Empowering Persian LLMs for Instruction Following: A Novel Dataset and Training Approach

Hojjat Mokhtarabadi[*], **Ziba Zamani**[†], **Abbas Maazallahi**[‡], **Mohammad Hossein Manshaei**[§]
[*]Department of Electrical and Computer Engineering, Isfahan University of Technology, Iran
[†]Department of Computer Engineering, Shahid Bahonar University of Kerman, Iran
[‡]Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[§]Department of Computer Science, Hunter College, City University of New York, NY, USA
h.mokhtarabadi@ec.iut.ac.ir, zibazamani@eng.uk.ac.ir, a.maazallahi@ut.ac.ir, mohammad.manshaei@hunter.cuny.edu

Abstract

Instruction-tuned large language models have demonstrated remarkable capabilities in following human instructions across various domains. However, their proficiency remains notably deficient in many low-resource languages. To address this challenge, we begin by introducing FarsInstruct: a comprehensive instruction dataset designed to enhance the instruction-following ability of large language models specifically for the Persian language—a significant yet underrepresented language globally. FarsInstruct encompasses a wide range of task types and datasets, each containing a mix of straightforward to complex manual written instructions, as well as translations from the Public Pool of Prompts, ensuring a rich linguistic and cultural representation. Furthermore, we introduce Co-CoLA, a framework designed to enhance the multi-task adaptability of LoRA-tuned models. Through extensive experimental analyses, our study showcases the effectiveness of the FarsInstruct dataset coupled with training by the Co-CoLA framework, in improving the performance of large language models within the Persian context. As of the current writing, FarsInstruct comprises 197 templates across 21 distinct datasets, and we intend to update it consistently, thus augmenting its applicability.

Keywords: Instruction-tuned LLMs, Low-resource languages, Parameter efficient fine-tuning

1 Introduction

The modern era of artificial intelligence is marked by numerous breakthroughs, among which is the rise of large language models (LLMs), such as GPT4 (OpenAI et al., 2024), Llama3 (Dubey et al., 2024) and PaLM (Chowdhery et al., 2022). Instruction-tuning emerges as a vital technique in the evolution of language models, involving training a model on a wide range of tasks described through natural language instructions. This method diverges from traditional task-specific fine-tuning and adapts the model's behavior to respond to user queries with relevant and helpful answers. This technique offers a more generalized and versatile approach to model training, thus contributing significantly to the advancement of LLMs.

Despite the steady progress of instruction-tuned language models, a persistent limitation remains: their difficulty in capturing the nuanced complexities of low-resource languages. This critical challenge stems from the significant gap in the availability of high-quality instruction datasets tailored to these languages. Wang et al. (2023b) highlights this concern, demonstrating that datasets lacking sufficient multilingual diversity can cause models to lose previously learned multilingual capabilities, leading to performance degradation. Moreover, translating English-centric datasets offers only partial solutions due to several inherent limitations (Naous et al., 2024; Ramesh et al., 2023; Vanmassenhove et al., 2021). While efforts have been made to compile extensive multilingual instruction datasets (Wang et al., 2022b; Singh et al., 2024; Muennighoff et al., 2022), gaps remain in creating diverse and complex prompts for languages like Persian compared to other languages.

In this study, we propose FarsInstruct, a comprehensive human-annotated instruction dataset created from existing Persian NLP datasets. It includes a mixture of manually written instructions ranging from basic to proficient language levels, alongside translations from the Public Pool of Prompts (P3) (Sanh et al., 2022), which is a collection of prompted English datasets. To ensure the diversity and representativeness of FarsInstruct, we developed 197 prompt templates derived from 21 distinct public datasets. Each prompt template comprises an input template and a target template, both of which function to extract relevant data fields from their respective datasets and reformat them into a unified structure designed for the instruction-tuning

Proceedings of the First Workshop on Language Models for Low-Resource Languages (LoResLM 2025), pages 31–67
January 20, 2025.

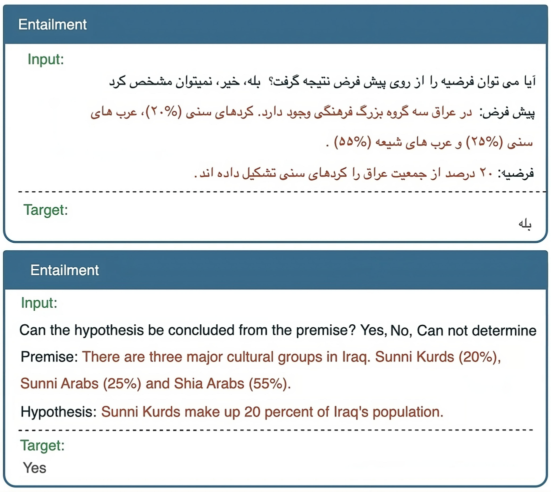

Figure 1: An example of the prompts utilized in the training process. The Persian version of the prompt is employed for training purposes, while the translated English version is provided to enhance comprehension. The instruction component is highlighted in black, the data fields are marked in orange, and the target answer is indicated in gray. In Appendix D, this example is shown in the PromptSource environment.

objective. For example, in the case of a Textual Entailment dataset containing the fields *Premise*, *Hypothesis*, and *Label*, an input template might be: "Can the hypothesis be concluded from the premise? Premise: {*Premise*}, Hypothesis: {*Hypothesis*}", while a corresponding target template could be "The answer is: {*label*}".

The collected public datasets encompass ten different task categories: Text Summarization, Textual Entailment, Text Classification, Sentiment Analysis, Word Sense Disambiguation, Query Paraphrasing, Question Answering, Reading Comprehension, Named Entity Recognition (NER), and Translation. Figure 1 depicts an instance of a prompt within our dataset after applying its respective template. A detailed overview of the FarsInsturct dataset is provided in Section 3.

Additionally, parameter-efficient fine-tuning (PEFT) methods, such as Low-Rank Adaptation (LoRA) (Hu et al., 2021), not only face challenges in multi-task settings but are also prone to catastrophic forgetting (Wang et al., 2023a; Li et al., 2024; Kalajdzievski, 2024). To address these issues, we propose Co-CoLA, a novel integration of CoLA (Xia et al., 2024) with rehearsal training (Kirkpatrick et al., 2017). More specifically, we adopt an iterative optimization framework that merges learned low-rank matrices into the model parameters and reinitializes optimization for new LoRA modules. At each iteration, we retrain a subset of data from previously learned tasks, mixing it with the current task's data during training. This periodic revisiting of earlier tasks ensures that the model retains performance across both old and new tasks, all while preserving computational efficiency. Section 4 presents an in-depth explanation of the Co-CoLA method.

In summary, our contributions to advancing Persian instruction understanding are threefold: (1) We present FarsInstruct, a comprehensive human-annotated instruction dataset for Persian, covering varied and representative tasks for different categories such as text summarization, named entity recognition, and translation. (2) We introduce Co-CoLA, a method that combines CoLA with rehearsal training to mitigate catastrophic forgetting in multi-task learning. (3) We release FarsInstruct as an open-source resource, with a commitment to its continued expansion to include a broader range of tasks and modalities[1,2].

2 Related work

Instruction-tuning: Instruction tuning refers to the process of training language models using specific input-output pairs derived from diverse data sources. This approach enhances the ability of a pre-trained LLM to interpret and respond to a wide range of human requests expressed in natural language. Instruction datasets used for this purpose are typically created in one of three ways: (1) manually created by researchers from existing NLP datasets (Wang et al., 2022b; Wei et al., 2021), (2) synthesized by prompting proprietary models with a small, seed dataset (Taori et al., 2023; Wang et al., 2022a; Honovich et al., 2023), or (3) generated entirely from scratch, involving human-written prompt-response pairs (Conover et al., 2023; Köpf et al., 2024). In this work, we adopt the first approach to develop FarsInstruct. Previous works such as FLAN (Wei et al., 2021) and P3 (Sanh et al., 2022) have been instrumental in advancing instruction dataset creation. FLAN encompasses over 60 NLP datasets, while P3 features more than 2,000 prompts from 177 datasets, each significantly contributing to the field. SuperNaturalInstruction (Wang et al., 2022b) further advanced the field by assembling a comprehensive benchmark

[1]`https://huggingface.co/datasets/PNLPhub/FarsInstruct`

[2]`https://github.com/Hojjat-Mokhtarabadi/FarsInstruct`

featuring 1,616 expert-written NLP tasks, covering 76 unique task types, and extending support to multiple languages. xP3 (Muennighoff et al., 2022) expanded on P3's groundwork by including content from 46 languages, adding new tasks like Translation and Program Synthesis that P3 had not tackled. Similarly, Aya (Singh et al., 2024) represents a major multilingual effort, featuring an extensive dataset of 513 million instances across 114 languages. This was achieved through a global collaboration involving fluent speakers who contributed instructional content. Our dataset distinguishes itself from these collections in its depth and adaptability, especially with the inclusion of more challenging tasks in Persian, offering a high level of detail not found in many multilingual efforts. While most such projects primarily use machine translations and cover a narrow range of tasks, our dataset presents a wide array of culturally and linguistically rich tasks.

Parameter effecient fine-tuning: Conventional full-parameter fine-tuning becomes computationally impractical as model size and the number of downstream tasks increase. To address this challenge, recent advancements in PEFT methods advocate for training only a small subset of parameters while leaving the majority of pre-trained model parameters intact. One of the most widely utilized paradigms in PEFT is Low-Rank Adaptation (LoRA) (Hu et al., 2021). LoRA modifies only a small, low-rank portion of the model's weights by incorporating low-rank matrices into the model's weights during the training process. Despite the significant computational advantage of LoRA, it falls short in multi-task adaptation, Additionally, Kalajdzievski (2024) demonstrated that PEFT techniques, including LoRA, remain vulnerable to catastrophic forgetting, where models lose previously acquired knowledge when fine-tuned on new tasks. MultiLoRA (Wang et al., 2023a) addresses the limitations of LoRA by reducing the dominance of top singular vectors, horizontally scaling LoRA modules, and altering the initialization of adaptation matrices, which leads to improved performance across multiple tasks with minimal additional parameters. MixLoRA (Li et al., 2024) introduces multiple LoRA-based experts within a frozen pre-trained model using a top-k routing strategy to efficiently distribute tasks, independently configure attention layer adapters, and apply auxiliary load balance loss, significantly enhancing performance while reducing GPU memory consumption and training latency. Further, CoLA (Xia et al., 2024) introduces an iterative optimization framework designed to improve the fine-tuning of LLMs by employing multiple iterations of LoRA. In this paper, we design Co-CoLA to address the issue of catastrophic forgetting, while ensuring an effective multi-task adaption.

3 FarsInstruct Dataset

With about 130 million[3] speakers, Persian — also referred to as Farsi in Iran — is an important language in the Middle East and Central Asia. FarsInstruct represents a project to provide a comprehensive public instruction dataset for the Persian community. As of this writing, FarsInstruct has 197 carefully designed and created prompt templates for 21 already-published public datasets and some translations from existing prompted datasets. Unlike multilingual collections focusing on common tasks such as Text Summarization and Question Answering, FarsInstruct introduces more task types, including Named Entity Recognition and Word Sense Disambiguation. The creation procedure, statistics, task augmentation, and quality of the dataset are covered in detail in the following subsections. Additional illustrations and tables are provided in the Appendix B, C, D.

3.1 Dataset Construction

The development of FarsInstruct entailed transforming Persian NLP datasets into their prompted format, described in plain language. This process involved a combination of manual ideation, during which our team meticulously brainstormed and refined prompt templates, along with invaluable insights from Persian language instructors. For datasets with multiple data fields, prompts were crafted to interrelate these fields, as elaborated in Section 3.2. Additionally, synonyms were employed to diversify the instructions within the prompts and reduce repetition. Each prompt template falls into one of two classes: categorization or generation. Categorization prompts guide the model in classifying text into predefined categories from dataset labels or identified through dataset analysis. In contrast, generation prompts require the model to produce full-length text, such as summarizing longer texts or answering questions based on the provided information. These instructions

[3] https://en.wikipedia.org/wiki/Persian_language

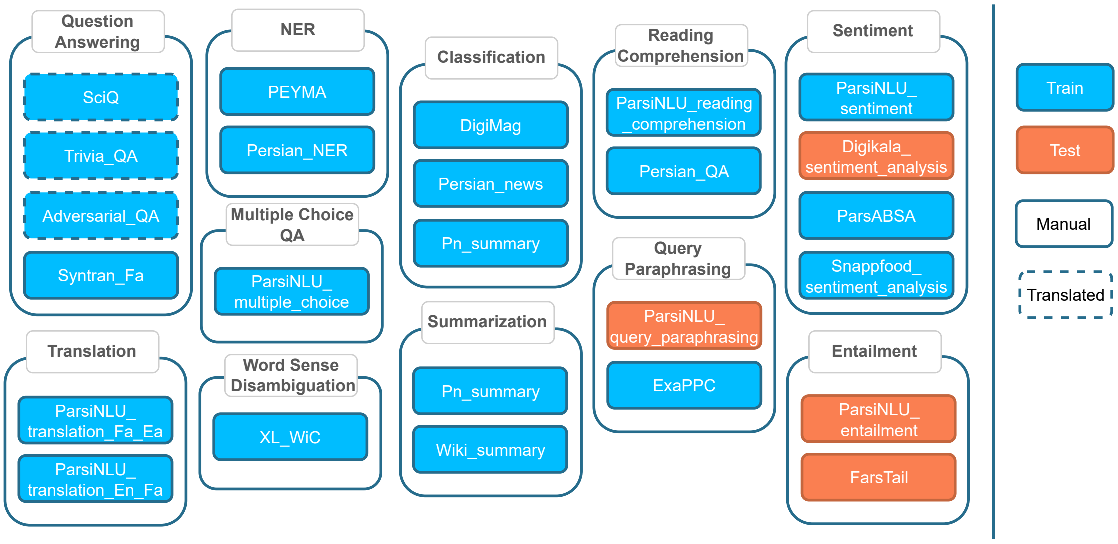

Figure 2: The detailed depiction of 11 task types utilized in our dataset. Each box within the figure lists the specific datasets associated with the respective task type. Datasets designated for training are highlighted in blue, and those reserved for testing are marked in orange. Additionally, manual datasets, which have been specifically curated and prompted by our team, are enclosed with solid borders. In contrast, datasets that have been translated from English to Persian are enclosed with dashed borders.

also include scenarios where the model needs to generate missing content from partial text inputs.

To efficiently create a large collection of prompts, we primarily utilized PromptSource (Bach et al., 2022), an open-source tool designed for creating, sharing, and managing prompts for NLP tasks. A key design choice in Bach et al. (2022) is the use of Jinja2[4] as a templating language, providing the flexibility crucial for crafting clear and effective prompts. Each dataset has multiple prompt templates, each of which consists of an input and a target template. These templates map raw data fields into natural language, structuring both the input and target sequences. Practically, templates allow users to mix arbitrary text with data fields. We refer to the text within the input template that guides the model's behavior as "Instruction". Additionally, each prompt template documents essential metadata, including evaluation metrics and the language used.

The PromptSource toolkit offers an interface for interactively writing prompts on datasets. However, the original version did not support Persian, so we modified its source code to handle Persian datasets. Our updated version is publicly available, providing the Persian community with a tool to simply create and develop prompts[5]. Appendix D depicts an illustration of the PromptSource interface with an example of a Textual Entailment dataset. Moreover, since this system was originally integrated with Huggingface Datasets library (Lhoest et al., 2021), we gathered datasets from various sources and consolidated them into a unified public repository on HuggingFace. Appendix D provides a sample of the crafted prompt templates for different datasets.

In addition to manual templating, we have decided to translate a subset of three question-answering datasets from the P3 collection (Sanh et al., 2022). This decision was made to enhance the comprehensiveness and utility of our work by providing a broader scope of data. To ensure a high-quality translation, we utilized the No Language Left Behind (NLLB) (Costa-jussà et al., 2022) machine translation model, capable of single-sentence translations between 200 languages and dialects in various scripts. We employed the largest NLLB model with 3.3B parameters to achieve the best performance. A complete list of manually templated and translated datasets is given in Figure 2.

The final dataset is standardized through a series of preprocessing steps like deduplication and

[4] https://jinja.palletsprojects.com/en/3.1.x/

[5] https://github.com/Hojjat-Mokhtarabadi/promptsource

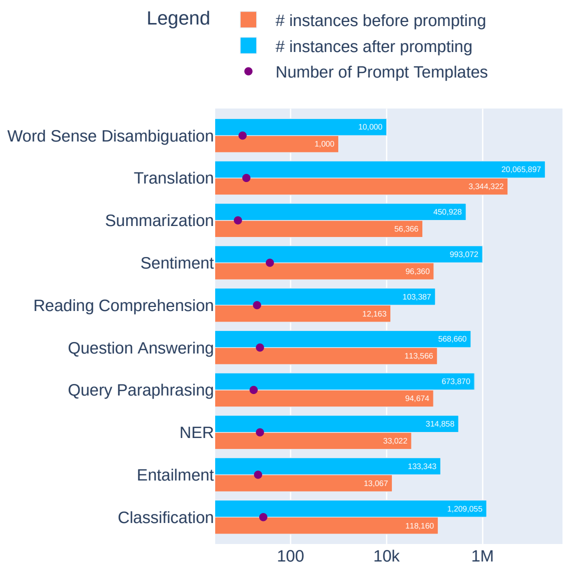

Figure 3: Distribution of NLP tasks across the FarsInstruct dataset, highlighting the expanded data volumes after applying prompt templates and the number of prompts designed per task type. For each dataset, the final size is determined by multiplying the number of samples (N) by the number of prompt templates (M), resulting in a dataset size of N*M.

removing irrelevant elements (HTML tags, hyperlinks, emojis, and offensive language). Figure 3 shows the distribution of tasks across FarsInstruct, with Table 1 listing the total number of categorization and generation prompts for each task type.

3.2 Task Augmentation and Quality Control

Instruction-tuned language models are known for their significant benefits from exposure to a broad array of tasks. In this regard, we aimed to diversify the tasks through two approaches. First, we phrased the instructions at varying language levels, ranging from basic to advanced. Second, building on best practices outlined in the FLAN Collection (Longpre et al., 2023), T0 (Sanh et al., 2022), and MetaICL (Min et al., 2022), we enhanced task diversity by mixing and swapping different data fields within a given dataset. For instance, while a dataset may initially assess a model's ability to answer question X based on input Y, we train the model to generate question X when provided with answer Y, thereby effectively broadening the range of prompts available within a limited data pool.

To ensure the accuracy and cultural relevance of the instructions, we incorporated public input and expert evaluations. Feedback was gathered from 15 randomly selected individuals and three experts in Persian literature and psychology. Participants were asked to help craft instructions in various writing formats, including formal and informal styles, and to express the same instruction in different ways, then two psychology experts and one literature professor were consulted to refine the instructions. Their expertise informed revisions, ensuring that the responses were grammatically and linguistically correct and resonated with the general Persian-speaking population. Further, the datasets adopted in FarsInstruct are predominantly used for single-task fine-tuning, as their widespread use indicates higher quality.

Task Type	**Cat**	**Gen**
Question Answering	1	9
Translation	2	10
NER (Named Entity Recognition)	4	19
Multiple Choice QA	9	1
Word Sense Disambiguation	10	0
Classification	15	12
Summarization	4	15
Reading Comprehension	2	18
Query Paraphrasing	10	7
Sentiment Analysis	24	13
Textual Entailment	16	5

Table 1: List of task types, along with the number of categorization and generation prompts dedicated to each task type. The expanded version of this table can be found in the Appendix C.

4 Methodology and Experimental Setup

To maintain our model's robustness and generalization capabilities, we integrate the CoLA framework (Xia et al., 2024) with continual learning (Kirkpatrick et al., 2017). This section offers a thorough overview of the training procedure and evaluation setup.

4.1 Training Procedure

Given the significant computational demands of full fine-tuning, we aim to employ LoRA for the training procedure, specifically using the FarsInstruct dataset. However, as noted in the studies by (Wang et al., 2023a; Li et al., 2024), LoRA tends to underperform in multi-task training scenarios due to its limitations in capturing complex interactions between tasks, leading to suboptimal performance. To mitigate this challenge, Chain of LoRA (CoLA) (Xia et al., 2024), presents an iterative opti-

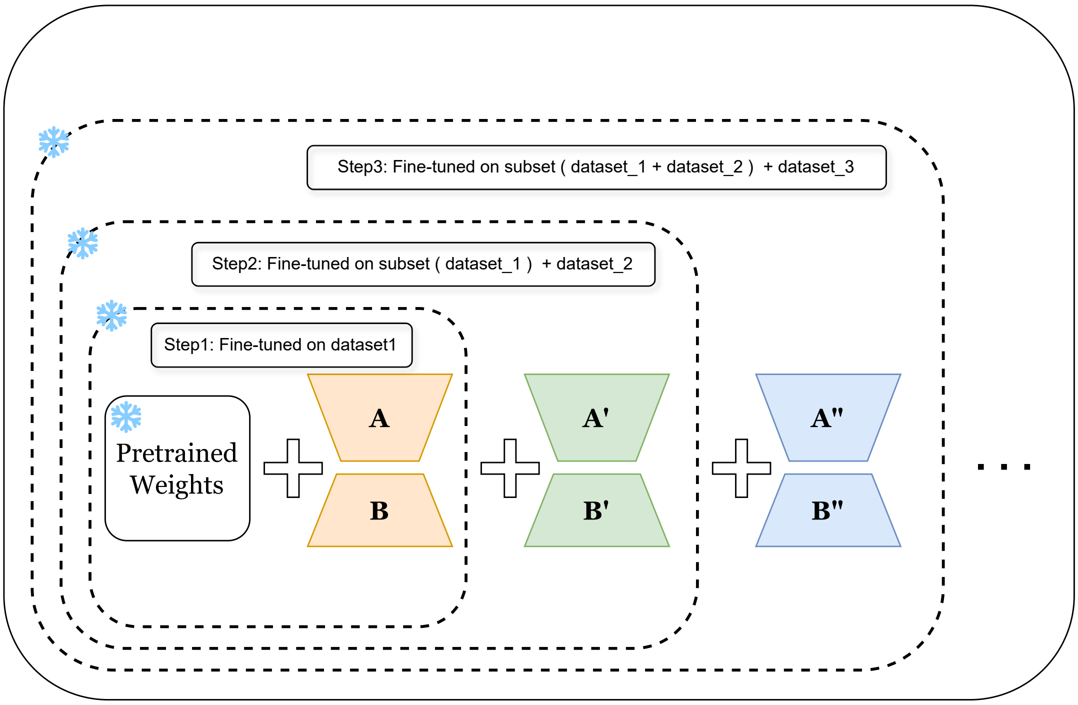

Figure 4: The Continual-Chain of LoRA training procedure, containing Tuning, Merging, and Expanding. In Step 1, the pretrained language model is LoRA-tuned on dataset_1, with the replay memory initialized as empty and merged. In Step 2, the model is expanded with a new LoRA module and further tuned on a subset of dataset_1, determined by the rehearsal hyperparameter, alongside dataset_2, preparing it for Step 3. This process is iteratively repeated in subsequent steps.

mization framework based on the principles of the Frank-Wolfe algorithm (Frank et al., 1956). This method involves an iterative process of LoRA fine-tuning on a single task, merging the learned parameters with the base model, and reinitializing with a new LoRA module. Xia et al. (2024) shows this process allows the model to learn higher-rank adaptations more effectively. Another persistent challenge affecting the performance of LoRA-tuned models is catastrophic forgetting. Kalajdzievski (2024) analyzed this phenomenon and revealed that forgetting significantly undermines both model safety and performance on reasoning benchmarks.

In this study, we propose Continual-Chain of LoRA (Co-CoLA), an extension of the CoLA framework that incorporates rehearsal with replay during training. More specifically, rehearsal training is an approach within the continual learning framework that involves revisiting a portion of previously learned tasks while training new tasks. The core mathematical operation in LoRA involves updating the low-rank matrices A and B, which are applied to modify the transformer layers of the model. The update rule can be expressed as $W' = W + BA$ where W represents the transformer layer's original weights, and W' shows the updated weights after applying the low-rank adjustments A and B. Essentially, Co-CoLA structures the training procedure by iterating over the following three phases:

Tuning: Following the standard LoRA approach, the weights of the base model remain frozen, while only the model's LoRA parameters, represented by matrices A and B are fine-tuned. During this phase, a subset of previously trained data is replayed along with the new data. Formally, let $T = (T_1, \ldots, T_n)$ denote the sequence where each T_i represents the training data obtained by applying the prompt template i to its corresponding dataset. The training data augmented with rehearsal is defined as:

$$T_i^r = T_i \cup \left(\sum_{j=1}^{i-1} rT_j \right) \tag{1}$$

where r is the rehearsal hyperparameter that controls the percentage of examples sampled from previous tasks $T_1, \ldots, T_{i-1}$.

Merging: After the tuning phase, the newly updated LoRA parameters are merged with the existing model weights based on the standard method in Hu et al. (2021). These merged weights are fixed and do not receive any gradient update in subsequent steps.

Expanding: The final phase involves preparing the model for subsequent training rounds by reinitializing the LoRA modules with new parameters (A' and B'). Following Hu et al. (2021) A' adopts Gaussian initialization and B' is initialized to zero.

An illustration of this iterative three-staged approach is provided in Figure 4.

4.2 Evaluation Setup

The performance of our model is assessed across two categories of task types: those included in the training dataset ("Held in") and those introduced for the first time during evaluation ("Held out"). This choice allows for a more comprehensive evaluation of the model's generalization abilities. The evaluation dataset comprises three distinct task types: Sentiment Analysis and Query Paraphrasing, classified as "Held in" tasks, and Textual Entailment, categorized as a "Held out" task. As shown in Figure 2, the evaluation includes one dataset each for sentiment analysis and paraphrase identification, as well as two datasets specifically for entailment tasks.

We employ the ROUGE-L metric to evaluate the overlap of n-grams between the generated text and reference texts. Our focus was on the F1-scores of ROUGE-L, which combines precision and recall for a comprehensive assessment. As shown by Wang et al. (2022b), the rankings generated by this metric correlate strongly with accuracy for categorization templates.

Task	Type	Model	ROUGE-L
parsinlu query paraphrasing	Held In	Aya-13B	45.58
		PersianMind-7B	17.07
		Mistral-7B	6.89
		Ava-8B	6.67
		Ava-LoRA-8B	8.73
		CoLA-8B	20.88
		Co-CoLA-8B	**45.86**
Digikala Sentiment Analysis	Held In	Aya-13B	28.41
		PersianMind-7B	18.19
		Mistral-7B	2.46
		Ava-8B	8.69
		Ava-LoRA-8B	5.72
		Ava-LoRA-8B	5.72
		CoLA-8B	25.62
		Co-CoLA-8B	**40.87**
FarsTail	Held Out	Aya-13B	**37.61**
		PersianMind-7B	17.05
		Mistral-7B	5.74
		Ava-8B	12.48
		Ava-LoRA-8B	9.07
		CoLA-8B	15.64
		Co-CoLA-8B	36.35
Parsinlu Entailment	Held Out	Aya-13B	42.64
		PersianMind-7B	4.45
		Mistral-7B	4.93
		Ava-8B	15.04
		Ava-LoRA-8B	7.18
		CoLA-8B	22.55
		Co-CoLA-8B	**55.32**

Table 2: ROUGE-L F1 Scores for Different Models across Tasks

5 Results

To investigate the applicability of FarsInstruct, we instruction-tuned Ava—a Llama-3-based Persian LLM—using the Co-CoLA framework across a suit of tasks. The results were compared with monolingual and multilingual instruction-tuned models, using quantitative evaluations. For a comprehensive overview of the training configuration, please refer to the Appendix A.

5.1 Quantitative Evaluation

We evaluate our model against several existing models fine-tuned on instruction data. Specifically, PersianMind (University of Tehran, 2024) is a Llama-2 7B-based model, trained in 3 phases on different Persian datasets. Though its training data is unavailable, Ava (Moghadam, 2024) is a newly introduced model, fine-tuned on the Llama-3 8B model for Persian tasks. Aya (CohereForAI, 2024) is a 13B encoder-decoder model trained on a subset of 25 million samples from the Aya dataset and Mistral-7B (MistralAI, 2024) is a decoder-only model trained on publicly available prompted datasets.

Table 2 summarizes the comparative performance of various models, including our proposed method, Co-CoLA, across several NLP Datasets: ParsiNLU Query Paraphrasing, Digikala Sentiment Analysis, FarsTail, and ParsiNLU Entailment. These models are evaluated using ROUGE-L F1 scores. As illustrated in Table 2, Co-CoLA performs comparably well to the Aya model, de-

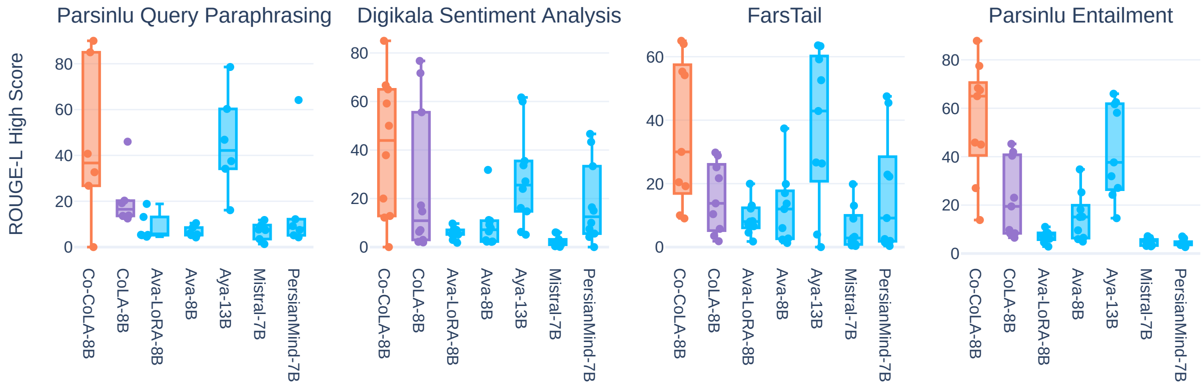

Figure 5: Comparative performance of different models on Persian language tasks using the ROUGE-L metric. The bar chart depicts the superior performance of Co-CoLA across multiple tasks, particularly excelling in the ParsiNLU Entailment task.

spite having fewer parameters and being trained on less instruction data and significantly outperforms all other models, indicating the effectiveness of Co-CoLA. The factors contributing to this performance gap are further discussed in Section 6. Moreover, the scores of Ava-LoRA, reflecting the performance of raw LoRA fine-tuning of Ava on FarsInstruct and naive CoLA are inferior to those achieved with Co-CoLA training, highlighting the effectiveness of our method.

6 Discussion

Figure 5 provides a detailed breakdown of the overall performance reported in Table 2. Each dot in the plot represents the ROUGE-L F1 score of the given model on the selected template. As clearly illustrated, other Persian instruction-tuned models fail to achieve a high ROUGE-L F1 score. One significant factor contributing to this disparity is the low precision score. The F1 score combines precision and recall and serves as a comprehensive metric for evaluation. Precision measures the proportion of the longest common subsequence (LCS) in the candidate text that matches the reference text, while recall measures the proportion of the LCS in the reference text that is present in the candidate text. Although these models achieve acceptable recall scores, they fall short in precision, a critical metric for categorization templates. In contrast, Aya demonstrates proficiency in handling both generation and categorization templates within the Persian context. Compared to Aya, Co-CoLA enhances the model's ability to manage both categorization and generation tasks effectively while being less computationally expensive. Despite the limited success of continual learning frameworks, the study by Scialom et al. (2022) demonstrated that continual training of language models, such as T0 (Sanh et al., 2022) with rehearsal, can effectively help them in comprehending new instructions via instruction composition. Our results confirm this finding within the Chain-of-LoRA framework, resulting in better generalization and improved performance on new tasks.

7 Conclusion

This study aims to address the limitations in instruction-following capabilities of language models for Persian, an important but underrepresented language, by introducing a novel instruction dataset and a training approach specifically designed to enhance the instruction comprehension of large language models. FarsInstruct presents a carefully curated dataset that combines human-annotated instruction data with translations from English-centric instruction datasets, featuring tasks in different forms and from varying levels of difficulty. Further, Co-CoLA leverages the strengths of CoLA with rehearsal training to mitigate catastrophic forgetting and improve multi-task adaptation, through its iterative optimization framework. Our results demonstrate that this allows for sustained model performance over diverse tasks while optimizing computational resources. We hope our dataset fills the critical gap and serves as a valuable resource to the multilingual NLP community.

8 Limitations

This section delineates the principal limitations of our study, which, while providing substantial contributions to Persian NLP, presents certain challenges. Addressing these challenges in future developments could enhance its utility and applicability in broader linguistic contexts.

Data Diversity and Representation: Although FarsInstruct broadens the corpus of Persian language resources, it does not fully capture the rich tapestry of dialects and sociolects that characterize the Persian-speaking world. Also, the collected templates are generally biased towards short responses, which might affect the overall performance of the model.

Complexity of Instructions: The dataset prompts vary in complexity but still may not sufficiently challenge or train models to handle the types of complex instructions encountered in everyday human interactions. Real-world applications often demand a high level of interpretative depth and context awareness—qualities that current models may struggle with when trained on existing datasets. Future versions of FarsInstruct could benefit from integrating prompts that require higher-order cognitive processing, such as irony, metaphor understanding, and techniques that involve prompting the model to break down complex tasks into intermediate steps, mimicking human reasoning processes (Wei et al., 2022).

Dependency on External Datasets: The effectiveness of the FarsInstruct dataset is contingent upon the quality and variety of the external datasets. This dependency creates vulnerability, as biases or errors in source datasets may be passed to FarsInstruct. A rigorous process for source data, coupled with efforts to develop original, high-quality training materials, could diminish reliance on external datasets and enhance the overall integrity of the dataset.

Evaluation Metrics: The metrics currently used to evaluate models trained on FarsInstruct may not fully capture the nuanced and multifaceted aspects of language comprehension and generation. Furthermore, for certain tasks such as rewriting, ROUGE-L may not serve as an adequate measure of quality.

Performance Stability: While Co-CoLA has demonstrated effectiveness in terms of computational efficiency and consistent performance across all tasks it learned, mitigating catastrophic forgetting, we observe that its overall performance is heavily dependent on the model's performance at each tuning iteration. We leave potential solutions to this problem to future work.

References

Hossein Amirkhani, Mohammad AzariJafari, Soroush Faridan-Jahromi, Zeinab Kouhkan, Zohreh Pourjafari, and Azadeh Amirak. 2023. Farstail: A persian natural language inference dataset. *Soft Computing*, pages 1–13.

Taha Shangipour Ataei, Kamyar Darvishi, Behrouz Minaei-Bidgoli, and Sauleh Eetemadi. 2019. Parsabsa: An aspect-based sentiment analysis dataset in persian. *CoRR*, abs/1908.01815.

Mohammad Yasin Ayoubi, Sajjad & Davoodeh. 2021. Persianqa: a dataset for persian question answering. https://github.com/SajjjadAyobi/PersianQA.

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

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the ai: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

CohereForAI. 2024. aya-101 model on hugging face. https://huggingface.co/CohereForAI/aya-101. Accessed: 2024-06-15.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe

Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L.

Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Mehrdad Farahani. 2020. Summarization using bert2bert model on wikisummary dataset. https://github.com/m3hrdadfi/wiki-summary.

Mehrdad Farahani, Mohammad Gharachorloo, and M. Manthouri. 2021. Leveraging parsbert and pretrained mt5 for persian abstractive text summarization. *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, pages 1–6.

Farhan Farsi, Sadra Sabouri, Kian Kashfipour, Soroush Gooran, Hossein Sameti, and Ehsaneddin Asgari. 2024. Syntran-fa: Generating comprehensive answers for farsi qa pairs via syntactic transformation.

Marguerite Frank, Philip Wolfe, et al. 1956. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Damjan Kalajdzievski. 2024. Scaling laws for forgetting when fine-tuning large language models. *Preprint*, arXiv:2401.05605.

Daniel Khashabi, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, Sarik Ghazarian, Mozhdeh Gheini, Arman Kabiri, Rabeeh Karimi Mahabagdi, Omid Memarrast, Ahmadreza Mosallanezhad, Erfan Noury, Shahab Raji, Mohammad Sadegh Rasooli, Sepideh Sadeghi, Erfan Sadeqi Azer, Niloofar Safi Samghabadi, Mahsa Shafaei, Saber Sheybani, Ali Tazarv, and Yadollah Yaghoobzadeh. 2021. ParsiNLU: A suite of language understanding challenges for Persian. *Transactions of the Association for Computational Linguistics*, 9:1147–1162.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024. Mixlora: Enhancing large language models fine-tuning with lora based mixture of experts. *arXiv preprint arXiv:2404.15159*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

et. al Mehrdad Farahani. 2020. Parsbert: Transformer-based model for persian language understanding. *ArXiv*, abs/2005.12515.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

MistralAI. 2024. Mistral-7B-Instruct-v0.2 model on hugging face. `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`. Accessed: 2024-06-15.

Mehdi Hosseini Moghadam. 2024. AVA-Llama-3-V2 model on hugging face. `https://huggingface.co/MehdiHosseiniMoghadam/AVA-Llama-3-V2`. Accessed: 2024-06-15.

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

Tarek Naous, Michael J. Ryan, Alan Ritter, and Wei Xu. 2024. Having beer after prayer? measuring cultural bias in large language models. *Preprint*, arXiv:2305.14456.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Hanieh Poostchi, Ehsan Zare Borzeshi, Mohammad Abdous, and Massimo Piccardi. 2016. PersoNER: Persian named-entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3381–3389, Osaka, Japan. The COLING 2016 Organizing Committee.

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A multilingual benchmark for evaluating semantic contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7193–7206, Online. Association for Computational Linguistics.

Krithika Ramesh, Sunayana Sitaram, and Monojit Choudhury. 2023. Fairness in language models beyond English: Gaps and challenges. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2106–2119, Dubrovnik, Croatia. Association for Computational Linguistics.

Reyhaneh Sadeghi, Hamed Karbasi, and Ahmad Akbari. 2022. Exappc: a large-scale persian paraphrase detection corpus. In *2022 8th International Conference on Web Research (ICWR)*, pages 168–175.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. *arXiv preprint arXiv:2205.12393*.

Mahsa Sadat Shahshahani, Mahdi Mohseni, Azadeh Shakery, and Heshaam Faili. 2018. Peyma: A tagged corpus for persian named entities. *ArXiv*, abs/1801.09936.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, et al. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. *arXiv preprint arXiv:2402.06619*.

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

Soheil Tehranipour. 2019. Digikala comments (persian sentiment analysis).

Soheil Tehranipour. 2022. Snappfood - persian sentiment analysis.

University of Tehran. 2024. PersianMind-v1.0 model on hugging face. `https://huggingface.co/universitytehran/PersianMind-v1.0`. Accessed: 2024-06-15.

Eva Vanmassenhove, Dimitar Shterionov, and Matthew Gwilliam. 2021. Machine translationese: Effects of algorithmic bias on linguistic complexity in machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2203–2213, Online. Association for Computational Linguistics.

Yiming Wang, Yu Lin, Xiaodong Zeng, and Guannan Zhang. 2023a. Multilora: Democratizing lora for better multi-task learning. *arXiv preprint arXiv:2311.11501*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023b. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36:74764–74786.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. 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, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. 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, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenhan Xia, Chengwei Qin, and Elad Hazan. 2024. Chain of lora: Efficient fine-tuning of language models via residual learning. *arXiv preprint arXiv:2401.04151*.

Appendix

A Training Configuration

All implementations were carried out using PyTorch (Paszke et al., 2019), Transformers (Wolf et al., 2020) and Accelerate (Gugger et al., 2022) libraries. For efficient training, we randomly selected 25 prompt templates and applied them to their corresponding datasets. Consequently, for instance, a dataset with two selected templates would be upsampled to twice its original size. To generate the training data for each iteration, we sampled up to 10,000 instances from the dataset based on the selected template, with the rehearsal hyperparameter of Co-CoLA set to 0.01. Following the established practices, we used Paged-AdamW as the base optimizer and trained for a total of four epochs in each tuning phase. A linear learning rate scheduler was applied, with an initial learning rate of 6×10^{-5} and a batch size of 16. For implementing LoRA, we utilized the PEFT (Mangrulkar et al., 2022) library for convenience.

B Datasets Details

- **Digikala Sentiment Analysis** (Tehranipour, 2019): A collection of Digikala product reviews labeled by customer star ratings. It categorizes sentiment into five labels (e.g., buy, not buy, neutral).

- **Snappfood Sentiment Analysis** (Tehranipour, 2022): A dataset of 70,000 user reviews from Snappfood, an online food delivery service. It contains equal numbers of positive and negative reviews (35,000 each), supporting effective sentiment analysis.

- **ParsiNLU** (Khashabi et al., 2021): A comprehensive suite for Persian NLP tasks, covering reading comprehension, multiple-choice question-answering, sentiment analysis, textual entailment, question-answering, and machine translation. These datasets are collected in a multitude of ways, often involving manual annotations by native speakers. This results in over 14.5k new instances across 6 distinct NLU tasks, serving as a key Persian NLP benchmark.

- **ExaPPC** (Sadeghi et al., 2022): A paraphrase corpus with 2.3 million Persian sentence pairs labeled as paraphrase or non-paraphrase. It includes both formal and colloquial sentences, making it ideal for models like BERT.

- **FarsTail** (Amirkhani et al., 2023): A Persian textual entailment dataset with 10,367 samples, classifying premise-hypothesis pairs into entailment, contradiction, or neutral, essential for natural language inference in Persian.

- **Pars-ABSA** (Ataei et al., 2019): A dataset for aspect-based sentiment analysis in Persian, with 5,114 positive, 3,061 negative, and 1,827 neutral data points. It is useful for studying fine-grained sentiment in reviews.

- **WikiSummary** (Farahani, 2020): A summarization dataset with 45,654 entries derived from Persian Wikipedia articles, paired with highlights designed for summarization tasks with reduced article lengths.

- **Pn-Summary** (Farahani et al., 2021): The Pn-Summary dataset contains 93,207 news articles from six news agencies, each paired with a human-generated summary. The dataset was curated from an initial pool of 200,000 articles, covering various categories.

- **PersianQA** (Ayoubi, 2021): PersianQA is a reading comprehension dataset with over 9,000 entries sourced from Persian Wikipedia, including both answerable and unanswerable questions. It supports the development of models that can recognize unanswerable queries, similar to SQuAD 2.0.

- **PersianNews** (Mehrdad Farahani, 2020): This dataset consists of 16,438 news articles from online Persian news agencies, categorized into eight classes such as Economic, International, Political, Science & Technology, and Sport.

- **DigiMag** (Mehrdad Farahani, 2020): DigiMag contains 8,515 articles from the Digikala Online Magazine, divided into seven categories including Video Games, Shopping Guide, Health & Beauty, and Art & Cinema.
- **PEYMA** (Shahshahani et al., 2018): The PEYMA dataset features 7,145 sentences with 302,530 tokens, 41,148 of which are annotated with seven entity classes, including Organization, Money, Location, Date, and Person.
- **Persian NER** (Poostchi et al., 2016): This is a manually-annotated named entity recognition dataset with 250,015 tokens and 7,682 sentences. The dataset includes six named entity classes like Person, Organization, Location, and Event, in IOB format.
- **Syntran-fa** (Farsi et al., 2024): A Farsi question-answering dataset with nearly 50,000 question-answer pairs. It extends short answers into fluent, complete responses using syntactic rules and parsing methods like stanza.
- **XL-WiC** (Raganato et al., 2020): XL-WiC is a multilingual dataset for word sense disambiguation, involving binary classification of word meaning across 12 languages, including Farsi. It evaluates models on cross-lingual semantic contextualization.
- **SciQ** (Lu et al., 2022): A multimodal dataset with 21,208 science questions from elementary and high school curricula. It covers various sciences, with questions annotated with images, lectures, and explanations, making it a rich resource for science QA.
- **TriviaQA** (Joshi et al., 2017): A large QA dataset with 950,000 question-answer pairs from Wikipedia and web documents. It is more challenging than datasets like SQuAD due to longer contexts and non-direct text spans, including human-verified and machine-generated pairs.
- **AdversarialQA** (Bartolo et al., 2020): A dataset designed to test the robustness of QA models against adversarially crafted questions. It includes adversarially modified questions from SQuAD, TriviaQA, and NewsQA to challenge model reasoning and generalization.

C Datasets Illustrations

Dataset	Categorization	Generation
DigiMag	9	1
Digikala_sentiment_analysis	9	1
ExaPPC	3	4
FarsTail	8	2
ParsABSA	5	1
ParsiNLU_entailment	8	3
ParsiNLU_multiple_choice	9	1
ParsiNLU_query_paraphrasing	7	3
ParsiNLU_reading_comprehension	1	9
ParsiNLU_sentiment	3	7
ParsiNLU_translation_En_FA	1	5
ParsiNLU_translation_FA_En	1	5
PEYMA	1	9
Persian_NER	3	10
Persian_news	3	3
Persian_QA	1	9
Pn_summary	3	8
Snappfood_sentiment_analysis	7	4
Syntran_FA	1	9
Wiki_summary	1	7
XL_WiC	10	0

Table 3: Detailed Overview of Datasets Utilized for Categorization and Generation Tasks. As shown in this table Categorization and Generation tasks are not equally distributed across all datasets. Some datasets, such as Digimag, are originally designed for categorization tasks. We have enhanced these datasets by incorporating generation prompts. Conversely, translation tasks, which are inherently generative, have been augmented with categorization prompts. This dual-purpose approach enriches the datasets, facilitating both categorization and generation tasks and providing a more versatile training and testing framework. This table provides insight into the distribution and specialization of prompts across different datasets, highlighting the balance and focus within the training and testing framework.

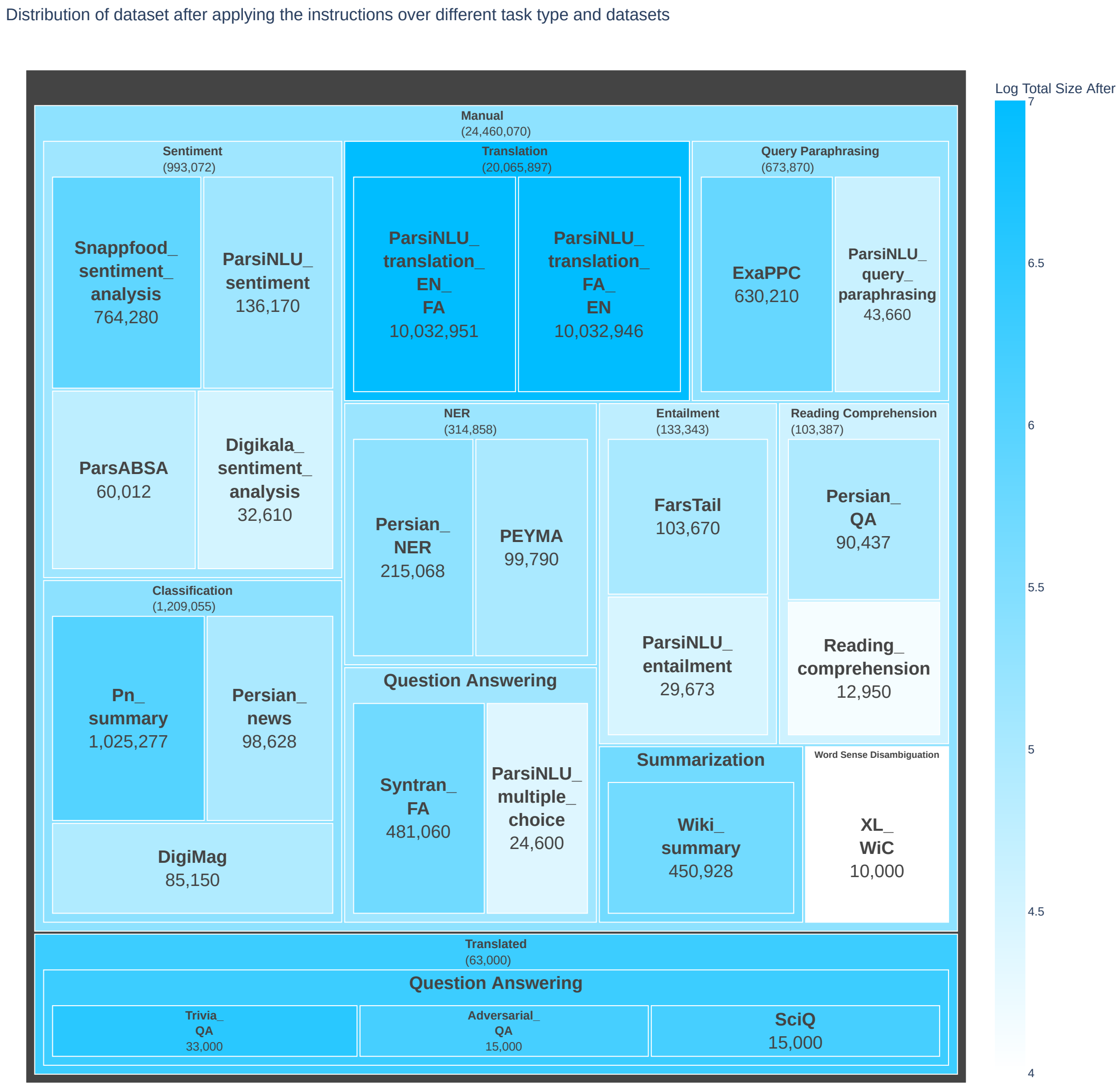

Figure 6: A treemap visualization that organizes datasets by task type, post-instruction application size, and data category (training vs. testing). Each primary rectangle represents a distinct task type within the natural language processing field, encompassing areas such as Question Answering, Classification, Translation, and more. Within these primary rectangles, smaller sub-rectangles represent individual datasets. The area of each sub-rectangle is scaled to the logarithm of the size of the dataset to accommodate the broad variance in dataset sizes, ensuring a more balanced visual representation that allows for the inclusion of both large and small datasets on the same scale.

D Prompts

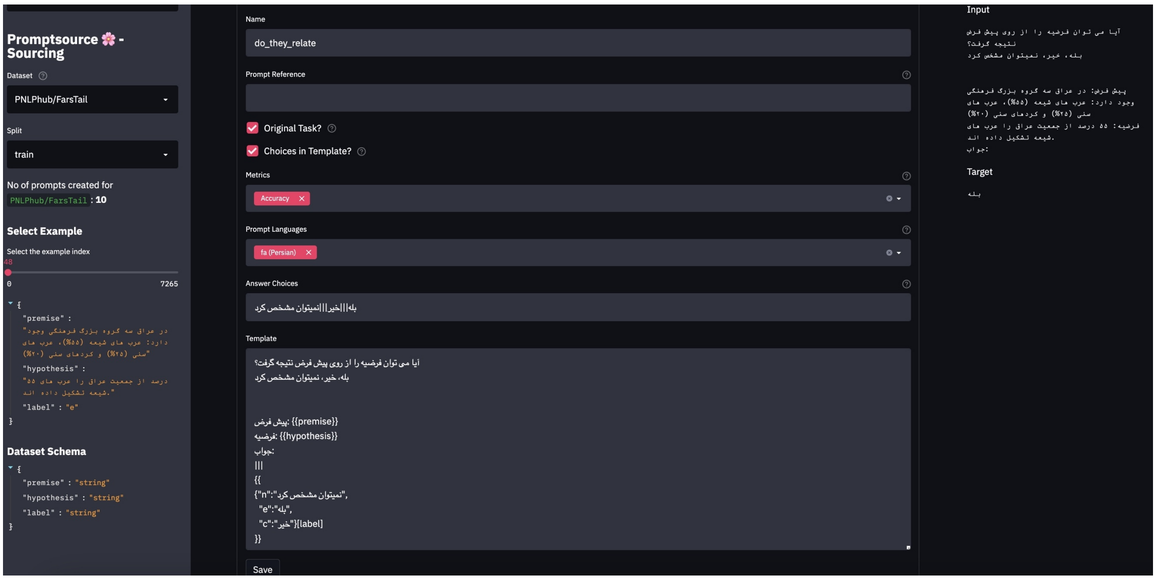

Figure 7: An example shown in the PromptSource environment. PromptSource is an advanced toolkit designed for creating, sharing, and utilizing natural language prompts. Prompt templates function as mappings that convert examples from datasets into natural language inputs and corresponding target outputs. In PromptSource, we develop input templates, target templates, and choice templates. Inputs typically consist of questions or instructions, while the output code specifies the expected answer or result. For categorization tasks, the choice template includes predefined options for answering questions, while generation tasks do not require this template. In this picture, The "Metrics" box is set to measure Accuracy for categorization tasks, and the "Prompt Language" used is Farsi (Persian). "Answer choices" are provided within the template, which comprises an instruction followed by data fields. The premise and hypothesis are selected from the "Data Schema" on the left side of the interface. The ||| symbol separates instructions from outputs, and the output employs Jinja code for conditional logic: if the label is c, it outputs (no); if the label is e, it outputs (yes); and if the label is n, it outputs (cannot determine).

Dataset: persiannlp/parsinlu_entailment
1. GPT3_Style
Input Template:

```
انتخاب کن که جمله اول و جمله دوم نسبت به هم چه نوع ارتباطی دارند؟ ارتباط منطقی وجود ندارد، مرتبط، نامرتبط


جمله اول: {{sent1}}
جمله دوم: {{sent2}}
جواب:
```

Target Template:

```
{{ {"c": "نامرتبط", "e": "مرتبط", "n": "ارتباط منطقی وجود ندارد" } [label] }}
```

Answer Choices Template:

```
مرتبط|||نامرتبط|||ارتباط منطقی وجود ندارد
```

---

2. based_on_the_previous_passage
Input Template:

```
با توجه به متن داده شده آیا میتوان عبارت را نتیجه گرفت؟
- بله
- خیر
- شاید


متن: {{sent1}}
عبارت: {{sent2}}
جواب :
```

Target Template:

```
{{ {"c": "خیر", "e": "بله", "n": "شاید" } [label] }}
```

Answer Choices Template:

```
بله|||خیر|||شاید
```

---

3. can_you_infer

Input Template:

```
تصور کن که عبارت اول داده شده است. آیا براساس آن میتوان عبارت دوم را استنتاج کرد؟ از بین گزینه های داده شده انتخاب کن
- اره
- نه
- شاید


عبارت اول: {{sent1}}
عبارت دوم: {{sent2}}
جواب:
```

Target Template:

```
{{ {"n": "شاید", "c": "نه", "e": "اره" } [label] }}
```

Answer Choices Template:

```
اره|||نه|||شاید
```

---

4. claim_relation
Input Template:

```
رابطه ی بین دو ادعای داده شده را تعیین کن: (مرتبط هست، نامشخص، مرتبط نیست)


ادعای اول: {{sent1}}
ادعای دوم: {{sent2}}
جواب:
```

Target Template:

```
{{ {"n": "نامشخص", "e": "مرتبط هست", "c": "مرتبط نیست" } [label] }}
```

Answer Choices Template:

```
نامشخص|||مرتبط هست|||مرتبط نیست
```

---

5. classify
Input Template:

نوع ارتباط این دو عبارت را در یکی از سه کلاس زیر دسته‌بندی کن
- کلاس دلالت: با توجه به عبارت مقدم، عبارت تالی درست می‌باشد
- کلاس تضاد: با توجه به عبارت مقدم، عبارت تالی غلط می‌باشد
- کلاس خنثی: با توجه به عبارت مقدم، نمی‌توان درباره‌ی درست یا غلط بودن تالی نظر قطعی داد

عبارت مقدم: {{sent1}}
عبارت تالی: {{sent2}}
جواب:

Target Template:

```
{{ {"n": "کلاس خنثی", "c": "کلاس تضاد", "e": "کلاس دلالت" } [label] }}
```

Answer Choices Template:

```
کلاس خنثی|||کلاس دلالت|||کلاس تضاد
```

---

6. comparison
Input Template:

با مقایسه بین پیش گزاره اول (فرض مقدماتی) و پیش گزاره دوم (پیش گزاره) چه نتیجه ای میگیرید؟

پیش گزاره اول: {{sent1}}
پیش گزاره دوم: {{sent2}}
نتیجه:

Target Template:

```
{{ {"n": "نامعلوم", "e": "هر دو پیش گزاره مشابه هستند", "c": "پیش گزاره ها متفاوت هستند" } [label] }}
```

---

7. classify
Input Template:

```
سطح اطمینان خود را در شباهت عبارات ارائه شده بیان کنید
- نامطمئن
- اطمینان پایین
- اطمینان بالا


عبارت اول: {{sent1}}
عبارت دوم: {{sent2}}
جواب:
```

Target Template:

```
{{ {"n": "نامطمئن", "c": "اطمینان پایین", "e": "اطمینان بالا" } [label] }}
```

Answer Choices Template:

```
نامطمئن|||اطمینان پایین|||اطمینان بالا
```

---

8. does_this_imply
Input Template:

```
آیا متن دوم میتواند مفهوم متن اول باشد؟ از بین گزینه های روبرو انتخاب کن
- بله
- خیر
- شاید


متن اول: {{sent1}}
متن دوم: {{sent2}}
جواب:
```

Target Template:

```
{{ {"c": "خیر", "e": "بله", "n": "شاید" } [label] }}
```

Answer Choices Template:

```
بله|||خیر|||شاید
```

---

9. evaluate
Input Template:

دو نظریه از دو منبع اطلاعاتی مختلف بیان شده اند. ارتباط بین آنها در کدام ارزیابی قرار دارد؟
الف) بسیار مرتبط
ب) نامرتبط
ج) نامشخص

نظریه اول: {{sent1}}
نظریه دوم: {{sent2}}
جواب:

Target Template:

{{ {"n": "ج", "c": "ب", "e": "الف" } [label] }}

Answer Choices Template:

ج|||ب|||الف

10. gen_sent
Input Template:

باتوجه به جمله ی زیر یک جمله بنویس به گونه ای که نوع ارتباطشان به صورت زیر باشد

نوع ارتباط: {{{"n":"نامشخص", "e":"مرتبط", "c":"نامرتبط"}[label]}}
جمله: {{sent1}}
جواب:

Target Template:

{{sent2}}

Dataset: PNLPhubsnappfoodsentimentanalysis
1. comment
Input Template:

با در نظر گرفتن دیدگاه کلی مشتریان نسبت به این محصول، آیا از خریدشان راضی بودند یا نه؟

دیدگاه: {{comment}}
جواب:

Target Template:

{% if label_id == 0%}
مشتری از خریدش راضی بود
{% else %}
مشتری از خریدش راضی نبود
{% endif %}

---

2. feelings
Input Template:

با در نظر گرفتن کامنت خریدار، این محصول مشتری را خوشحال یا ناامید کرده است؟

دیدگاه: {{comment}}
جواب:

Target Template:

{% if label == "HAPPY"%}
این خرید مشتری را خوشحال کرده است
{% else %}
این خرید مشتری را ناامید کرده است
{% endif %}

---

3. gen_sentiment
Input Template:

```
عبارت ارائه شده را با دقت مطالعه کن و تصمیم بگیر که محتوای آن براساس برچسب داده شده چه حسی را منتقل میکند؟

برچسب: {{label}}
عبارت: {{comment}}
احساس:
```

Target Template:

```
{% if label == "SAD"%}
ناراحت
{% else %}
خوشحال
{% endif %}
```

4. is_it_neg
Input Template:

```
آیا محتوای داده شده حس منفی یا بد را به خواننده منتقل میکند؟ ارزیابی باید دقیق و براساس نحوه بیان متن باشد

متن: {{comment}}
جواب:
```

Target Template:

```
{% if label_id == 1%}
بله
{% else %}
خیر
{% endif %}
```

5. is_it_pos
Input Template:

```
آیا متن ارائه شده دارای بار احساسی مثبت است؟

متن: {{comment}}
جواب:
```

Target Template:

```
{% if label_id == 0%}
بله
{% else %}
خیر
{% endif %}
```

6. possibility
Input Template:

```
نظر مشتری را نسبت به جنبه های مختلف کالایی که خریداری کرده، بسنجید و تصمیم بگیرید که آیا احتمال دارد که مجدد این محصول را خریداری کند؟

نظر: {{comment}}
جواب:
```

Target Template:

```
{% if label_id == 0%}
احتمال اینکه مجدد این محصول را خریداری کند زیاد است
{% else %}
احتمال اینکه مجدد این محصول را خریداری کند کم است
{% endif %}
```

7. rate
Input Template:

فرم نظرسنجی از مشتری دریافت شده است و به صورت زیر میباشد. چه امتیازی به آن میدهید؟
- پنج ستاره
- یک ستاره

فرم نظرسنجی: {{comment}}
امتیاز:

Target Template:

```
{% if label == ”HAPPY”%}
پنج ستاره
{% else %}
یک ستاره
{% endif %}
```

Answer Choices Template:

```
یک ستاره|||پنج ستاره
```

8. what_is_sentiment
Input Template:

کاربری پس از خرید یک محصول نظر زیر را در مورد آن دارد. بررسی کن که آیا او از خریدش خوشحال است یا ناراحت؟

نظر: {{comment}}
جواب:

Target Template:

```
{{ {”SAD”: ”ناراحت”, ”HAPPY”: ”خوشحال”} [label] }}
```

Answer Choices Template:

```
خوشحال|||ناراحت
```

0.1 Prompts (Translated to english)

Dataset: persiannlpparsinlu_entailment

1. GPT3_Style
Input Template:

```
Choose what kind of relationship exists between the first and second sentence? No logical connection, Related, Unrelated


First sentence: {{sent1}}
Second sentence: {{sent2}}
Answer:
```

Target Template:

```
{{ {”c”: ,”Unrelated” ”e”: ,”Related” ”n”: ”No logical connection” } [label] }}
```

Answer Choices Template:

```
Related|||Unrelated|||No logical connection
```

---

2. based_on_the_previous_passage
Input Template:

```
Based on the given text, can the statement be concluded?
- Yes
- No
- Maybe


Text: {{sent1}}
Statement: {{sent2}}
Answer :
```

Target Template:

```
{{ {”c”: ,”No” ”e”: ,”Yes” ”n”: ”Maybe” } [label] }}
```

Answer Choices Template:

Yes|||No|||Maybe

3. can_you_infer
Input Template:

Imagine the first statement is given. Based on that, can the second statement be inferred? Choose from the given options
- Yes
- No
- Maybe

First Statement: {{sent1}}
Second Statement: {{sent2}}
Answer:

Target Template:

{{ {"n": ,"Maybe" "c": ,"No" "e": "Yes" } [label] }}

Answer Choices Template:

Yes|||No|||Maybe

4. claim_relation
Input Template:

Determine the relationship between the two given claims: (Related, Uncertain, Unrelated)

First Claim: {{sent1}}
Second Claim: {{sent2}}
Answer:

Target Template:

{{ {"n": ,"Uncertain" "e": ,"Related" "c": "Unrelated" } [label] }}

Answer Choices Template:

Uncertain|||Related|||Unrelated

5. classify
Input Template:

```
Classify the relationship between these two statements into one of the three categories below
- Implication class: Considering the premise, the subsequent statement is correct
- Contradiction class: Considering the premise, the subsequent statement is incorrect
- Neutral class: Considering the premise, it’s not possible to definitively state whether the subsequent statement is correct or incorrect


Premise: {{sent1}}
Subsequent statement: {{sent2}}
Answer:
```

Target Template:

```
{{ {”n”: ,”Neutral class” ”c”: ,”Contradiction class” ”e”: ,”Implication class” } [label] }}
```

Answer Choices Template:

```
Neutral class|||Implication class|||Contradiction class
```

6. comparison
Input Template:

```
By comparing the first premise (preliminary assumption) and the second premise, what conclusion do you draw?


First premise: {{sent1}}
Second premise: {{sent2}}
Result:
```

Target Template:

```
{{ {”n”: ,”Unknown” ”e”: ,”Both premises are similar” ”c”: ”The premises are different” } [label] }}
```

7. classify
Input Template:

```
Express your confidence level in the similarity of the given statements
- Uncertain
- Low confidence
- High confidence


First statement: {{sent1}}
Second statement: {{sent2}}
Answer:
```

Target Template:

```
{{ {"n": ,"Uncertain" "c": ,"Low confidence" "e": ,"High confidence" } [label] }}
```

Answer Choices Template:

```
Uncertain|||Low confidence|||High confidence
```

---

8. does_this_imply
Input Template:

```
Can the second text be the meaning of the first text? Choose from the options
- Yes
- No
- Maybe


First text: {{sent1}}
Second text: {{sent2}}
Answer:
```

Target Template:

```
{{ {"c": ,"No" "e": ,"Yes" "n": ,"Maybe" } [label] }}
```

Answer Choices Template:

```
Yes|||No|||Maybe
```

9. evaluate
Input Template:

```
Two theories from different information sources are stated. In which evaluation do their relationships belong?
a) Highly related
b) Unrelated
c) Uncertain


First theory: {{sent1}}
Second theory: {{sent2}}
Answer:
```

Target Template:

```
{{ {"n": ,"Uncertain" "c": ,"Unrelated" "e": ,"Highly related" } [label] }}
```

Answer Choices Template:

```
Uncertain|||Unrelated|||Highly related
```

10. gen_sent
Input Template:

```
Considering the sentence below, write a sentence such that their relationship is as follows


Relationship type: {{{"n":"Uncertain", "e":"Related", "c":"Unrelated"}[label]}}
Sentence: {{sent1}}
Answer:
```

Target Template:

```
{{sent2}}
```

Dataset: PNLPhub/snappfood-sentiment-analysis
1. comment
Input Template:

Considering the overall customer perspective towards this product, were they satisfied with their purchase?

Perspective: {{comment}}
Answer:

Target Template:

{% if label$_i d == 0\%$}
The customer was satisfied with their purchase
{% else %}
The customer was not satisfied with their purchase
{% endif %}

2. feelings
Input Template:

Considering the buyer's comment, did this product make the customer happy or disappointed?

Perspective: {{comment}}
Answer:

Target Template:

{% if label == "HAPPY"%}
This purchase made the customer happy
{% else %}
This purchase disappointed the customer
{% endif %}

3. gen sentiment
Input Template:

Carefully read the provided statement and decide what emotion it conveys based on the given label.

Label: {{label}}
Statement: {{comment}}
Emotion:

Target Template:

{% if label == ”SAD”%}
Sad
{% else %}
Happy
{% endif %}

4. is it neg
Input Template:

Does the given content convey a negative or bad feeling to the reader? The evaluation should be precise and based on the way the text is expressed.

Text: {{comment}}
Answer:

Target Template:

{% if label$_i d == 1\%$}
Yes
{% else %}
No
{% endif %}

5. is it pos
Input Template:

Does the presented text have a positive emotional charge?

Text: {{comment}}
Answer:

Target Template:

```
{% if label_id == 0%}
Yes
{% else %}
No
{% endif %}
```

6. possibility
Input Template:

```
Assess the customer's opinion on various aspects of the product they purchased and decide whether there is a likelihood of repurchasing it?


Opinion: {{comment}}
Answer:
```

Target Template:

```
{% if label_id == 0%}
The likelihood of repurchasing this product is high
{% else %}
The likelihood of repurchasing this product is low
{% endif %}
```

7. rate
Input Template:

```
A customer feedback form has been received as follows. What rating would you give it?
- Five stars
- One star


Feedback form: {{comment}}
Rating:
```

Target Template:

```
{% if label == ”HAPPY”%}
Five stars
{% else %}
One star
{% endif %}
```

Answer Choices Template:

```
One star|||Five stars
```

---

8. what is sentiment
Input Template:

```
A user has the following opinion about a product they purchased. Determine whether they are happy or sad about their purchase.


Opinion: {{comment}}
Answer:
```

Target Template:

```
{{ {”SAD”: ,”Sad” ”HAPPY”: ”Happy” [label] }}
```

Answer Choices Template:

```
Happy|||Sad
```