Console**[3], as well as collaborative platforms designed for prompt development and testing. Table 1 presents and compares the features of PromptLab compared to other web-based platforms and tools.

CLI-based tools, while also popular among developers, tend to attract researchers and those who translate research into usable toolkits for broader adoption. This category includes frameworks like **OpenPrompt** (Ding et al., 2021), **DSPy** (Khattab et al., 2023), **GreaterPrompt** (Zheng et al., 2025), **LangChain**[4], **LlamaFactory(Zheng et al., 2024)**, **LlamaIndex**[5], **Mirascope**[6], and **promptwright**[7]. These tools often emphasize programmatic approaches when interacting with LLMs, including prompt engineering, treating it as a software engineering discipline with modules, signatures, and systematic optimization approaches.

PromptLab distinguishes itself within this landscape by focusing on collaboration-oriented research workflows. Unlike existing tools that primarily target application developers and individual prompters, PromptLab is designed specifically for the research community, emphasizing collaborative prompt development, systematic evaluation, and reproducible research practices. This positioning addresses a gap in the current ecosystem where collaboration and research-oriented features are often secondary considerations.

3 Platform Architecture and Design

PromptLab builds upon the foundational work of PromptSource (Bach et al., 2022), a pioneering work in template-based prompting research. However, PromptSource requires initial setup and local deployment. Furthermore, the collaboration setup via GitHub pull requests creates barriers for domain experts lacking a computing background. PromptLab addresses these limitations through a comprehensive web-based architecture that eliminates technical overhead while introducing seamless collaborative features. At a high level, PromptLab organizes work around *projects*, each associated with one or more HuggingFace datasets. Within a project, users operate under one of three

[2] https://platform.openai.com/playground/prompts
[3] https://console.anthropic.com/
[4] https://www.langchain.com
[5] https://github.com/run-llama/llama_index
[6] https://github.com/Mirascope/mirascope
[7] https://github.com/StacklokLabs/promptwright

Table 1: Comparison of prompt engineering tools across key dimensions relevant to collaborative research environments. Rows with "Limited Teams" collaboration are for the freemium tier.

Platform	Audience	Collaboration	HF Datasets	Pricing
PromptSource	Research	GitHub PRs	Native	Free
Agenta	Developers	Limited teams	No	Freemium
PromptHub	Developers	Teams	No	Freemium
LLMs Playground	Individuals	N/A	No	Pay-per-use
PromptLayer	Developers	Limited Teams	No	Freemium
ChainForge	Research	Link Sharing	No	Free
Langfuse	Developers	Limited Teams	No	Freemium
PromptLab	**R&D**	**Teams**	**Native**	**Free**

roles: *prompters*, *reviewers*, and *administrators*, and prompts progress through a structured lifecycle: draft creation, peer review, revision, and final approval. Figure 1 illustrates this end-to-end pipeline.

3.1 Core System Architecture

The platform migrated from Streamlit (as in promptsource) to employ Django as a Python backend framework. Database was setup with PostgreSQL for persistence. Redis was utilized for caching HuggingFace communications, saving time during dataset-intensive calls. Celery enables background task processing for computationally intensive operations, including dataset synchronization, and batch HuggingFace dataset refresh operations. The system exposes RESTful APIs with project-specific authentication tokens, enabling programmatic access from any programming environment. Docker-based deployment and CI/CD pipelines support both public research collaborations and private institutional use cases. PromptLab, instead of prompt review processes via GitHub pull requests, implemented role-based project management with access controls for prompters, reviewers, and administrators. Additionally, administrators can optionally set a minimum prompting workload where the datasets will be randomly distributed among the prompters.

3.2 Enhanced Templating and User Experience

PromptLab follows PromptSource practices when designing prompts, maintaining metadata fields like the name, and answer in classification tasks. For the prompt template, PromptLab extends Jinja2[8] templating with an optional alternative intelligent field insertion through double backslash triggers that display a dropdown list of available dataset fields. Selected fields render as visual tag components, improving the user experience while maintaining the Jinja2 $\{\{dataset\_key\}\}$ syntax under the hood. PromptLab adds a tag field that enables prompt organization and filtering by specific properties like their style (Chain-of-Thought, role-playing, formal), and task type (reasoning, classification).

The integrated template testing view provides real-time visual feedback through color-coded validation indicators. Figure 4 demonstrates this validation across different tasks, showing how the platform identifies well-formed templates versus those with syntax errors or logical inconsistencies, following the original prompting guidelines from PromptSource, and reducing review iteration cycles required for prompt refinement.

3.3 AI-Assisted Generation and LLMs Real-Time Evaluation

As primarily inspired by (Wang et al., 2022b; Taori et al., 2023), PromptLab allows prompters to generate AI-assisted prompts using ChatGPT as a strong pretrained language model. Prompters can generate diverse prompt variations from seed templates and carefully check the generated prompts before submitting for review. Real-time evaluation with OpenRouter models enables immediate testing across multiple language models without leaving the platform interface.

[8] https://jinja.palletsprojects.com/en/stable/

3.4 Collaborative Workflow: Review Lifecycle and Communication

A central contribution of PromptLab is its structured multi-stage collaborative workflow, replacing ad hoc coordination systems. Once a prompter submits a draft, it enters a review queue where reviewers can *approve* it, *return it for modification* with inline comments, or *reject* it. Each revision is tracked as a new version, preserving the full development history of a prompt. Administrators retain override capabilities at any stage and can monitor workload distribution across prompters. Figure 5 illustrates the complete state transitions from prompt creation to final approval, ensuring reviewer approvals for considered prompts.

RESTful APIs enable programmatic access with project-specific authentication, supporting diverse programming environments beyond Python-specific bindings. Docker-based deployment options and CI/CD pipelines accommodate both public research collaborations and private institutional projects.

4 Platform Evaluation

We conducted two complementary evaluations to evaluate PromptLab for practical considerations: prompt quality assessment comparing PromptLab-generated and manually-created prompts, and human usability analysis.

4.1 Prompt Quality Evaluation

To evaluate whether PromptLab facilitates higher-quality prompt development, we conducted a controlled comparison study where we compared prompts created by the platform with those created without using the tool.

4.1.1 Experiment Setup

We selected five benchmark datasets spanning classification (IMDB (Maas et al., 2011)), ToxicChat (Lin et al., 2023), MMLU (Hendrycks et al., 2020)) and generation tasks (IWSLT2017 (Cettolo et al., 2017), XSUM (Narayan et al., 2018)). Six researchers[9] with a strong background in prompt engineering were divided into two groups: the **Manual Group** developed prompts without PromptLab assistance, while the **PromptLab Group** used PromptLab. Each researcher created three prompts per dataset, yielding a total of 90 prompts (45 per group) employing diverse strategies including direct instruction, chain-of-thought (Wei et al., 2022b), and role-playing.

We evaluated all prompts across 13 LLMs spanning small (<=1B), medium (1–10B), and large (>10B) parameter scales. Each prompt was tested on a stratified subset of 1,000 samples from the test split of each dataset, where samples were randomly shuffled with a random seed for reproducibility. To account for evaluation variance, we conducted 10 independent runs per model using deterministic sampling (temperature=0.0) and averaged the results. Models were evaluated locally using vLLM (Kwon et al., 2023). For classification tasks, we explicitly instructed models to generate only the class label without additional text; violations were penalized. Similarly, for generation tasks, models were instructed to output only translations or summaries. Metrics followed task conventions: standard classification metrics (accuracy, precision, recall, and macro F1-score, F1-score is reported in the figures) for classification tasks, BLEU (Papineni et al., 2002) for translation, and ROUGE-L (Lin, 2004) for summarization. The evaluation code is available in GitHub[10].

4.1.2 Results and Analysis

Results are reported with normalized prompt performance. We normalized scores by setting the best-performing prompt to 100, with all other prompts scaled proportionally: $(x/y) \times 100$, where x is the prompt's score and y is the best score.

Figure 2 presents normalized prompt performance across small models (<=1B parameters). Medium and large parameter figures are provided in the appendices (Figures 7 and 8). PromptLab prompts (green), in the small LLMs, consistently achieve higher performance compared to manual prompts (red) across all five datasets and models. Manual prompts also exhibit higher variance. At medium and large scales, this difference diminishes as models become more capable and less sensitive to prompt formulation.

Figure 3 aggregates results across all model size categories, showing per-model mean performance and variance for each dataset. PromptLab prompts demonstrate a performance advantage across small and medium models. The consistency of this advantage across model scales suggests that PromptLab's features, particularly real-time validation and

[9] 3 Masters, 1 PhD student, 1 Post-Doc, and 1 Professor, all in computer science-related domains

[10] https://github.com/KFUPM-JRCAI/promptlab-evaluation

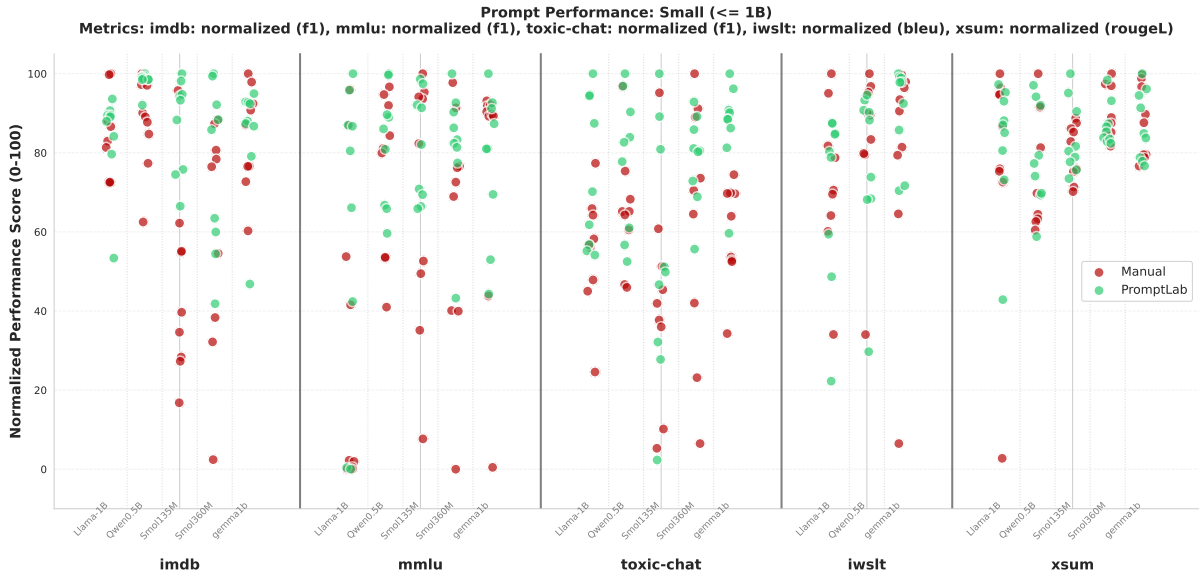

Figure 2: Prompt performance for small models (<1B parameters) across five datasets. Each point represents a single prompt's normalized performance. PromptLab prompts (green) consistently outperform manually-created prompts (red).

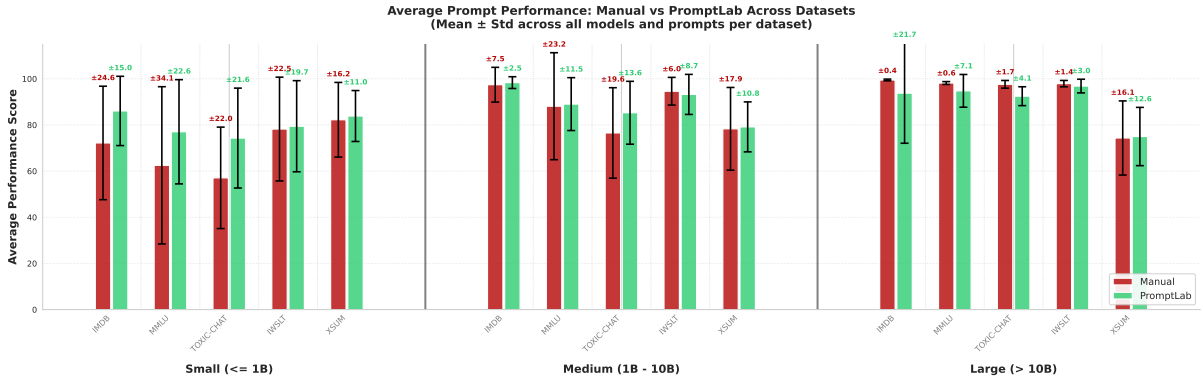

Figure 3: Per-model mean performance comparing manual versus PromptLab prompts across model size categories. Each bar represents one model's (averaged) performance across all the prompts in that group.

template feedback, enable researchers to identify and iterate on more effective prompting.

4.2 Usability Evaluation

To validate PromptLab's usability effectiveness, we conducted a user evaluation study[11] with 14 participants representing diverse backgrounds: 4 undergraduates, 8 graduate students/engineers, and 2 senior researchers. Participants ranged from beginners to advanced practitioners in NLP research, with varying prompt engineering experience levels. The evaluation assessed platform usability across key dimensions using 5-point Likert scales. Results demonstrate strong user satisfaction with mean scores of 4.3/5 for interface intuitiveness, 4.4/5 for AI-assisted validation, 4.5/5 for template feedback clarity, 4.2/5 for navigation ease, and 4.6/5 for collaborative workflow support. Participants rated PromptLab favorably compared to existing methods (4.4/5), with high likelihood to use (4/5) and recommend (4.2/5) the platform. Appendix G provides more in-depth details and insights on the evaluation results.

5 Case Study: Arabic NLP Instruction Tuning Research

To demonstrate PromptLab's effectiveness in real-world research scenarios, we present a case study from an Arabic instruction tuning project that represents one of the most substantial collaborative prompt engineering efforts for Arabic NLP to date.

[11] We used the following evaluation form https://forms.gle/nG9jXHRStUkgvLqE9

The project involved seven researchers and three reviewers who developed over 350 manually crafted prompt templates spanning 20+ Arabic NLP tasks across 80+ datasets (refer to Appendix C for the full datasets listing), totaling more than 6.3 million samples. Figure 6 in Appendix D presents the prompts distributions over the datasets. Tasks covered a broad spectrum including dialect identification, sentiment analysis, sarcasm detection, natural language inference, machine translation, and summarization, among others. Researchers employed both manual and AI-assisted prompt creation workflows. The platform's assignment and review interfaces allowed members to track individual progress and maintain a full revision history for each prompt submission. As a side result of this research, a framework for scoring prompts across similarity, performance, efficiency, and consistency dimensions was developed and presented in (Refai et al., 2025).

6 Conclusion and Future Work

PromptLab addresses the gaps in research-based collaborative prompt engineering through a comprehensive platform that supports the complete prompt life cycle from creation to validation and publication. The platform's design balances accessibility with technical sophistication, combining intuitive visual interfaces with powerful programmatic capabilities to enable broader participation while preserving expressiveness. The Arabic instruction tuning case study demonstrates the platform's effectiveness, with over 350 prompts across 80+ datasets successfully developed and validated through systematic collaborative workflows involving multiple researchers and reviewers.

The integration of AI-assisted prompts development with human oversight establishes a productive collaborative paradigm that enhances researcher productivity without compromising quality control. Future work will focus on integrating automated prompt optimization techniques, model evaluation, model fine-tuning, and developing comprehensive analytics for collaborative pattern analysis. The platform's open-source availability and deployment flexibility position it as a foundation for continued progress in prompt engineering, contributing to the democratization of advanced NLP research across diverse individual and institutional environments.

Acknowledgments

We gratefully acknowledge the support of the SDAIA-KFUPM Joint Research Center for Artificial Intelligence (JRCAI) for providing the resources and infrastructure that made this work possible through grant number JRC-AI-UCG-07. We extend our thanks to the 14 participants who contributed their time and expertise to the platform usability evaluation study, as well as the six researchers who participated in the controlled prompt quality comparison experiment. We also thank the team members involved in the Arabic instruction tuning case study for their dedicated efforts in developing and validating over 350 prompt templates. Finally, we appreciate the open-source community and the developers of the tools and libraries upon which PromptLab is built.

References

Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, and 1 others. 2022. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.

Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A Clifton. 2024. A brief review of hypernetworks in deep learning. *Artificial Intelligence Review*, 57(9):250.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. *arXiv preprint arXiv:1609.09106*.

Yun He, Huaixiu Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, Yaguang Li, Zhaoji Chen, Donald Metzler, and 1 others. 2022. Hyperprompt: Prompt-based task-conditioning of transformers. *arXiv preprint arXiv:2203.00759*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, and 1 others. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. 2023. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation. *Preprint*, arXiv:2310.17389.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

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

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, and 1 others. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Dania Refai, Maged S Al-Shaibani, and Irfan Ahmad. 2025. Is this the best prompt? scoring prompts for arabic nlp across llms. *IEEE Access*.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024a. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024b. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, and 1 others. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Sander Schulhoff, Michael Ilie, Neel Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Fu, Barnabás Póczos, and 1 others. 2024a. The prompt report: A systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, and 1 others. 2024b. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Hendrik Strobelt, Albert Webson, Victor Sanh, Benjamin Hoover, Johanna Beyer, Hanspeter Pfister, and Alexander M Rush. 2022. Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1146–1156.

Samuel Sučik, Daniel Skala, Andrej Švec, Peter Hraška, and Marek Šuppa. 2023. Prompterator: Iterate efficiently towards more effective prompts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 471–478, Singapore. Association for Computational Linguistics.

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. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. 2023b. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022b. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, and 1 others. 2022c. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Zijie Wang, Aishwarya Chakravarthy, David Munechika, and Duen Horng Chau. 2024. Wordflow: Social prompt engineering for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 42–50, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Hao Sun Zhang, Alihan Hüyük, and Mihaela van der Schaar. 2023. Query-dependent prompt evaluation and optimization with offline inverse rl. *arXiv preprint arXiv:2309.06553*.

Wenliang Zheng, Sarkar Snigdha Sarathi Das, Yusen Zhang, and Rui Zhang. 2025. GreaterPrompt: A unified, customizable, and high-performing open-source toolkit for prompt optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 405–415, Vienna, Austria. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

A PromptLab templating pipeline

Figure 4 and Figure 5 show the general templating pipeline utilized by PromptLab.

Between the following two sentences, which is more realistic? 1. {{first_sentence}} 2. {{second_sentence}} Select the correct option: {{answer_choices | join(' or ')}}. ||| {{answer_choices[label]}}	Valid
Among the following pairs of sentences, one is more likely or makes more sense: 1. {{first_sentence}} 2. {{second_sentence}} Evaluate and determine which one is more reasonable: {{label | join(' or ')}} ||| {{label[label]}} ✘	Invalid Does not render!
You have the following Arabic sentence: {{arabic}}. Based on this sentence, select the dialect from these options: {{answer_choices | join(', ')}}, and provide the appropriate dialect. ||| {{answer_choices[label]}}	Valid
Review the following text: '{{Text}}'. Determine the dialect from the following options:{{answer_choices | join(', ')}} ||| ✘	Invalid No output!

Figure 4: Examples for valid and invalid templates for common sense validation and dialect identification tasks.

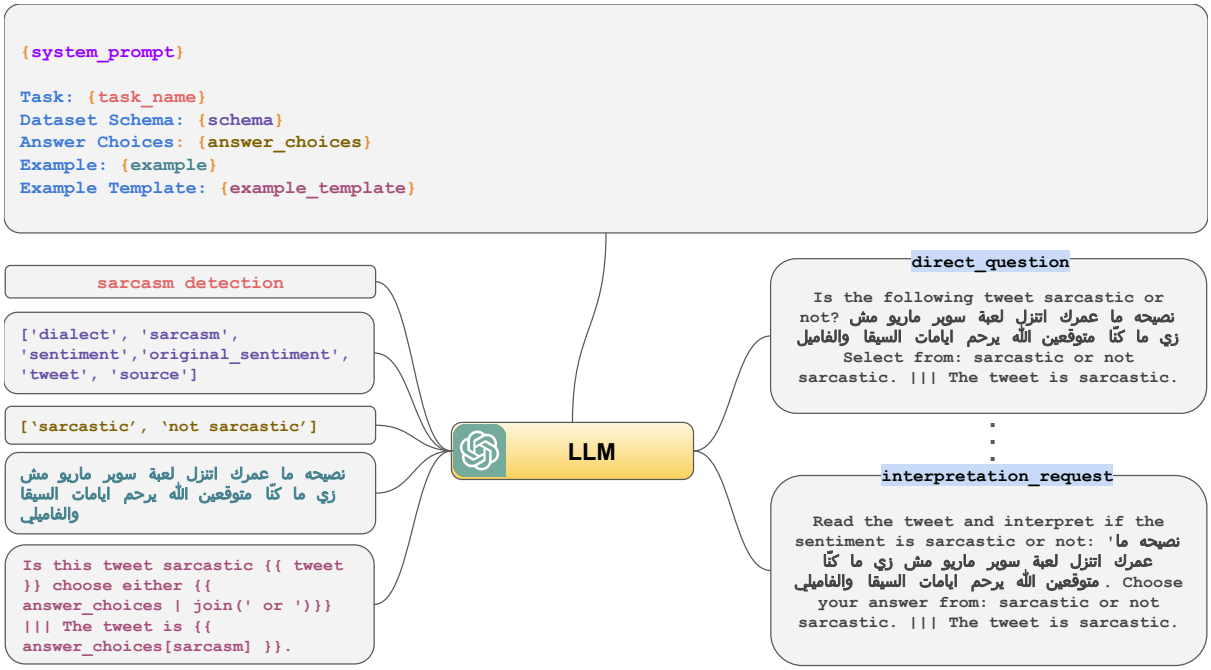

Figure 5: LLM template creation pipeline.

B AI generated prompts

We prompt the LLM to generate templates for a given dataset. The most important part is the system prompt. We use the following system prompt to create the templates for a given dataset. Texts shown in red are variables. For example, we can replace English with Arabic to create templates in Arabic instead of English.

System Prompt: You are a prompt template creator. Given a task, dataset schema, and answer_choices. You should create a template using Jinja that can be applied to an example in the dataset. The prompt and completion are separated by |||. You should create 5 different templates in English language as a json with a key that represents each template content. Please choose creative templates with enough variation. The order of the jinja variables can be changed. Do Not use a general name of the template like "template", USE more representative name. Do NOT print any other text except the json. Do NOT use any integer features. If there are answer_choices, use as is do NOT change. If there are answer_choices use the variable as is, do NOT introduce any new answer choices. All the jinja variables must be from the schema. Do NOT introduce new variable names. If the answer_choices in the example template exists use it in the completion without any changes. This is an important test. Please respect all the mentioned points.

C Prompts from Arabic NLP Datasets

Table 2 presents a listing of 80+ Arabic NLP datasets currently available on the Hugging Face platform that we utilized for our case study. This collection, totaling over 6.3 million samples, spans various tasks and linguistic phenomena. The datasets exhibit significant variation in size, from small specialized collections like PAAD (206 samples) to extensive corpora such as APCD (1,831,770 samples) and APCD2 (1,657,003 samples). As shown in Table 2, the datasets cover fundamental text processing tasks like diacritization (Arabic Text Diacritization, Shakkelha) as well as complex semantic tasks such as

commonsense validation and natural language inference (ArEntail). Many datasets focus on specific dialect variations or regional Arabic variants, such as the Shami dataset (66,251 samples) and the Arabic Dialects Dataset (9,992 samples). Others target particular applications like sentiment analysis (HARD, LABR) and text classification (SANAD, Ultimate Arabic News Dataset). While this collection represents a substantial resource for Arabic NLP research, the distribution of samples across different tasks and the varying quality of annotations present both opportunities and challenges for instruction-tuning approaches. Each dataset in Table 2 is accompanied by its Hugging Face URL, facilitating easy access and integration into research workflows. Each of these datasets has at least one prompt associated with it.

Table 2: A comprehensive list of Arabic NLP datasets curated for prompts.

Dataset	**Size**	**HuggingFace ID**
Commonsense validation	11,000	arbml/Commonsense_Validation
Arabic Text Diacritization	55,000	arbml/arabic_text_diacritization
Shakkelha	533,384	arbml/shakkelha
AraBench	42,113	arbml/AraBench_dev
Arabic Dialects Dataset	9,992	arbml/Arabic_Dialects_Dataset
araData	12,231	arbml/araData
Shami	66,251	arbml/Shami
Twt15DA_Lists	3,037	arbml/Twt15DA_Lists
Emotional-Tone	10,065	emotone-ar-cicling2017/emotone_ar
SemEval-2018 Task 1	4,381	SemEvalWorkshop/sem_eval_2018_task_1
ArMATH	6,000	arbml/ArMATH
APCD	1,831,770	arbml/APCD_remap
APCD2	1,657,003	arbml/APCDv2
ashaar	212,499	arbml/Ashaar_dataset
ArabicMMLU	14,575	arbml/ArabicMMLU
cidar-mcq	100	arbml/CIDAR-MCQ-100
belebele	900	facebook/belebele
QA4MRE	160	arbml/qa4mre
EXAMS	562	OALL/Arabic_EXAMS
AQMAR	4,188	arbml/AQMAR_batched
CANERCorpus	16,139	arbml/caner_batched
Cross-lingual NER	1,208	arbml/Zero_Shot_Cross_Lingual_NER_ar_batched
Disease NER	3,906	arbml/Disease_NER_batched
Named Entities Lexicon	48,753	arbml/Named_Entities_Lexicon
ArEntail	6,000	arbml/ArEntail
Textual Entailment	422	arbml/ArabicTE
Arabic OSACT4	7,837	arbml/OSACT4_hatespeech
MLMA hate speech	3,353	arbml/MLMA_hate_speech_ar
MPOLD	4,000	arbml/MPOLD
OffensEval 2020	9,666	arbml/offenseval_2020
Dangerous Speech Dataset	5,009	arbml/Dangerous_Dataset
Arabic Hate Speech 2022	9,823	arbml/Arabic_Hate_Speech

Continued on next page

Table 2 – continued from previous page

Dataset	Size	HuggingFace ID
Arabic POS Dialect	1,400	arbml/QCRI_arabic_pos_dialect
Arabic senti-lexicon	3,941	arbml/Senti_Lexicon
AQAD	17,911	arbml/AQAD
Arabic RC datasets	1,008	arbml/Arabic_RC_AQA
ARCD	1,395	hsseinmz/arcd
tydiqa-goldp	15,726	khalidalt/tydiqa-goldp
MKQA	10,000	apple/mkqa
xquad	1,190	google/xquad
TYDIQA	15,726	asas-ai/tydiqa-ar
ACVA	9,000	arbml/ACVA
ArSarcasm	10,547	iabufarha/ar_sarcasm
Sa`7r	19,804	arbml/SaudiIrony
ArSarcasm-v2	15,548	arbml/ArSarcasm_v2
iSarcasmEval	1,400	arbml/iSarcasmEval_task_A
nsurl	3,715	arbml/nsurl_2019_task8_test
Quran Hadith Datasets	8,144	arbml/Quran_Hadith
HARD	105,698	Elnagara/hard
LABR	14,695	mohamedadaly/labr
OCLAR	3,916	arbml/oclar
BRAD 1.0	510,598	arbml/BRAD
AJGT	1,800	komari6/ajgt_twitter_ar
ArSAS	19,897	arbml/ArSAS
ArSentiment	8,364	hadyelsahar/ar_res_reviews
ASTAD	68,070	arbml/Sentiment_Analysis_Tweets
ASTD	9,694	arbml/ASTD
BBN Blog Posts	1,200	arbml/BBN_Blog_Posts
ElecMorocco2016	10,254	arbml/ElecMorocco
ATT	2,154	arbml/ATT
MSAC	1,829	arbml/MSAC
NileULex	5,953	arbml/NileULex
Sudanese Dialect tweets	2,119	arbml/Sudanese_Dialect_Tweet
Sudanese Telecom tweets	5,346	arbml/Sudanese_Dialect_Tweet_Tele
Syria Tweets	2,000	arbml/Syria_Tweet_Sentiment
TSAC	11,871	arbml/TSAC
AT-ODTSA	3,000	arbml/AT_ODSTA
AraStance	4,063	arbml/arastance
Mawqif	3,502	arbml/Mawqif
ANS CORPUS	3,786	arbml/ANS_stance
ArCovidVac	9,988	arbml/ArCovidVac
AraSum	49,603	arbml/AraSum

Continued on next page

Table 2 – continued from previous page

Dataset	Size	HuggingFace ID
WikiLingua	9,995	esdurmus/wiki_lingua
XLSum	46,897	GEM/xlsum
AGS-Corpus	141,467	FahdSeddik/AGS-Corpus
Goud-sum	158,282	Goud/Goud-sum
ARGEN	1,200	arbml/ARGEN_title_generation
ANTCORPUS	10,161	arbml/antcorpus
Khaleej-2004	5,690	arbml/khaleej_2004
OSAC	5,070	arbml/OSAC_CNN
PAAD	206	arbml/PAAD
SANAD	141,807	arbml/SANAD
Ultimate Arabic News	196,279	arbml/ultimate_arabic_news
Watan-2004	20,291	arbml/watan_2004
Total	6,324,527	

D Arabic instructions tuning dataset distribution

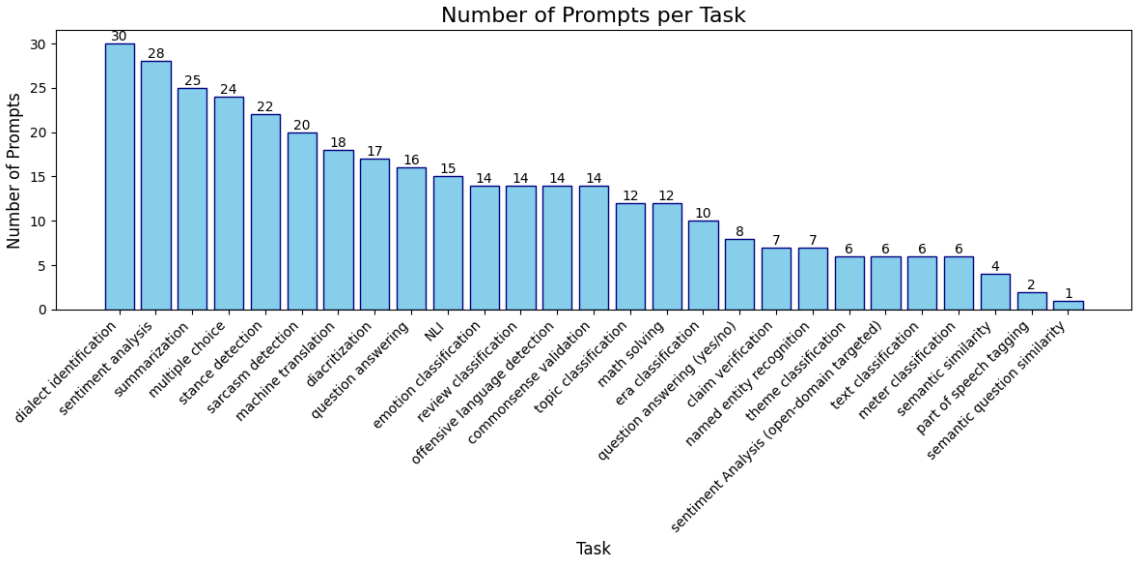

Figure 6: Prompts distribution across different Arabic NLP tasks.

```python
import requests
import json

# The URL for the API endpoint
url = "https://promptlab.up.railway.app/api/prompt/create"

# The headers for the request
headers = {
    "Content-Type": "application/json"
}

# The data payload
data = {
    "name": "<prompt-name>",
    "template": "Translate {text} to {language}",
    "dataset_huggingface_name": "<dataset-name>",
    "dataset_subset": "",  # optional
    "project_secret_key": "<prompting-project-secret-key>",
    "created_by": "<created-by>",
    "tags": ["AI generated", "AI translated"],  # optional
    "text_direction": "ltr",  # optional, defaults to ltr
    "answer_choices": json.dumps([{"value": "choice1"}])
}

# Send the POST request
response = requests.post(url, headers=headers, json=data)

# Check the response
if response.status_code == 201:
    print("Prompt created successfully!")
    print("Response:", response.json())
else:
    print("Failed to create prompt")
    print("Status code:", response.status_code)
    print("Response:", response.text)
```

Listing 1: Python code for creating prompts via PromptLab API

E REST API communication

To facilitate programmatic access, we implemented a RESTful API to easily integrate with any existing pipelines. We mainly implemented two primary endpoints:

- **Prompt Creation Endpoint:** Enables programmatic creation of prompts. The endpoint implements proper validation and authentication mechanisms to ensure data integrity.
- **Prompt Retrieval Endpoint:** Provides filtered access to the prompt repository related to a prompting project.

These endpoints are useful as they:

- Automate prompt generation.
- Automate prompt sharing and merging on different datasets when appropriate.
- Integrate the platform into existing research pipelines.
- Help in implementing prompt evaluation workflows.

Listings 2,1 provide Python code snippets to interact with these endpoints.

To retrieve existing prompts, the prompt listing endpoint can be utilized to list prompts related to a project as in Figure 2:

```python
import requests

url = 'https://promptlab.up.railway.app/api/prompt/list'
api_response = requests.get(
    url=f'{url}?project_secret_key=<project-secret-key>',
)

if api_response.ok:
    prompts = api_response.json()
else:
    print(api_response.text)
```

Listing 2: Python code for retrieving prompts via PromptLab API

F Platform Prompt Quality Evaluation

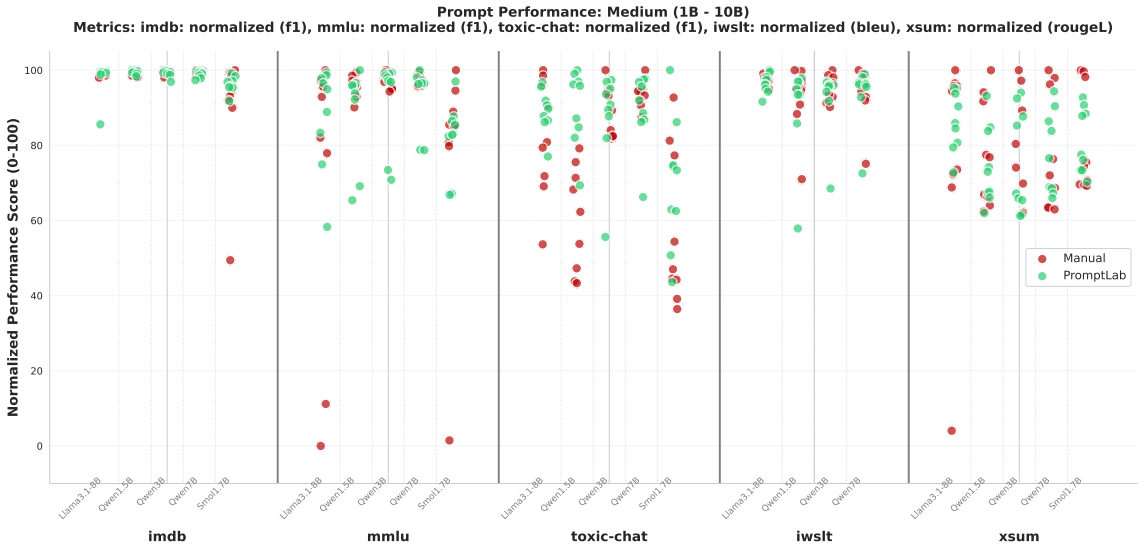

Figure 7: Prompt performance for medium models (10B<1B parameters) across five datasets. Each point represents a single prompt's normalized performance. PromptLab prompts (green) consistently outperform manually-created prompts (red).

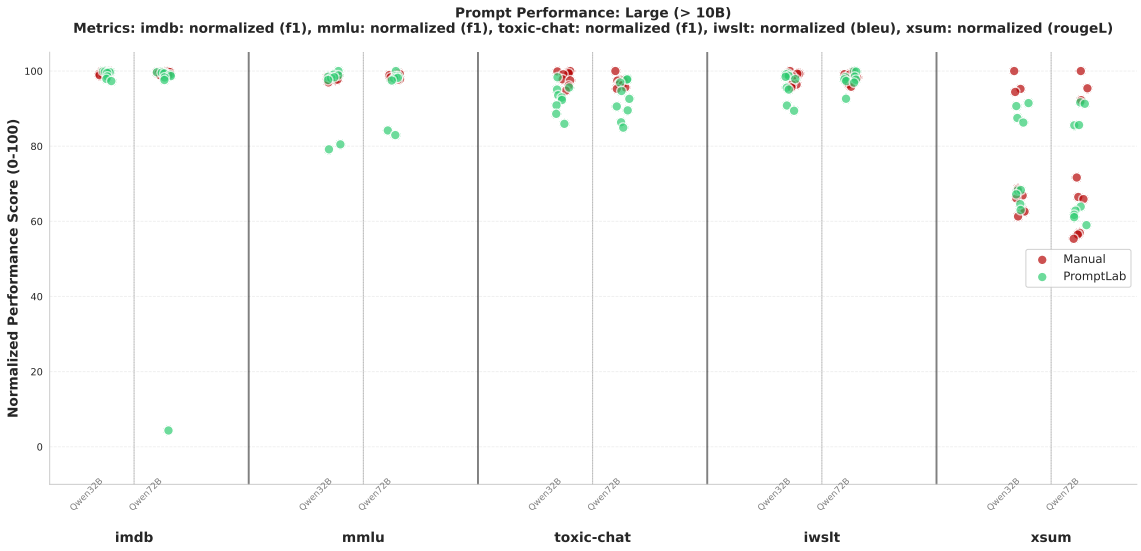

Figure 8: Prompt performance for large models (>10B parameters) across five datasets. Each point represents a single prompt's normalized performance. PromptLab prompts (green) consistently outperform manually-created prompts (red).

G Platform Usability Evaluation Results

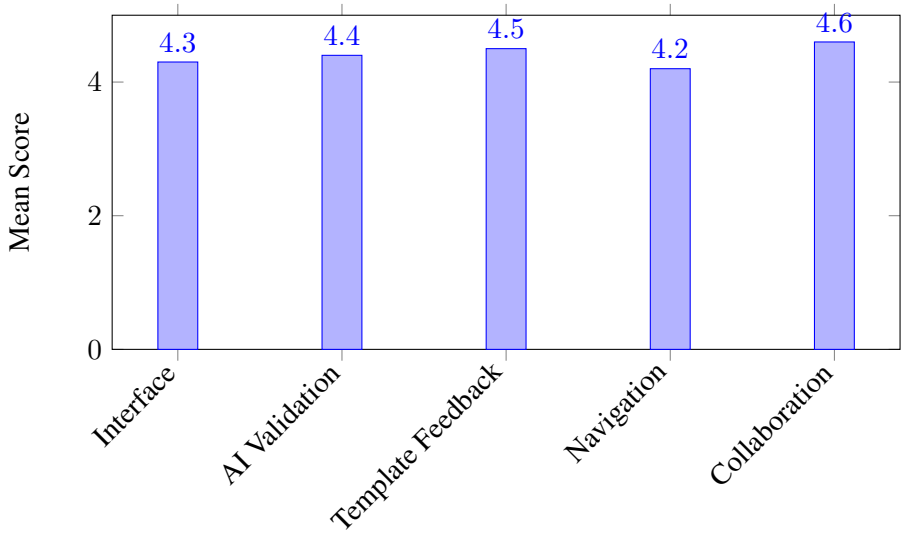

Figure 9: Platform usability scores across key dimensions (5-point Likert scale).

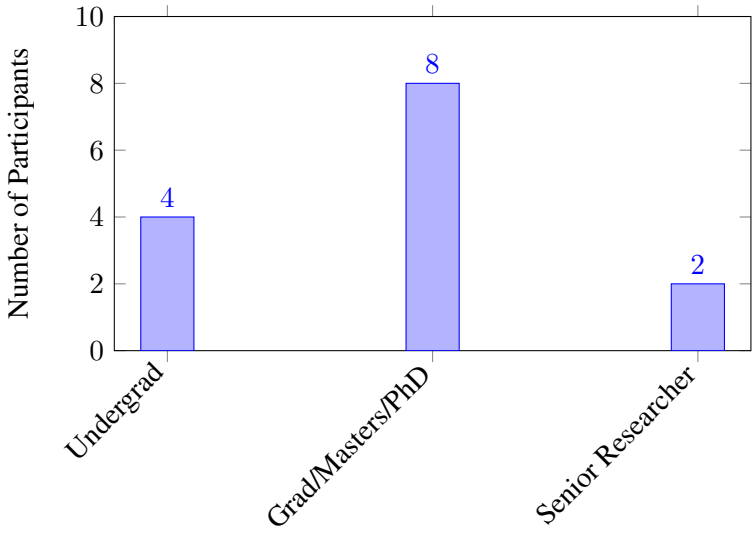

Figure 10: Participant distribution by education level (N=14).

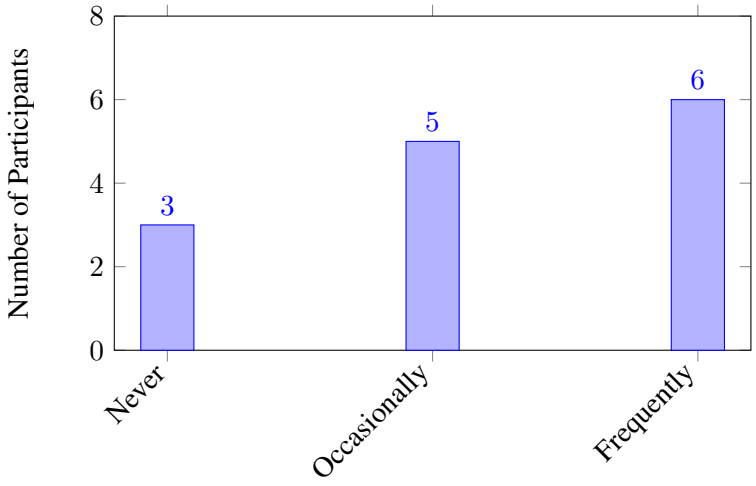

Figure 11: Prior prompt engineering experience distribution (N=14).

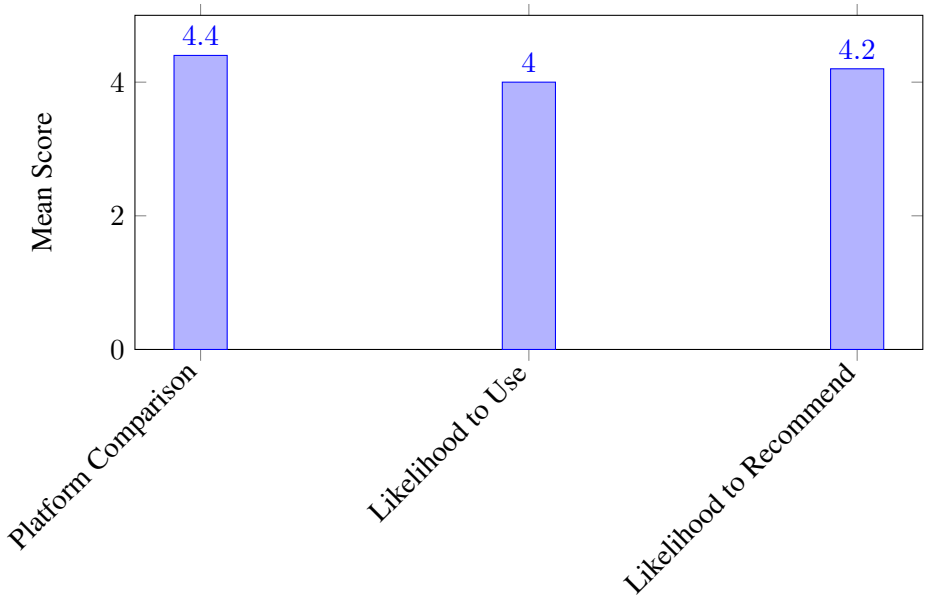

Figure 12: Overall platform satisfaction metrics (5-point Likert scale).

H PromptLab: Tool Documentation

PromptLab is designed to streamline and enhance the process of creating, refining, and managing prompts for Large Language Models (LLMs). Unlike existing prompt engineering platforms, PromptLab integrates a comprehensive set of functionalities—ranging from dataset exploration and prompt editing to review workflows and AI-assisted prompt generation. By offering a centralized environment where researchers, linguists, and developers can collaboratively develop and maintain prompts, PromptLab aims to foster an environment for preparing high-quality datasets.

Furthermore, PromptLab is structured to support an iterative feedback loop, guiding users through the entire lifecycle of prompt creation—from initial brainstorming to final approval. By providing dedicated interfaces for dataset inspection, prompt variation, structured reviews, and automated transformations (such as translation or AI-generated expansions), the tool facilitates a more dynamic and data-driven approach to prompt engineering. The platform's detailed record-keeping of revision histories, reviewer decisions, and dataset metadata further ensures reproducibility and accountability in collaborative research settings. In doing so, PromptLab empowers the NLP community with a scalable, user-friendly resource that promotes best practices, accelerates model training preparation, and ultimately contributes to the

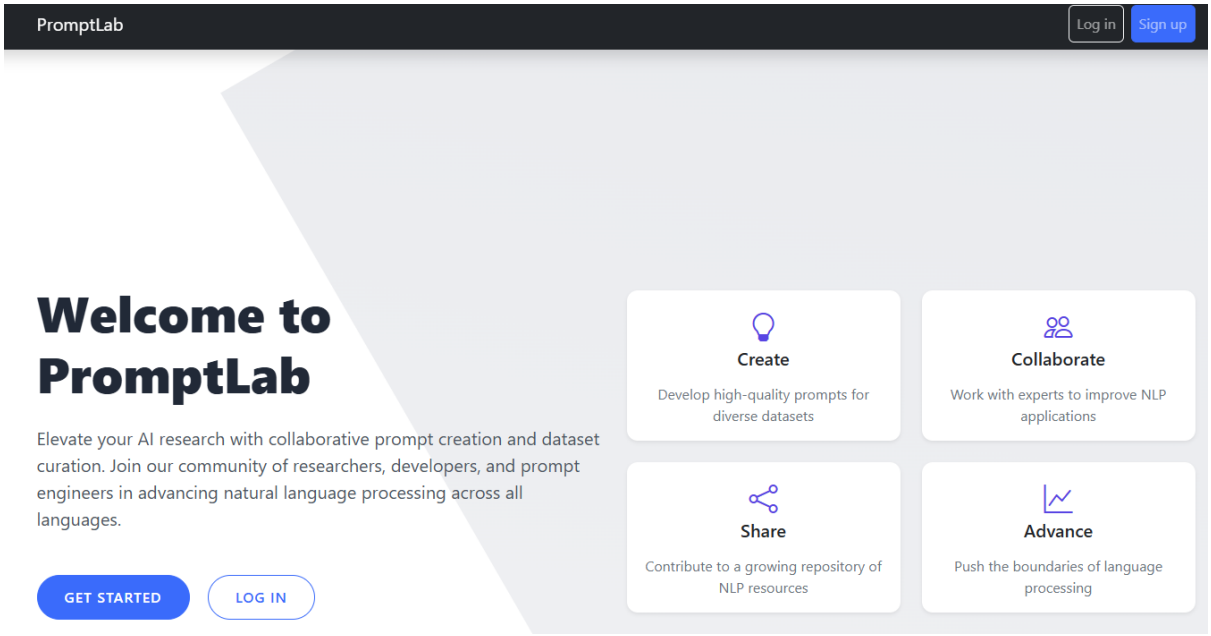

Figure 13: PromptLab: The main dashboard.

advancement of language processing technologies.

In the next section, we will present the main functionalities of the tool along with sample screens.

H.1 Main Features of the Tool

Upon logging into the platform, users are presented with a primary dashboard designed to facilitate streamlined navigation and highlight the tool's core functionalities. The dashboard is shown in Figure 13.

The top portion of the interface displays the platform's logo and a concise header menu offering essential actions, such as "Log In" or "Sign Up," depending on the user's authentication state. The central section of the screen prominently features a welcoming message ("Welcome to PromptLab") accompanied by a brief explanatory tagline emphasizing the platform's focus on elevating NLP research through collaborative prompt creation and dataset curation.

Below the heading, four distinct feature panels are arranged horizontally, each represented with an icon and descriptive label:

- Create: Encourages the development of high-quality prompts.
- Collaborate: Invites users to engage with experts and peers, fostering an environment that promotes shared learning and collective improvements in NLP.
- Share: Provides pathways for contributing to a communal repository of NLP resources, thereby expanding the overall dataset and knowledge base accessible to the community.
- Advance: Focuses on the progressive enhancement of natural language processing, guiding users toward advanced practices and cutting-edge methodologies.

PromptLab organizes work into prompting projects that group related datasets, prompts, and collaborators under a single umbrella. From the My Projects page, users can browse, search, and sort all accessible projects, each shown as a card summarizing its title, description, number of datasets, and team size, with quick actions to view the workspace, open tasks and datasets, or edit project details as illustrated in Figure 14. Opening a project leads to a project overview screen displaying high-level metadata (name, description, owner, creation details, and minimum-prompt requirements), project-level API information including a secret key and example usage code, and a Team Members panel listing collaborators and their roles with search and pagination controls as shown in Figure 15. Together, these views provide the

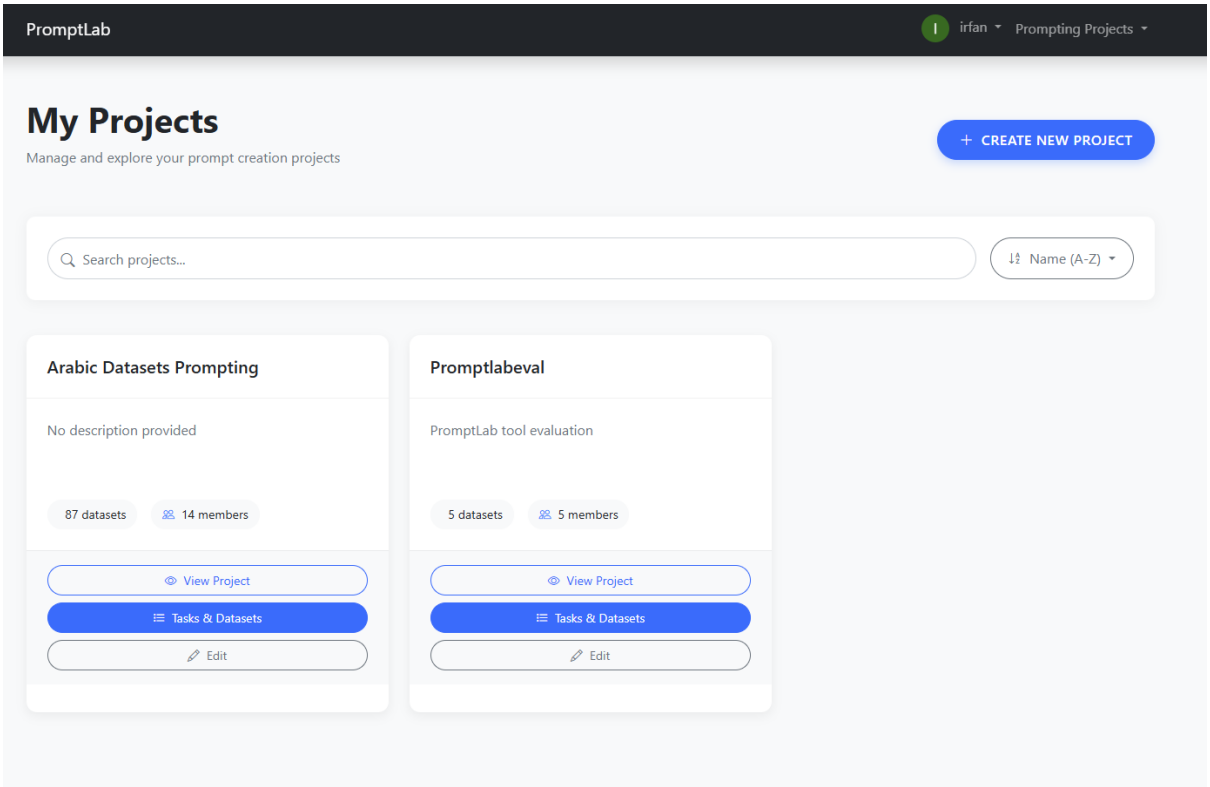

Figure 14: PromptLab: List of projects.

main project-management layer in PromptLab, from which users configure access, connect automated workflows, and then proceed to detailed task and dataset management.

A prominent "Get Started" call-to-action button is positioned beneath the introductory text, directing users to begin creating or exploring content immediately. The overall visual presentation employs a clean, modern design with intuitive iconography and a balanced color palette, enhancing ease of use and navigability. This layout ensures that both newcomers and experienced researchers can efficiently locate key functionalities and engage with the platform's core features.

Prominent call-to-action buttons direct new users to begin exploring available resources, while returning researchers can effortlessly resume their work. Transitioning from this home environment, the "NLP Tasks" interface (Figure 16) provides a structured overview of diverse language-related tasks, each aligned with relevant datasets and associated prompt counts. The clean tabular layout simplifies the discovery process by allowing users to quickly scan the catalog of tasks, gauge their complexity through prompts and dataset availability, and select actions such as "View Datasets" to delve deeper into a particular dataset.

Building upon the task-level insights, the "PromptLab Prompts List" interface, as shown in Figure 17, offers a detailed registry of prompts associated with chosen datasets for a specific user. Here, users can track prompt submission status (e.g., draft or submitted), review semantic tags that characterize the prompt's nature (e.g., "CoT" for chain-of-thought reasoning), and make informed decisions about which prompts require refinement and validation. This structured presentation supports a workflow where researchers iterate on prompt quality and scope.

The "My Assigned Datasets" interface, as in Figure 18, personalizes the experience by displaying only those datasets a user is currently assigned to work on. The interface quantifies progress through a simple count (e.g., "2/5" prompts created), helping users maintain momentum and accountability in their prompt development process. This tailored view closes the loop from broad task exploration to individual dataset responsibilities, encouraging sustained engagement and focused contributions to the platform's collaborative research ecosystem.

As researchers advance deeper into the platform's resources, they encounter specialized interfaces

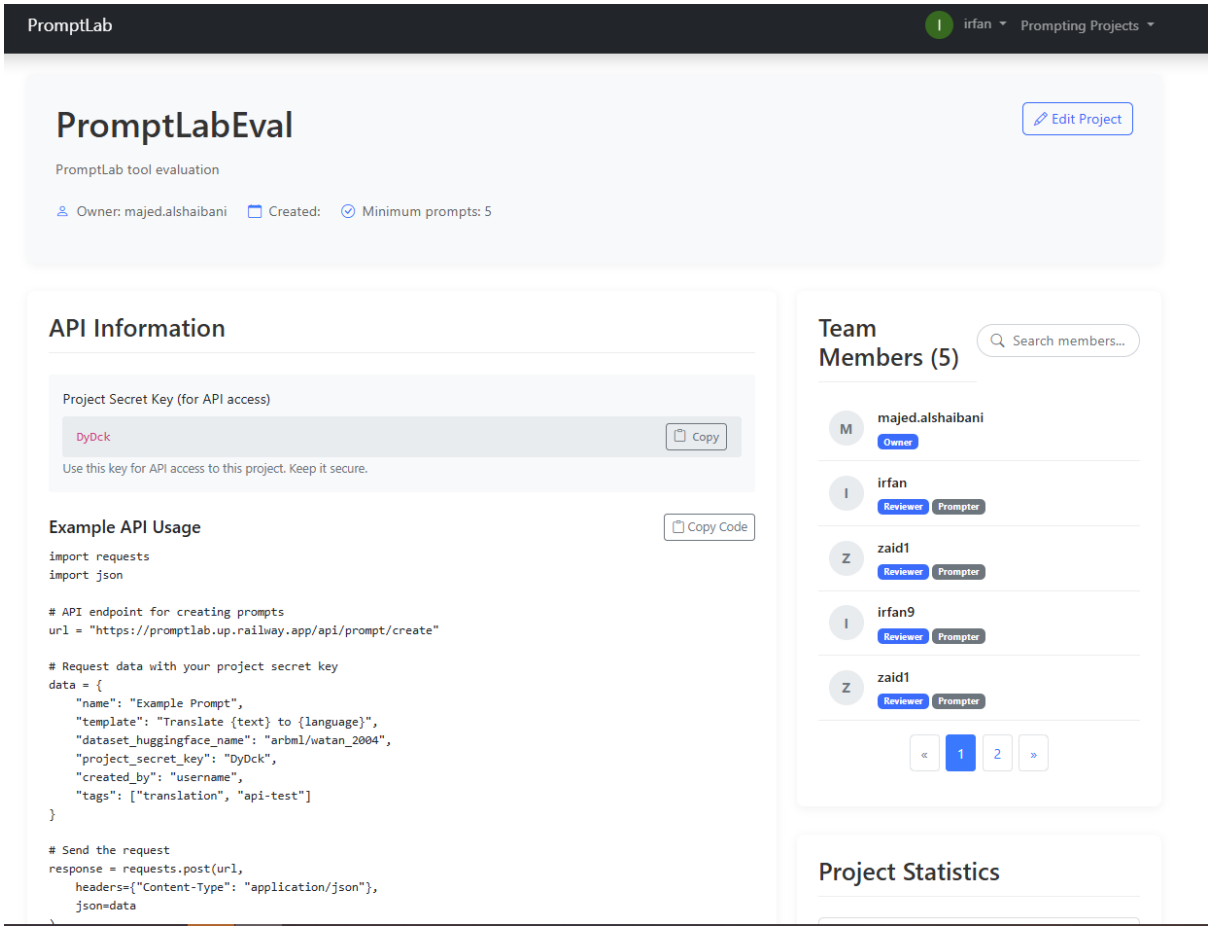

Figure 15: PromptLab: Project management dashboard.

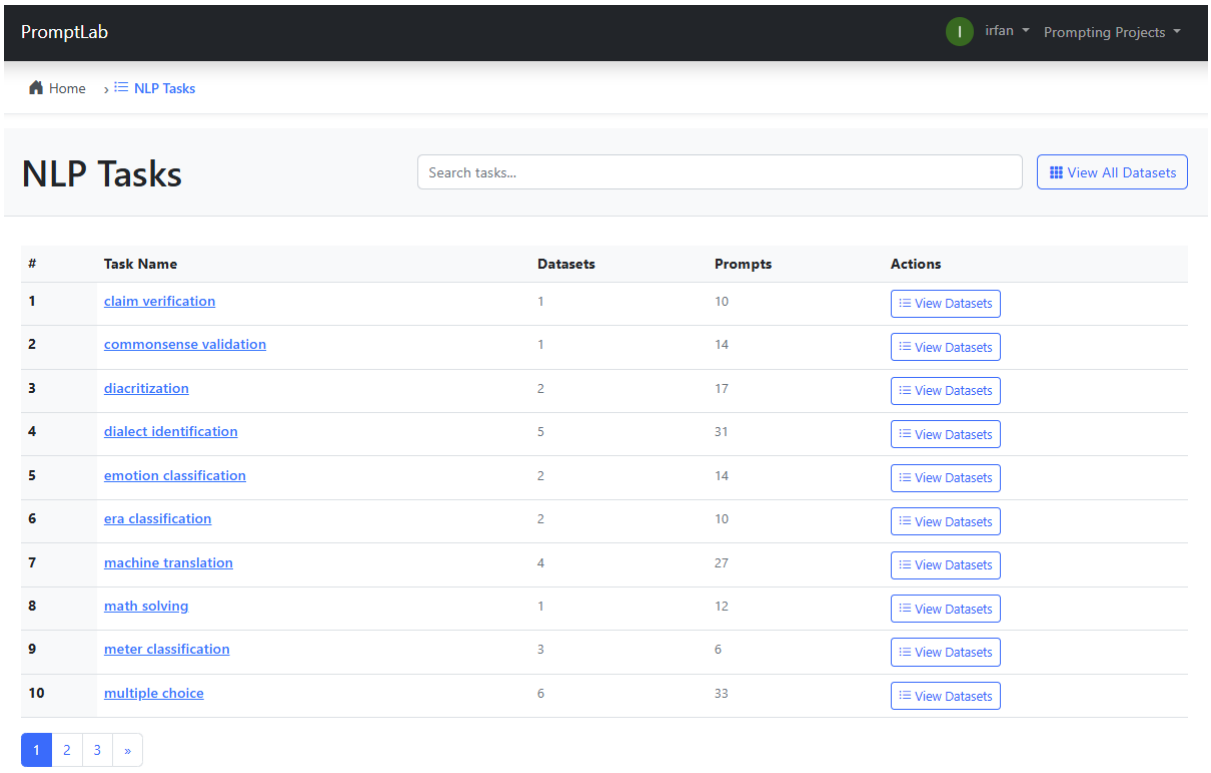

Figure 16: PromptLab: List NLP Tasks.

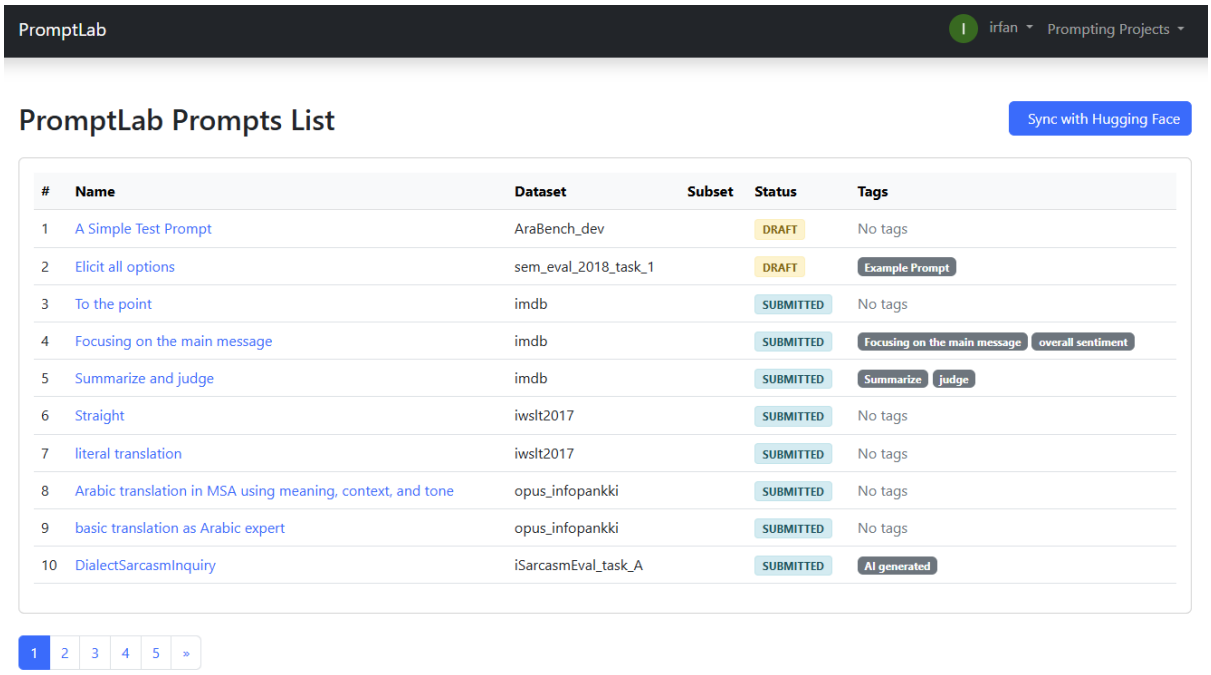

Figure 17: PromptLab: An interface to monitor prompts created by a user.

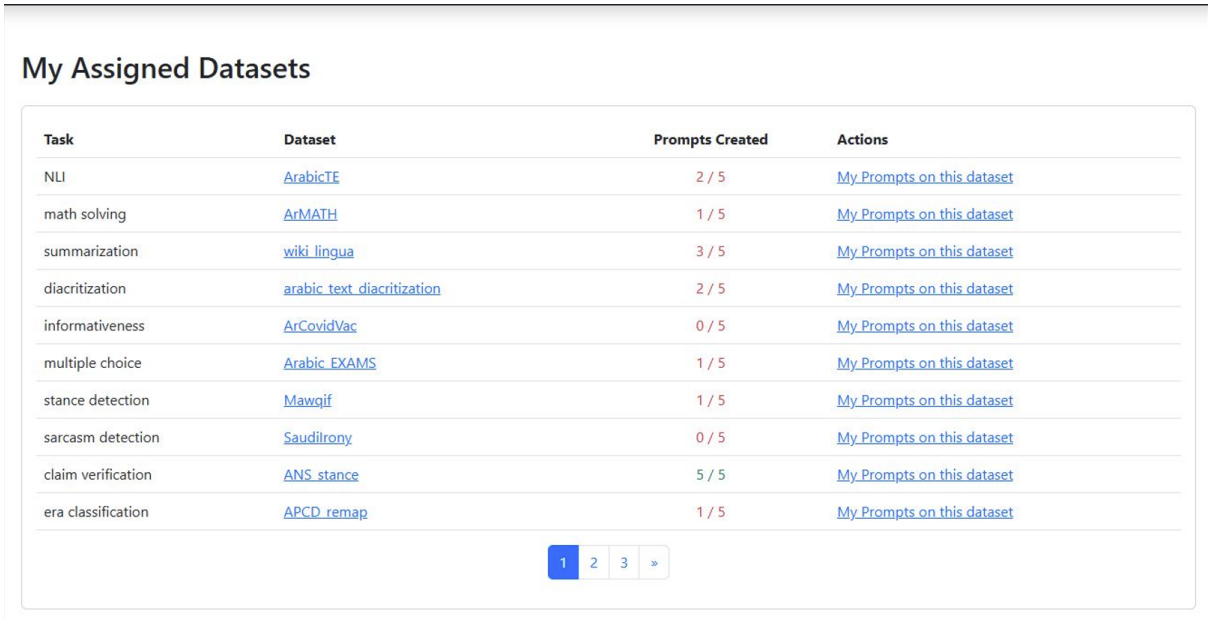

Figure 18: PromptLab: Explore the progress of a user for the datasets assigned to him.

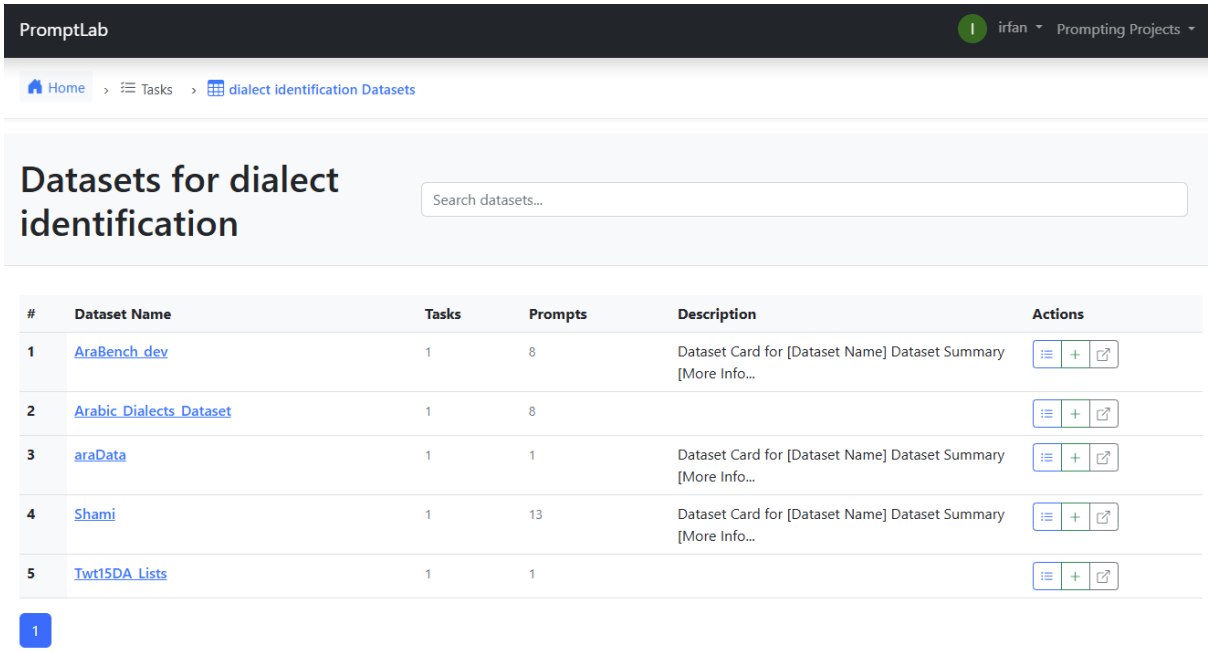

Figure 19: PromptLab: Explore the datasets for a specific task.

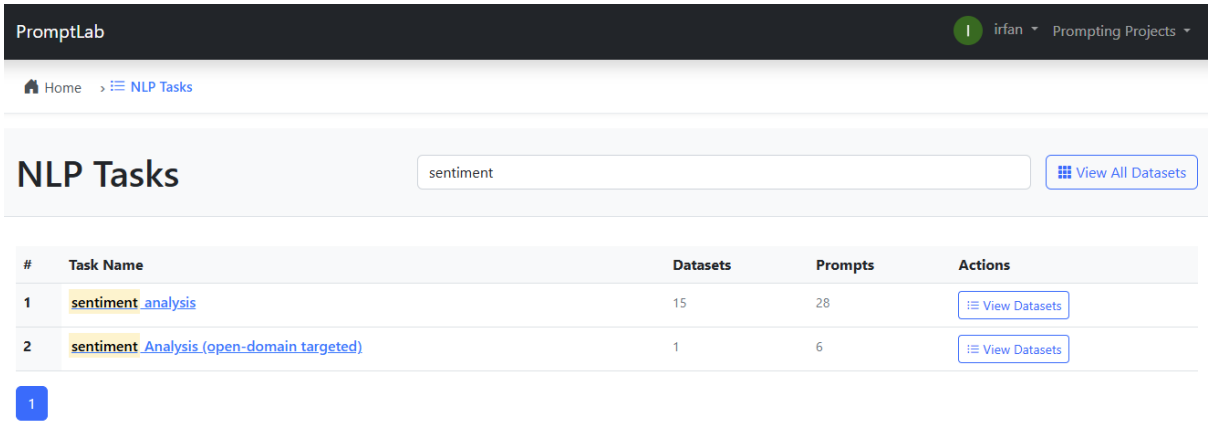

Figure 20: PromptLab: Explore tasks and datasets associated with a task.

that serve as navigation hubs for narrowing down tasks and datasets. One such interface focuses on "Datasets for a <Task, like Dialect Identification>" as shown in Figure 19, guiding the user from a specific task domain (e.g., dialect identification) to a curated selection of relevant datasets. Users can quickly filter through entries using a search bar, skim tabulated information about the number of tasks and prompts linked to each dataset, and view a brief textual description. The interface also provides actionable controls to directly access dataset details, add new prompts, or edit existing records, thereby ensuring that researchers can promptly engage with or contribute to the datasets they find most relevant or needs attention.

Similarly, the "NLP Tasks" interface can be refined through thematic filtering—an example being a direct keyword search (e.g., "sentiment"). This refined view, as shown in Figure 20, distills the platform's broad inventory of tasks into a more focused subset that matches the researcher's interests. Each returned task entry is accompanied by the count of datasets and prompts currently associated with it, as well as straightforward navigation buttons leading to those datasets. Such a search-driven interface encourages exploratory browsing and rapid iteration, allowing researchers to quickly locate tasks that align with their specific research questions or methodological preferences.

Expanding outward from this thematic filtering, the "All Datasets" interface (Figure 21) provides a

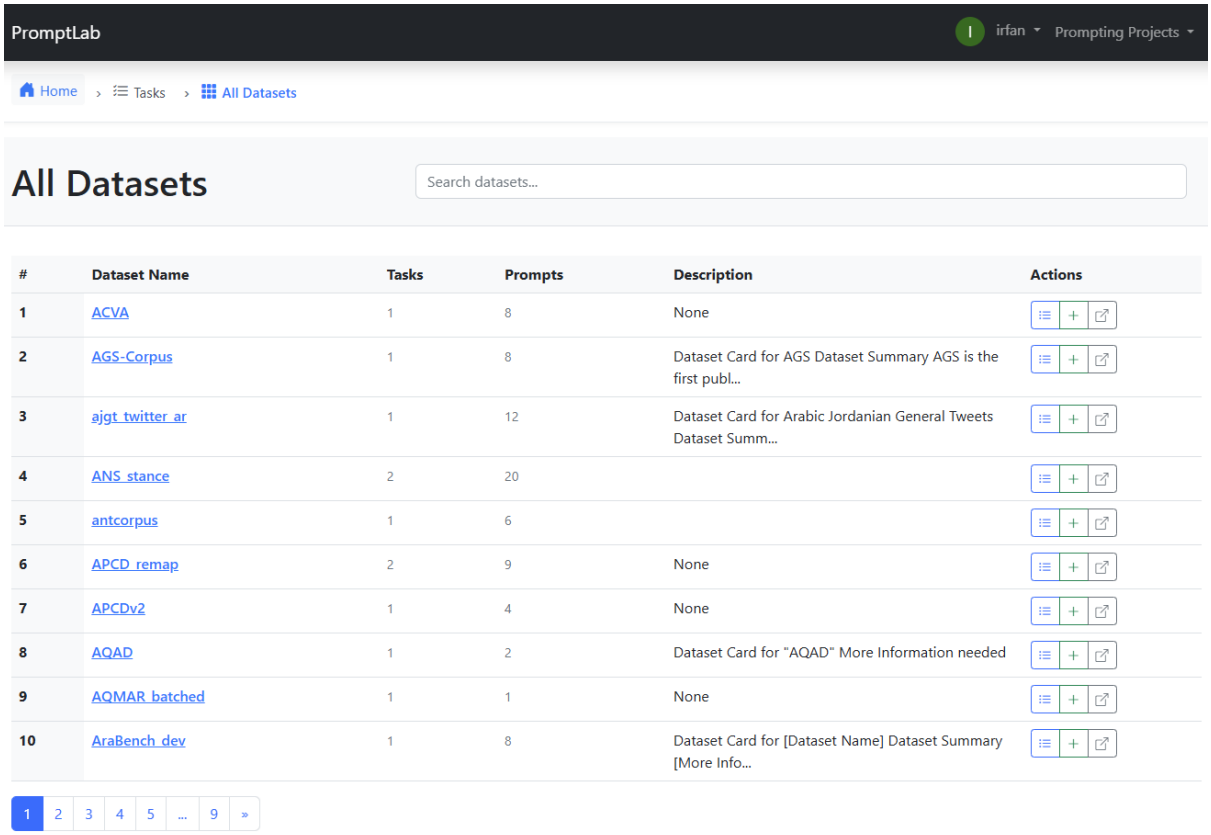

Figure 21: PromptLab: Explore datasets.

comprehensive catalog of the platform's entire dataset repository. Researchers are greeted with a sortable, paginated table presenting the dataset name, the number of associated tasks and prompts, and a concise description—a dataset card that can contain summaries, key points, or additional metadata. Integrated search functionality allows users to instantly narrow this broad collection down to a manageable subset tailored to their investigative needs. The interface's interactive controls—such as the ability to add prompts or view dataset details—foster an environment of continual enrichment and refinement of the datasets themselves.

By applying search terms on the "All Datasets" interface (e.g., filtering by the keyword "sentiment" as shown in Figure 22), users can locate specific resources that resonate with their target analysis domains. The returned listings not only confirm that the desired subject matter is present in the repository but also quantify its richness via the number of prompts and tasks available. This synergy of broad cataloging, fine-grained filtering, and direct action links creates a cohesive workflow: from scanning an extensive resource library to pinpointing niche datasets, and finally, to taking meaningful steps in prompt creation and data curation.

As a researcher's exploration narrows down to a single dataset, the platform provides specialized interfaces to support prompt-level oversight and refinement. Consider the "Prompts for Dataset" interface (Figure 23) : upon selecting a specific dataset—such as AraBench—users are presented with a structured listing of all associated prompts. This listing is organized into intuitive tabs (e.g., "My Prompts," "Submitted Prompts," and "All Prompts (Drafts Excluded)") that filter the view based on the user's involvement and the prompt's lifecycle stage. Each prompt entry includes pertinent information such as the task category and creation metadata. Status indicators (e.g., "APPROVED") and tag labels (e.g., "Example Prompt") help track a prompt's review stage and thematic relevance at a glance. A prominent "Create New Prompt" button encourages contribution and continuous dataset enrichment, allowing researchers to easily add their own prompts once they have reviewed existing entries.

As users interact with their assigned datasets and refine their own contributions, the platform supports

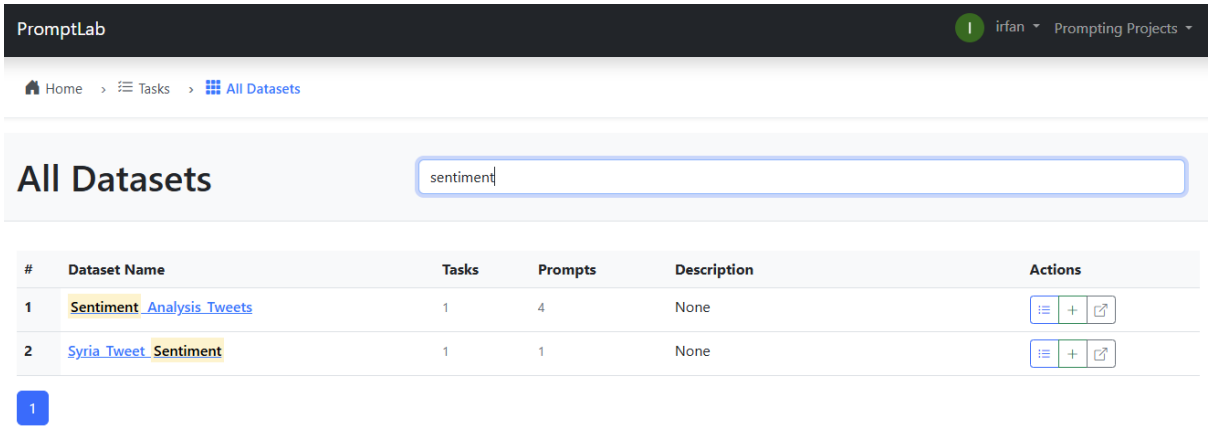

Figure 22: PromptLab: Search datasets.

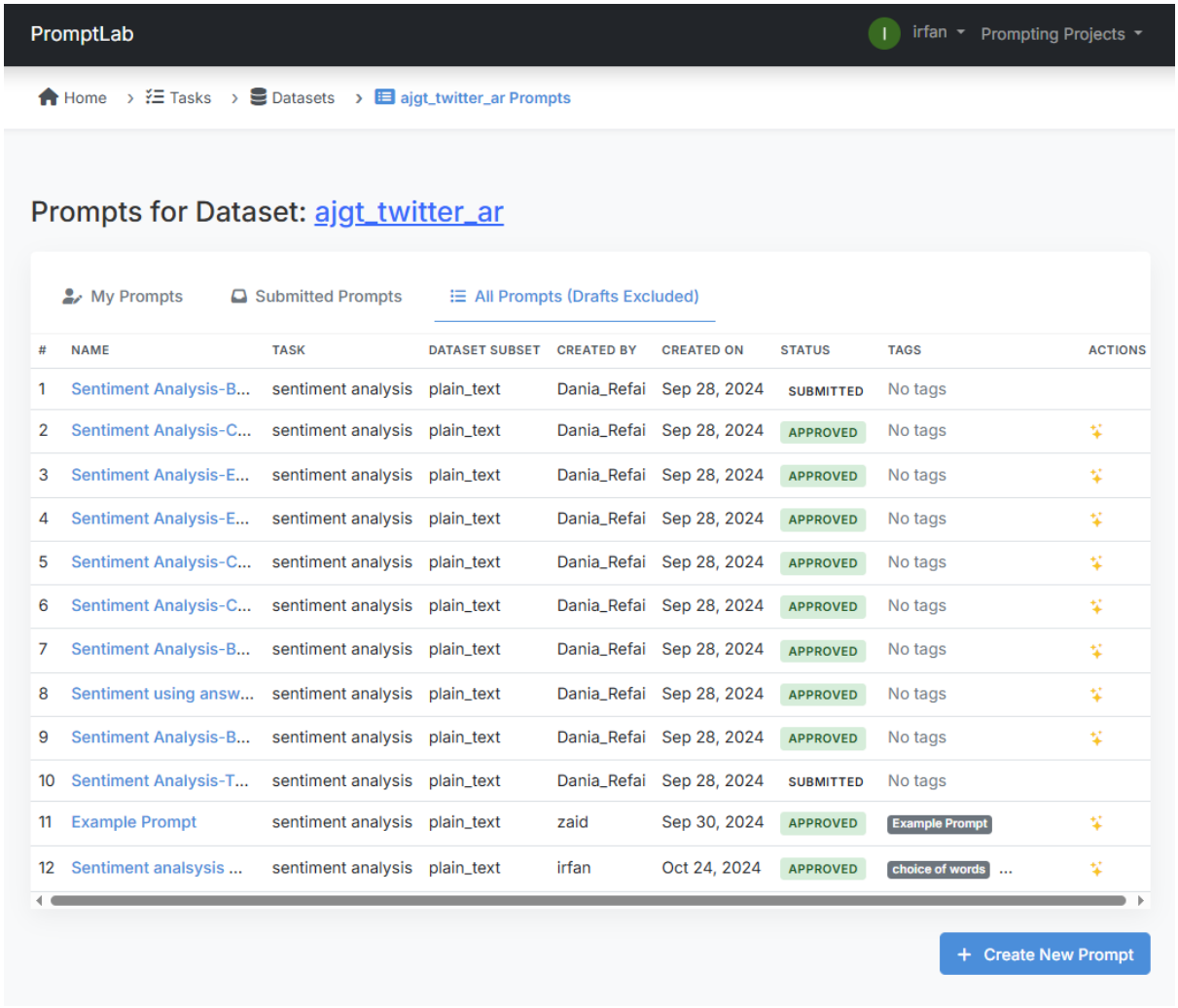

Figure 23: PromptLab: Prompt list for a specific dataset.

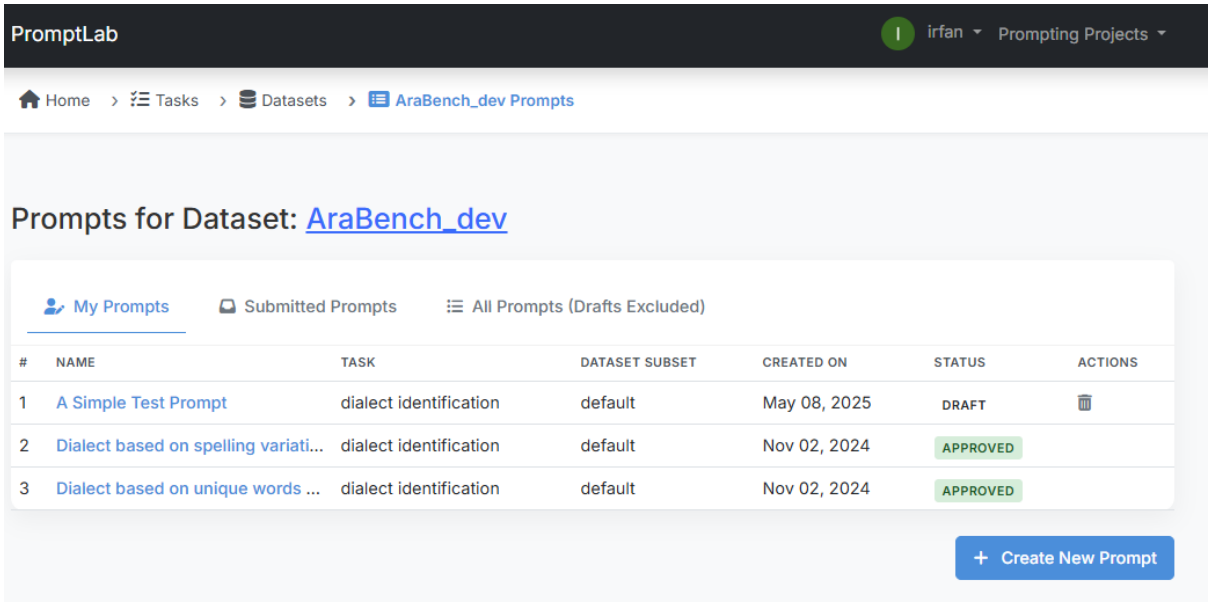

Figure 24: PromptLab: Prompt list for a specific dataset for a specific user.

a smooth transition from broad exploration to personal accountability. In the refined interface focusing on "My Prompts (Figure 24," researchers find only those prompts they have authored. This personal view simplifies the cognitive load, enabling them to quickly identify which prompts need attention, revision, or further annotation. By separating personal work from the larger collaborative corpus, the interface ensures that researchers can maintain a clear, well-defined workflow—even as the number of prompts grows across various tasks and datasets.

When working with datasets used in multiple tasks—such as "ANS stance", which may span multiple related tasks (e.g., stance detection, claim verification)—the system's interface offers convenient filters directly above the prompt listings. These thematic filters (e.g., "stance detection prompts," "claim verification prompts," or a unified "all ANS-stance prompts" view) empower researchers to toggle between different thematic slices of the dataset's prompt inventory, streamlining the discovery of prompts relevant to their current focus. By segmenting prompts along conceptual lines, the interface enhances both navigability and methodological clarity as illustrated in Figure 25.

The dedicated prompt creation interface encapsulates the platform's commitment to facilitating comprehensive, high-quality prompt generation. When initiating a new prompt for a dataset (e.g., AraBench as shown in Figure 26), users encounter a dual-pane view: on one side, the platform provides contextual dataset information, including links to external resources (like Hugging Face pages), summaries of dataset content, and controls for selecting subsets or splits. On the other side, a streamlined prompt creation form guides researchers through the process of defining prompt characteristics—such as the answer choices relevant to dialect classification, specifying tags for metadata, and setting the appropriate text direction. This environment merges dataset understanding with structured prompt-authoring features, culminating in a highly informed, user-driven creation process. Once satisfied, researchers can apply templates, save, or submit their prompts for review, ensuring continuous improvement of the platform's collective corpus.

The platform provides interfaces that emphasize prompt review, iteration, and adherence to established standards of quality and consistency. The interfaces presents a dedicated workspace for applying and editing prompt templates. Here, users encounter a side-by-side view: on one side, a carefully curated prompt template offering guidance, context, and structure; on the other, a free-form text area where the researcher can craft and finalize the actual prompt content. A prominent "Apply Template" (Figure 27) button allows for effortless insertion of predefined structures, ensuring that all prompts align with the dataset's thematic requirements and desired formatting conventions. Meanwhile, the main panel encourages prompt authors to incorporate domain-specific answer choices, add descriptive tags, and configure the prompt's text direction. The synergy between these panels—contextual dataset insights on

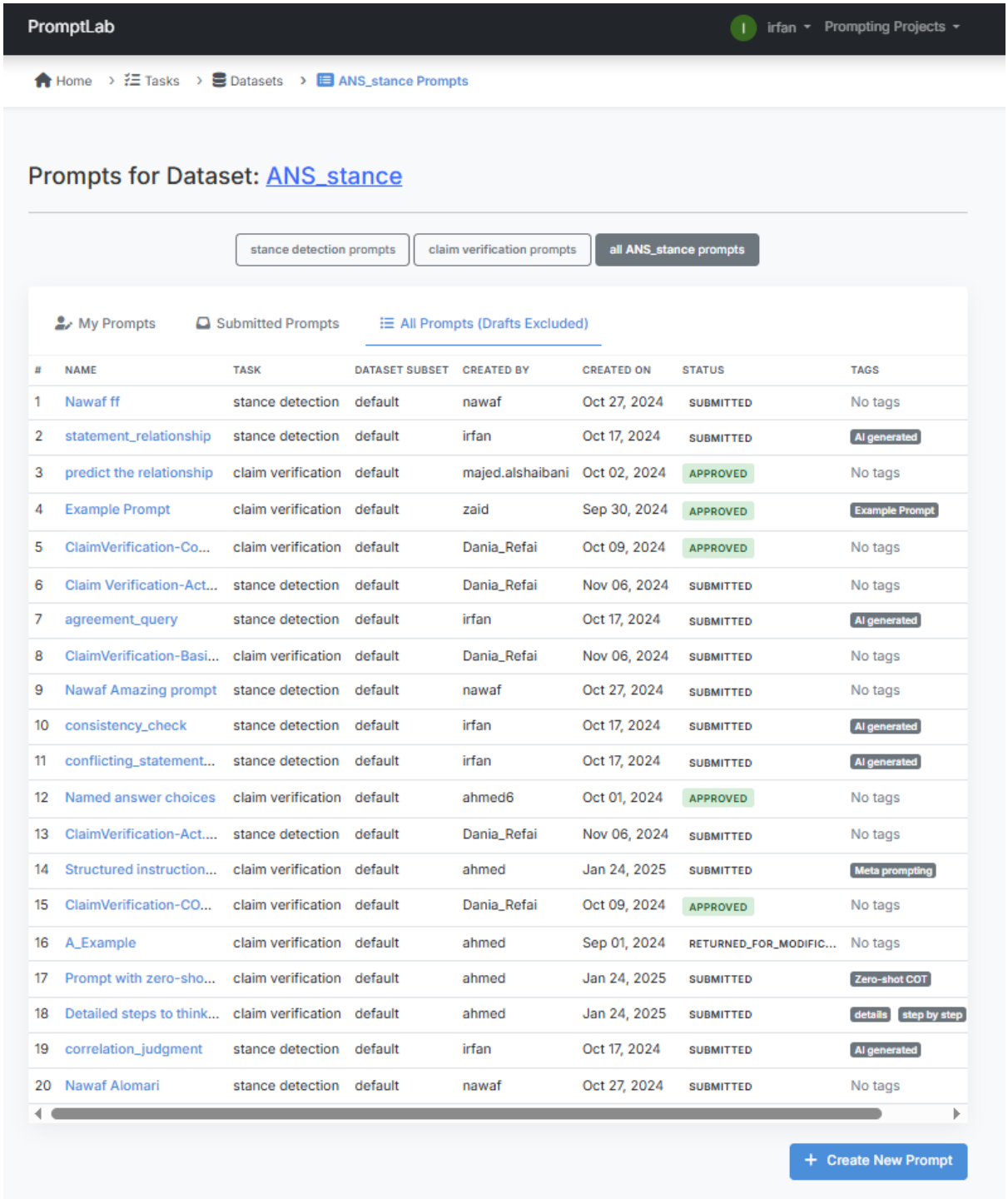

Figure 25: PromptLab: Prompt list for a specific dataset encompassing multiple tasks.

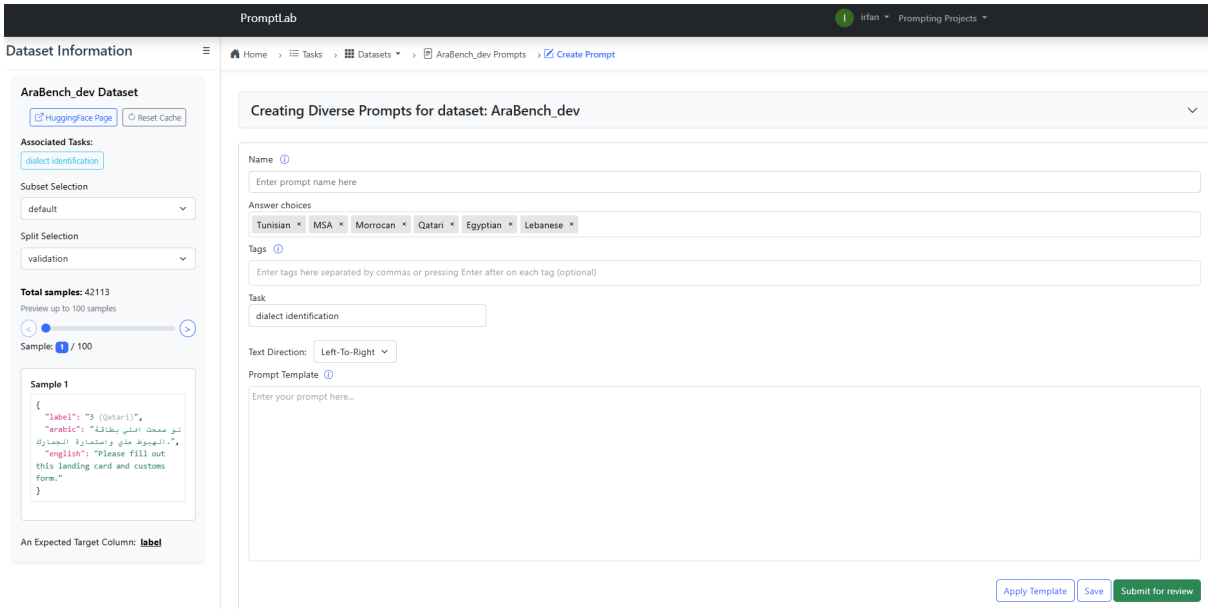

Figure 26: PromptLab: An interface to create a new prompt.

the left and customizable prompt details on the right—empowers researchers to produce prompts that are both contextually relevant and methodologically sound.

The platform also provides higher-level guidance as shown in Figure 28. An interface introduces an instructional overlay—presented as a collapsible, prominently styled banner—that encourages researchers to explore a range of prompt variation strategies. By suggesting axes such as interrogative vs. affirmative framing, localization of the task description, and implicit contextualization, this interface invites users to break out of rigid formulae and experiment with more nuanced, versatile prompt formulations. This guidance serves as a gentle pedagogical layer, reinforcing the platform's ethos: the pursuit of prompt diversity can yield more robust, generalizable model performance.

The platform offers a specialized modal window (as shown in Figure 29) that encapsulates the essence of effective prompt engineering. Within this dialogue box, users find a concise tutorial on employing Jinja syntax for dynamic prompt construction, instructions for leveraging "answer choices" as a means of fine-grained control, and recommended best practices for clarity, context, and output specification. Far from being a static reference, this modal is seamlessly integrated into the workflow: a user can consult it at any moment, refining their approach on the fly. By coupling the authoring environment with immediate, contextually relevant guidance, the platform facilitates an iterative learning process in which each new prompt crafted is better informed, better structured, and ultimately more valuable to the broader research community.

Once a prompt template has been defined, PromptLab allows users to test it directly against integrated large language models (LLMs) without leaving the interface. In the "LLM Testing" tab on the right-hand side, the user selects the desired subset and split, chooses an AI model from the model drop-down (via OpenRouter), and clicks "Test Prompt" to run the current dataset example through the full template. The response panel then displays the model's output along with basic metadata such as prompt and completion token counts, so users can inspect whether the instructions are followed, compare different models, and estimate cost. By iteratively editing the prompt, re-testing, and finally using "Apply Template", "Save", or "Submit for review", users can systematically refine prompts before applying them more broadly. This functionality is shown in Figure 30.

Below this workspace, a "Reviewer Actions" panel enables peer review. In this section, reviewers can leave comments, record decisions, and track the prompt's movement through a formalized review and approval pipeline. Such a multi-tiered interface integrates authoring, advising, and authoritative feedback into a unified process, advancing not just individual prompt quality, but also the platform's collective standards.

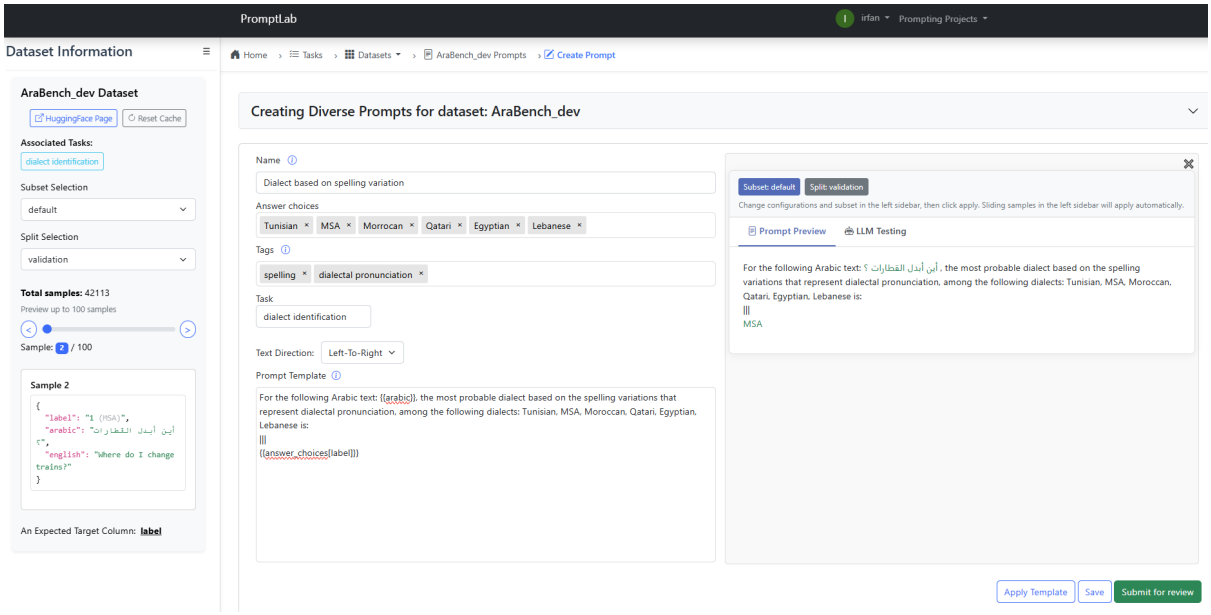

Figure 27: PromptLab: An interface apply a prompt on a dataset sample.

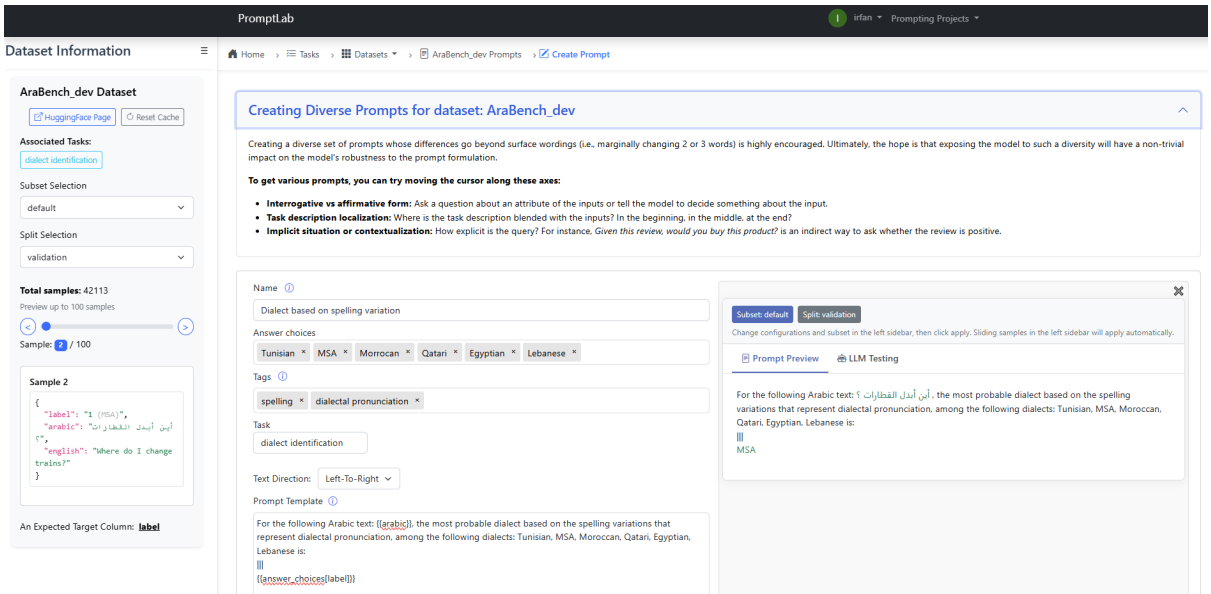

Figure 28: PromptLab: Basic guidelines on creating prompts.

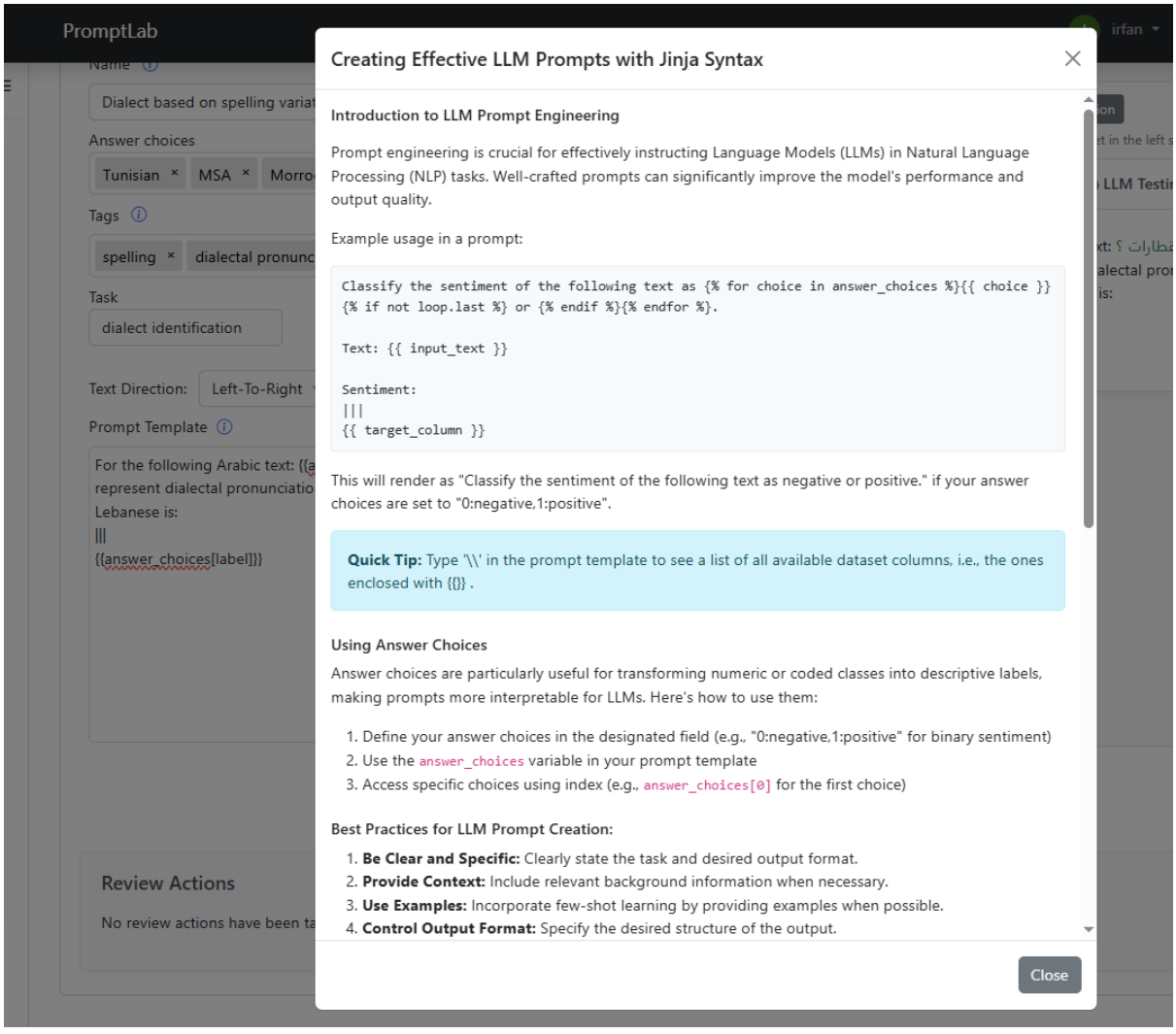

Figure 29: PromptLab: Detailed guidelines on creating prompts.

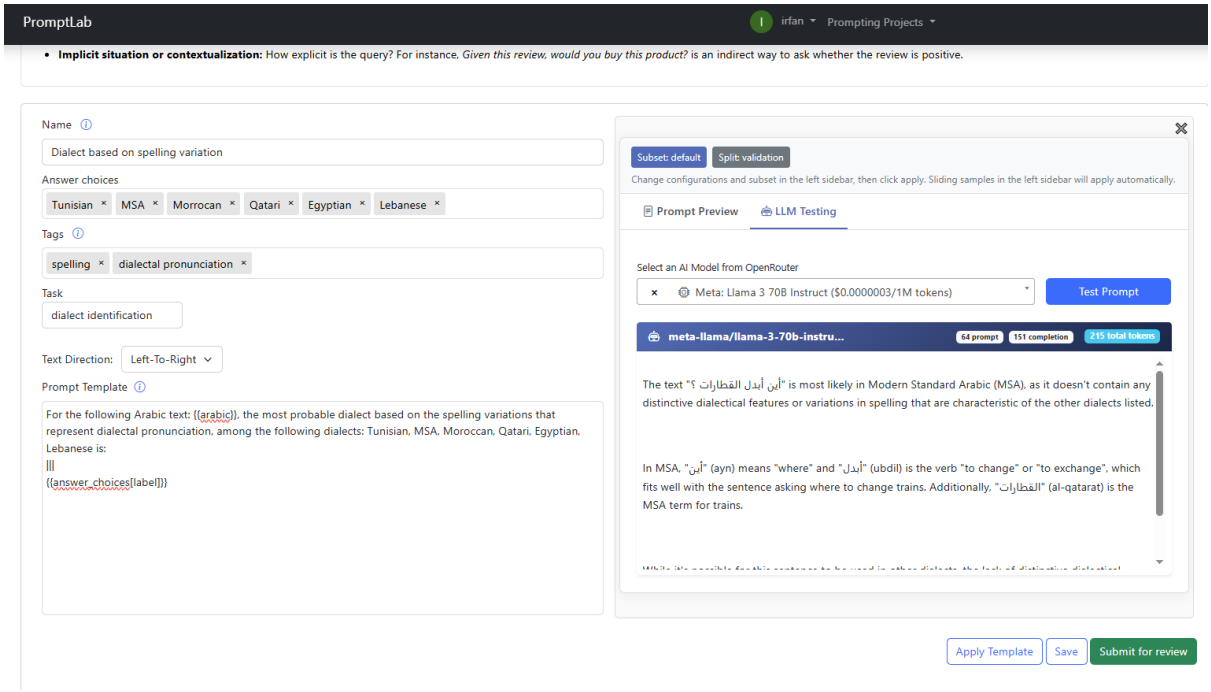

Figure 30: PromptLab: An illustration of testing a prompt on an LLM.

The “Reviewer Actions” panel operates as a real-time checkpoint for quality control and finalization. Positioned at the bottom of the prompt editing or viewing interface, this section enables reviewers to document their feedback and indicate a formal decision (e.g., “Approved,” “Returned for Modification”) as illustrated in Figure 31. By coupling the refinement process with immediate evaluative commentary, the platform ensures that the progression from initial draft to finalized prompt is both rigorous and responsive. Users can integrate feedback promptly, leading to a continuous improvement loop that enhances the quality and relevance of the entire prompt repository.

In some cases, a prompt may not meet the necessary criteria on its first submission, requiring authors to revisit their work. When “Returned for Modification,” a prompt re-enters the editing stage with actionable guidance on how to improve its clarity, formatting, or content. In this interface (Figure 32), prompt authors encounter any attached reviewer comments directly adjacent to their prompt creation tools. This spatial juxtaposition of feedback and editing capability expedites the revision cycle—authors can immediately apply suggested changes, enhancing both efficiency and accuracy in the refinement process. In doing so, the platform streamlines the feedback loop, transforming what could be a tedious back-and-forth exchange into a targeted, productive revision session.

As the prompt creation workflow matures into a cycle of iterative refinement, the platform introduces interfaces dedicated to documenting and managing the historical progression of a prompt’s review process. The “Prompt Review History” interface (Figure 33) provides a chronological record of every significant interaction—submission events, reviewer comments, modification requests, and final approvals. Each review action is clearly timestamped and attributed to a specific contributor, offering full transparency into how and why the prompt has evolved. This historical overview fosters a culture of accountability and collaborative learning: prompt authors gain insights into recurring areas of improvement, while reviewers can revisit past decisions to ensure alignment with evolving quality standards.

The platform introduces interfaces that extend beyond manual creation and refinement, reflecting the ecosystem’s dynamic and forward-looking ethos. The first of these interfaces exemplifies a scenario in which a fully integrated prompt inventory for a given dataset is on display. Prompts are presented with a familiar tabular structure, featuring clear status indicators and minimalistic tagging fields. However, the “Actions” column now incorporates advanced functionalities accessible via a subtle dropdown icon. Researchers can choose to “Generate AI prompts from this prompt,” (Figure 34) leveraging automated

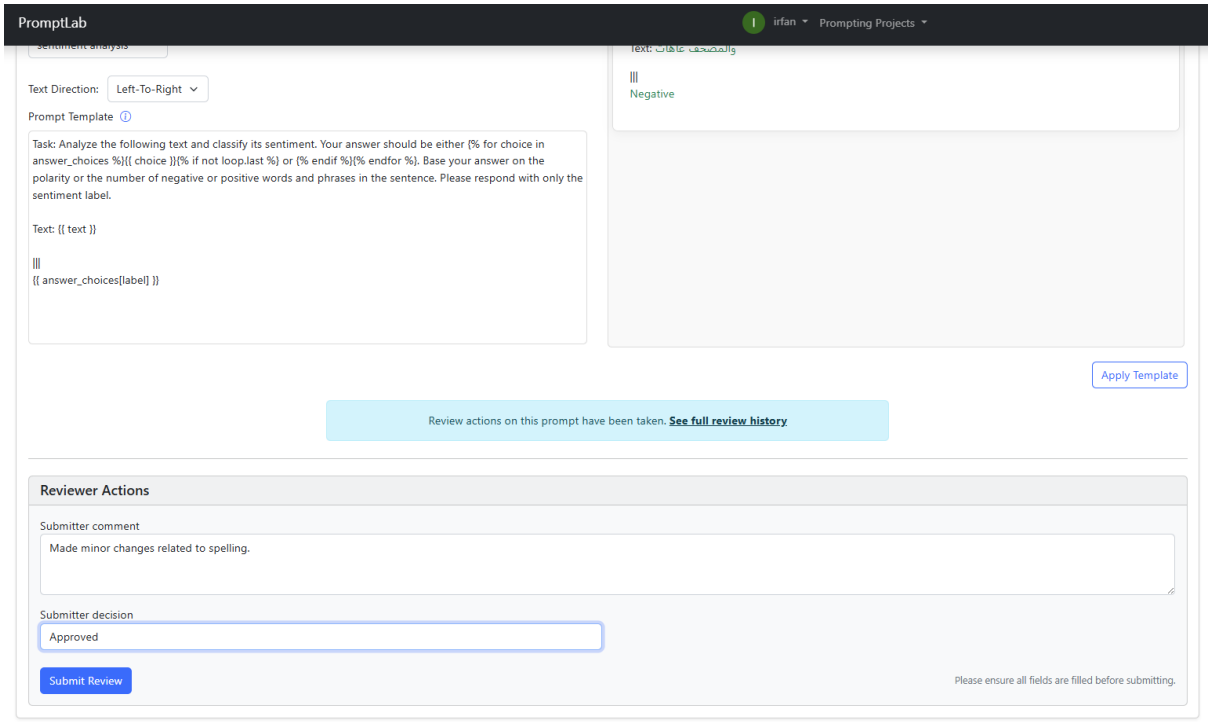

Figure 31: PromptLab: An interface to review a submitted prompt.

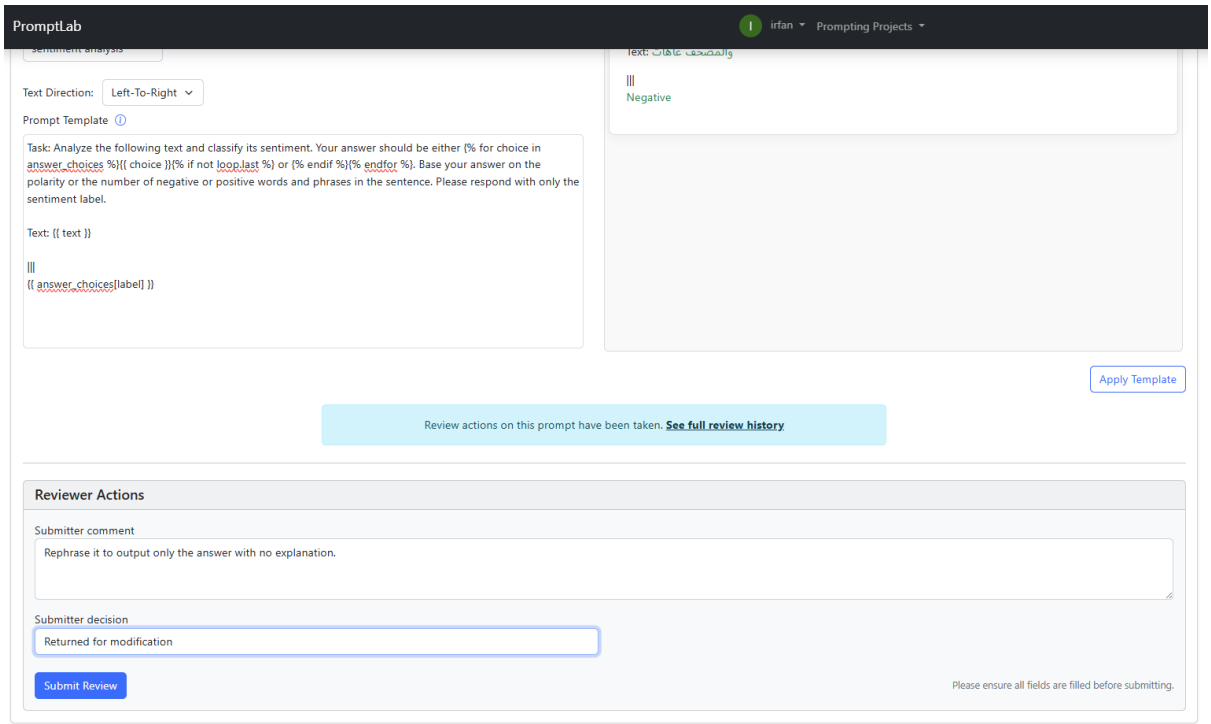

Figure 32: PromptLab: An example interface to review a submitted prompt.

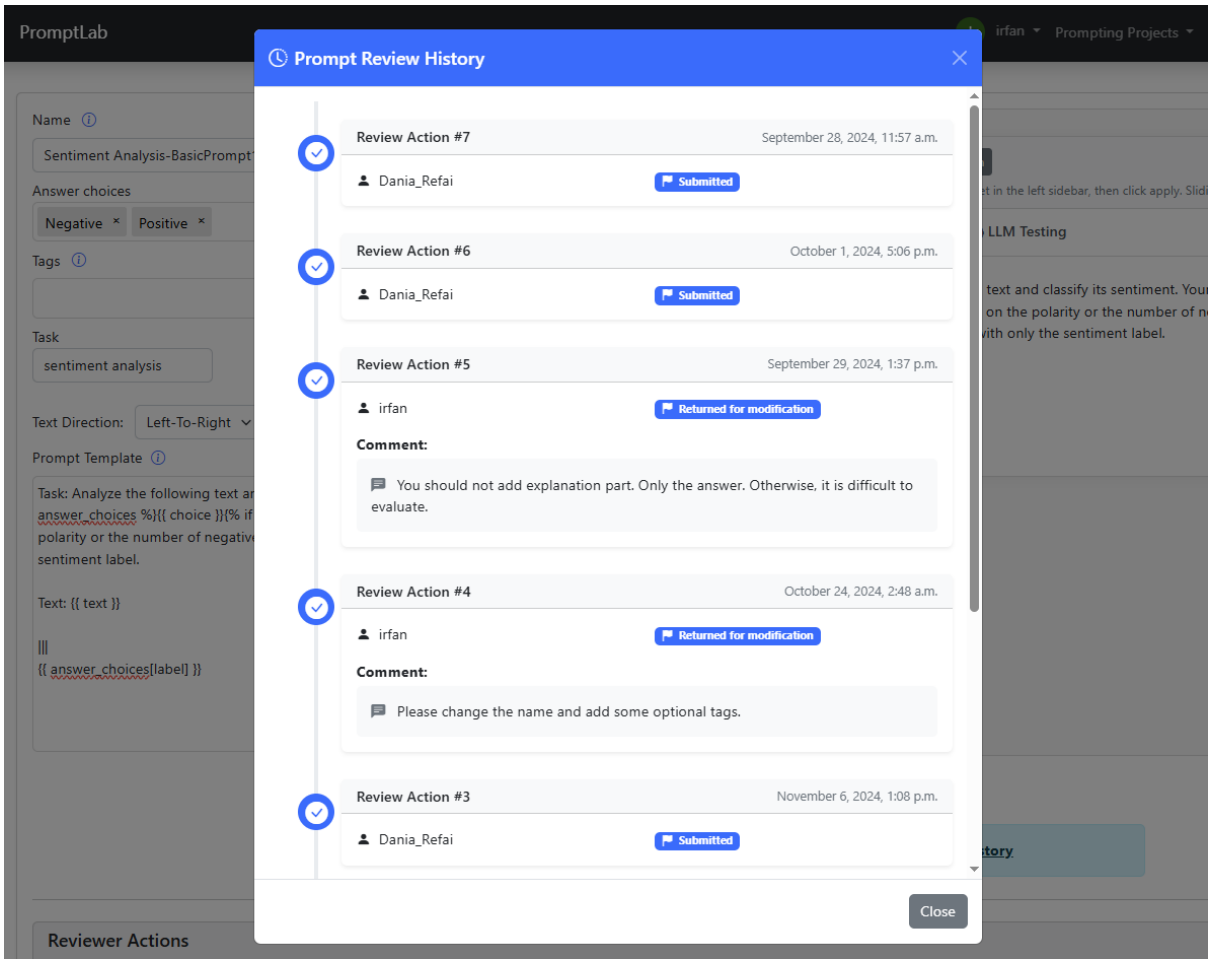

Figure 33: PromptLab: Viewing prompt review history.

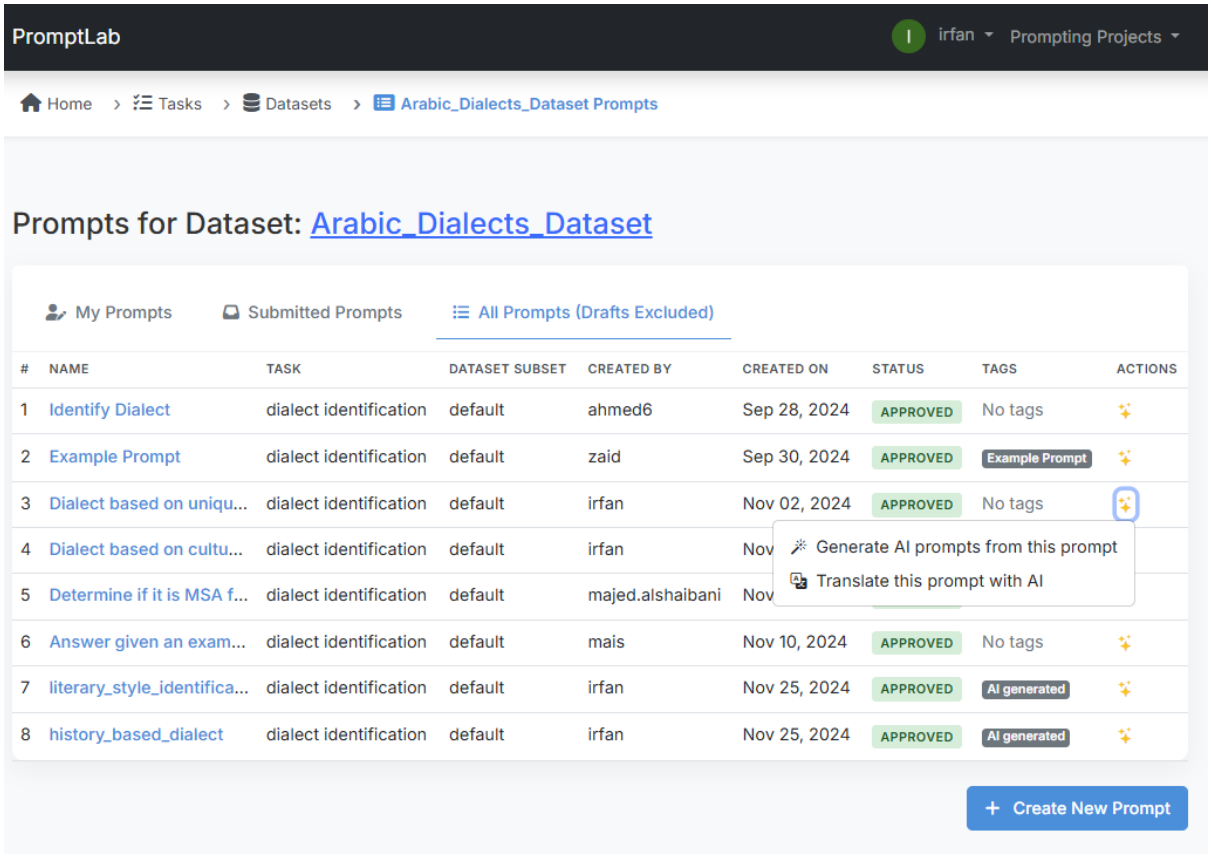

Figure 34: PromptLab: AI-assisted prompt generation.

capabilities to expand and diversify the prompt set.

On selecting the "Generate AI prompts from this prompt," option, the researcher encounters a layout consistent with previous views—dataset details on one side, customizable prompt attributes on the other (as shown in Figure 35). Yet this time, the submission workflow includes decisive action buttons: "Submit for review" and "Reject Prompt." These controls crystallize the platform's commitment to maintaining quality standards and editorial rigor. Researchers can confidently propose their newly AI-formed prompts for inclusion, aware that they will undergo a structured review cycle facilitated by the platform's features as explained before. Meanwhile, the "Reject Prompt" option provides a fallback mechanism, allowing the user to withdraw or discard those AI-generated prompts that do not meet their evolving criteria. Together, these final two interfaces encapsulate the platform's overarching philosophy: an iterative, user-driven ecosystem enriched by AI-assisted features and supported by a governance layer that ensures each prompt's relevance, integrity, and contribution to advancing NLP research.

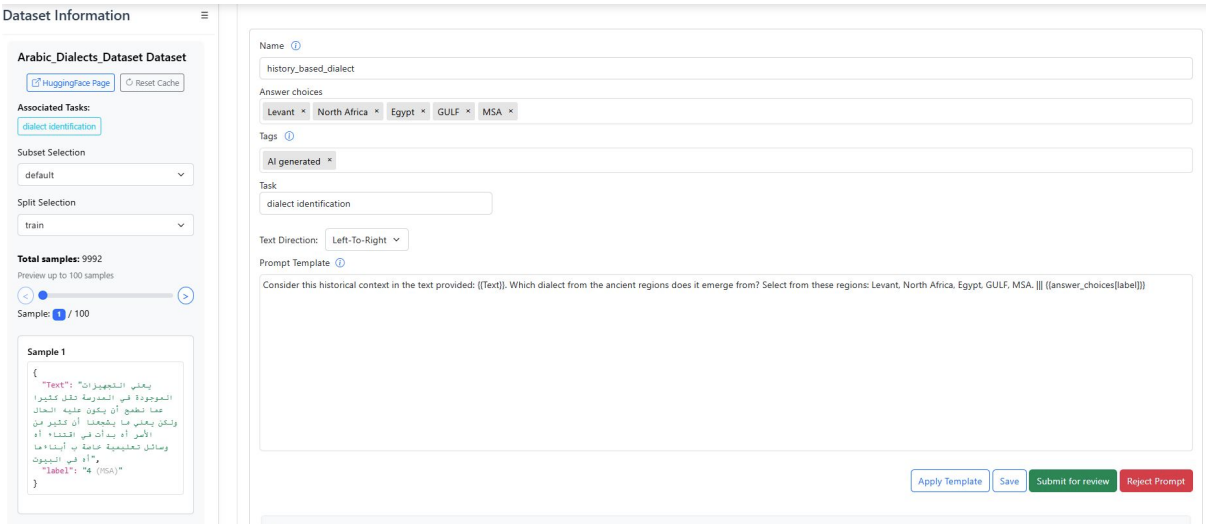

Figure 35: PromptLab: AI-assisted prompt generation an illustration.