FarsInstruct: Empowering Large Language Models for Persian Instruction Understanding

Anonymous ACL submission

Abstract

Instruction-tuned large language models, such as T0, have demonstrated remarkable capabilities in following instructions across various domains. However, their proficiency remains notably deficient in many low-resource languages. To address this challenge, we introduce 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 Public Pool of Prompts, ensuring a rich linguistic and cultural representation. Furthermore, we introduce the Co-CoLA, a framework to enhance the multi-task adaptability of LoRA-tuned models. Through extensive experimental analyses, our study showcases the effectiveness of Farsinstruct dataset coupled with training by Co-CoLA framework, in improving the performance of large language models within the Persian context. As of the current writing, FarsInstruct comprises more than 200 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). These models, such as PaLM (Chowdhery et al., 2022), GPT4 (OpenAI et al., 2024), and Llama2 (Touvron et al., 2023) with continuous scaling of their parameters and training data, are known to exhibit emergent properties. Wei et al. (2022a) considers an ability to be emergent if it is not present in smaller models but is present in larger models. This is an unpredictable phenomenon that can not be predicted simply by extrapolating the performance of smaller models. One such ability is instruction-following, which enables models to execute unseen natural language processing (NLP) tasks from reading instructions provided within the input text. Previously, the capability for instruction-following was primarily attributed to the scale of these models. However, recent studies have demonstrated that instruction-following does not exclusively rely on the large size of language models (Wei et al., 2021). By instruction-tuning on a collection of instructional NLP tasks, smaller language models can learn to follow prompts. This approach has proven to be remarkably efficient, allowing these smaller models to perform competitively and, in some specific tasks, even outperform their larger counterparts (Sanh et al., 2022; Wei et al., 2021; Wang 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, offering 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, they still struggle to accurately grasp the subtleties of low-resourced languages due to the scarcity of prompted data and challenges inherent in translating English datasets into other languages (Naous et al., 2024; Ramesh et al., 2023; Vanmassenhove et al., 2021). While efforts have been made to compile extensive multilingual instruction-following datasets, gaps remain in creating diverse and complex prompts for languages like Persian. For example, the SuperNaturalInstructions benchmark (Wang et al., 2022), encompassing various task types across 55 languages, contains

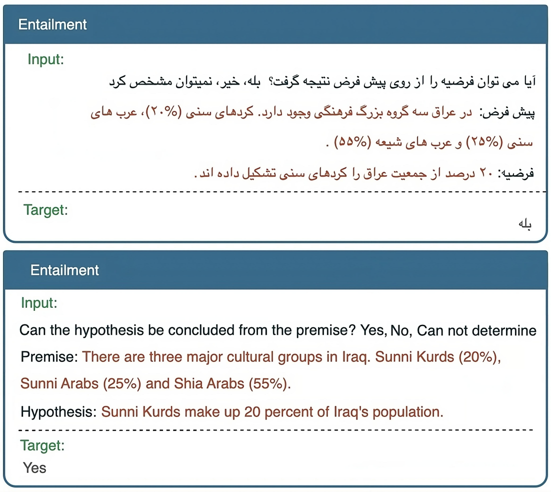

Figure 1: An example of the prompts utilized in the training process. The Persian version of the prompt is employed for training purposes, while the translated English version is provided to enhance comprehension. The instruction component is highlighted in blue, the data field is marked in orange, and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

merely 2.1% of Persian content. Similarly, the Aya Dataset (Singh et al., 2024), a human-curated effort to enhance AI's instruction-following abilities across 65 languages, includes 1% of Persian content. This underscores the disparity in the diversity and quantity of the Persian Language tasks compared to other languages.

In this study, we propose FarsInstruct, a comprehensive prompted dataset tailored to the Persian language. It comprises a mixture of manually written instructions ranging from basic to proficient language levels, as well as translations from Public Pool of Prompts (P3) (Sanh et al., 2022) which is a collection of prompted English datasets. In particular, we created more than 200 prompt templates (roughly 10 templates for each of the 21 unique public datasets) that we selected from a variety of sources. These datasets collectively cover 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, and a detailed overview of FarsInsturct dataset is provided in Section 3.

Additionally, in order to facilitate the multi-task adaptation of our model and mitigate the problem of catastrophic forgetting, we introduce Co-CoLA, an 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 involve retraining a subset of data from previously learned tasks and mixing it with the current task's data during training. With this periodic revisiting of earlier tasks, the model maintains performance on both old and new tasks while preserving computational efficiency. Section 4 presents an in-depth explanation of Co-CoLA method.

FarsInstruct is publicly available and open-source and we are committed to enhancing it by continually expanding our dataset with a broader range of tasks, instruction entries, and modalities. We hope this dataset fills the critical gap and serves as a valuable resource to the NLP community.

2 Related work

Instruction-tuning. In the landscape of AI, the capabilities of LLMs have expanded far beyond mere text processing. These sophisticated models are now being fine-tuned in a practice known as instruction-tuning, where models are trained with specific input-output pairs drawn from a wide array of data sources. This technique enables a pre-trained LLM to produce tailored outputs based on given inputs, enhancing its versatility and effectiveness. FLAN (Wei et al., 2021) and T0 (Sanh et al., 2022) pioneered the exploration of instruction-tuned language models, each contributing significantly to the field. FLAN (Wei et al., 2021) adapted a 137-billion parameter pre-trained model, refining it with over 60 NLP datasets using natural language instructions. On the other hand, T0 (Sanh et al., 2022) applied instruction tuning to various T5 models across 2073 prompts from 177 datasets. SuperNaturalInstruction (Wang et al., 2022) further advanced the field by assembling a comprehensive benchmark 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 (Sanh et al., 2022) by including content from 46 languages, adding new tasks like Translation and Program Synthesis that P3 had not tackled. In a similar expansive effort, Aya (Singh et al., 2024) emerged as a significant multilingual project, featuring an impressive collection of 513 million instances across 114 languages, achieved through a collaborative research effort that involved fluent speakers from around the world to compile and complete instructional content. Our dataset distinguishes itself from these collections in its depth and adaptability, especially with the inclusion of more challenging Persian tasks, 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 the model size and the number of downstream tasks increase. To address this challenge, recent advancements in parameter-efficient fine-tuning methods suggest training only a small portion of parameters while keeping the majority of pre-trained model parameters unchanged. One of the most widely used paradigms in parameter-efficient fine turning is Low-Rank Adaptation (LoRA) (Hu et al., 2021). LoRA only modifies a small, low-rank portion of the model's weights. This is achieved by adding low-rank matrices to the model's weights during training. Despite the significant computational advantage of LoRA, it falls short in multi-task adaptation, and also Kalajdzievski (2024) showed that PEFT strategies, such as LoRA, are still susceptible to catastrophic forgetting. MultiLoRA (Wang et al., 2023) 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. Additionally, 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 designed Co-CoLA to address the issue of catastrophic forgetting, while ensuring an effective multi-task adaption.

3 FarsInstruct Dataset

With about 130 million[1] 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 prompted dataset for the Persian community. As of this writing, FarsInstruct has more than 200 carefully designed and created prompt templates for 21 already-published public datasets and some translated prompted datasets. Unlike multilingual collections focusing on common tasks such as Text Summarization and Question Answering, FarsInstruct introduces more innovative and challenging tasks, 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.

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 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 as a

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

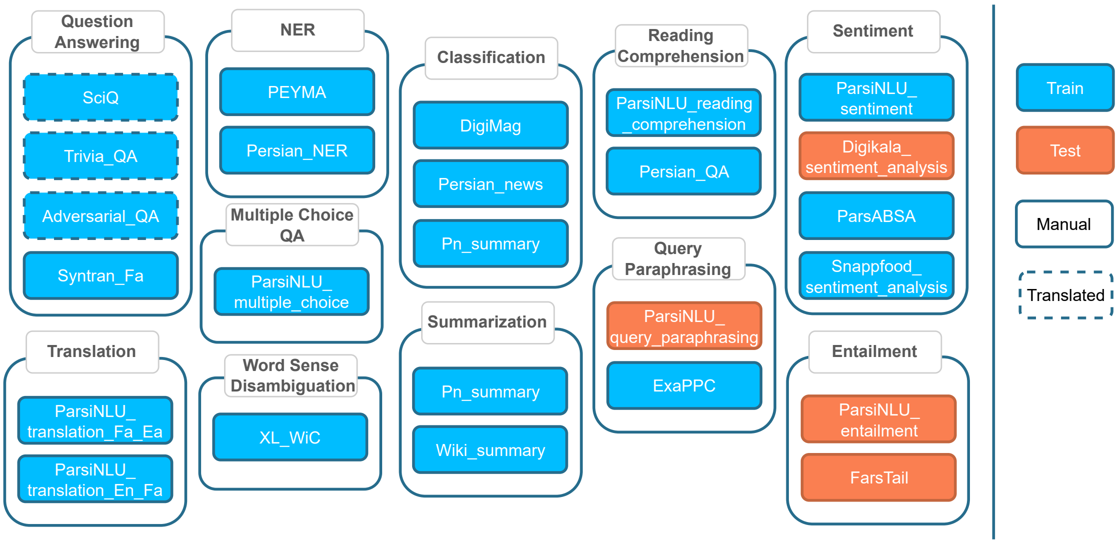

Figure 2: Detailed depiction of 11 task types utilized in our dataset. Each box within the figure lists the specific datasets associated with the respective task type. Datasets designated for training are highlighted in blue, and those reserved for testing are marked in orange. Additionally, manual datasets, 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.

templating language, providing the flexibility crucial for crafting clear and effective prompts. Specifically, a template is a function that maps dataset examples into input-output natural language pairs, while a prompt is the combination of an input template and a target template along with a collection of specific meta-data. Instructions are specific directives within input-templates that guide the model's behavior. Appendix C provides some exmaples of prompt template. However, as the original version of PromptSource did not support Persian, we modified its source code to accommodate Persian datasets. Since this system is originally integrated with Huggingface Datasets (Lhoest et al., 2021) library, we gathered datasets from various sources and consolidated into a unified public repository on HuggingFace

In addition to manual templating, we have decided to translate a subset of three question-answering tasks from the P3 dataset (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.

Finally, since the datasets were sourced from multiple repositories, we applied a series of preprocessing steps such as deduplication and stripping out non-alphanumeric characters like emojis to ensure normalized text across all data.

3.2 Task Augmentation

It is widely recognized that instruction-tuned models benefit significantly from extensive and varied tasks. Given this context, we focus on developing diverse prompts, spanning from basic to proficient language levels. Furthermore, drawing from the methodologies outlined in FLAN Collection (Longpre et al., 2023), T0 (Sanh et al., 2022), and MetaICL (Min et al., 2022), we enhance task diversity by mixing and swapping different data fields within a given dataset. For instance, whereas a dataset might initially be structured to evaluate a model's ability to answer question x given input y, we train the model to generate question x when provided with answer y. This approach effectively broadens the spectrum of prompts within a limited data pool.

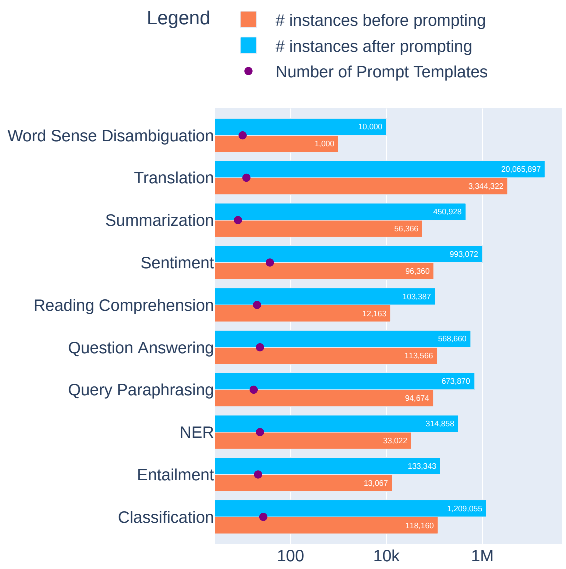

Figure 3: Distribution of NLP tasks across the FarsInstruct dataset, highlighting the expanded data volumes post-prompt application and the number of prompts designed per task type.

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.

3.3 Data statistics

The statistics of final dataset after applying templates across all datasets is presented in Figure 3. Table 1 also presents the total number of categorization and generation prompts for each task type.

3.4 Quality Control

We selectively chose all publicly available Persian datasets, predominantly used for single-task fine-tuning, as their extensive use ensures high quality. Furthermore, to ensure the accuracy and quality of the instructions, we conduct human evaluations through consultations with the general public and experts in the field of literature. This review process allowed us to assess the instructions from multiple perspectives and incorporate cultural and linguistic nuances, critical for ensuring the prompts' clarity, accuracy, and relevance.

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 highlighted in studies by (Wang et al., 2023; 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 optimization framework based on the principles of the Frank-Wolfe algorithm (also known as the Conditional Gradient Method). This method involves an iterative process of fine-tuning on a single task, merging it with the base model, and reinitializing with a new LoRA module. Xia et al. (2024) shows that 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) observed a strong inverse linear relationship between the fine-tuning performance and the amount of forgetting when fine-tuning LLMs with LoRA.

In this study we propose Continual-Chain of LoRA (Co-CoLA), an extension of CoLA framework which 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 during training new tasks. 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 instruction via instruction composition, resulting in better generalization and improved performance on 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 this training procedure into an iterative three phases:

Tuning: In this phase, following the standard LoRA, the base model weights remain frozen, while only the model's LoRA parameters (represented by matrices A and B) are fine-tuned. Additionally, a subset of previously trained data is replayed along with the new data. Formally, given the sequence $T = (T_1, \ldots, T_n)$ where T_i represents the training data after applying an individual template, 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 hyper-parameter that controls the percentage of examples sampled from previous templates $T_1, \ldots, T_n$.

Merging: After the tuning phase, the newly updated LoRA parameters are merged with the existing model weights. 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

Evaluation Tasks: Our model's performance was evaluated through two categories of task types: those that were part of the training dataset ("Held in") and those introduced to the model for the first time during evaluation ("Held out"). The evaluation dataset encompasses three distinct types of

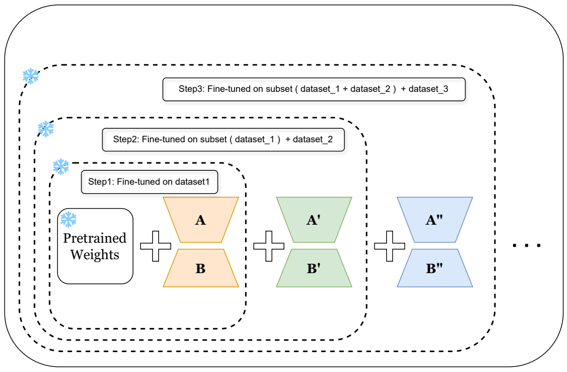

Figure 4: The Continual-Chain of LoRA Training Procedure

tasks: Sentiment analysis and Query paraphrasing, classified as "Held in" tasks and Textual Entailment which is categorized as a "Held out" task. As illustrated in Figure 2, the evaluation includes one dataset each for sentiment analysis and paraphrase identification, alongside two datasets dedicated to entailment tasks.

Evaluation Metric: To assess the performance of our model relative to several baseline models, we utilized the ROUGE-L metric, which measures the overlap of n-grams between the generated text and reference texts. Specifically, we concentrated on the F1-scores of ROUGE-L, a metric that integrates precision and recall to provide a balanced evaluation. As demonstrated by (Wang et al., 2022), the rankings produced by this metric exhibit a strong correlation with accuracy for categorization templates.

5 Results

To investigate the applicability of FarsInstruct, we choose the Ava model and instruction-tune it using the Co-CoLA framework on a diverse set of templates. Our results were compared against a series of mono-lingual and multi-lingual instruction-tuned models and to effectively assess the performance of our model we conduct both quantitative and linguistic 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-specific data. Specifically, PersianMind (University of Tehran, 2024) is a Llama-2 7B based model, trained in 3 phases on different Persian datasets. Though

Task	Type	Model	ROUGE-L
parsinlu query paraphrasing	Held In	Aya-13B	45.58
		PersianMind-7B	17.07
		Mistral-7B	6.89
		Dorna-8B	3.85
		Ava-8B	6.67
		Ava-LoRA-8B	8.73
		Co-CoLA-8B	**45.86**
Digikala Sentiment Analysis	Held In	Aya-13B	28.41
		PersianMind-7B	18.19
		Mistral-7B	2.46
		Dorna-8B	2.42
		Ava-8B	8.69
		Ava-LoRA-8B	5.72
		Co-CoLA-8B	**40.87**
FarsTail	Held Out	Aya-13B	**37.61**
		PersianMind-7B	17.05
		Mistral-7B	5.74
		Dorna-8B	4.81
		Ava-8B	12.48
		Ava-LoRA-8B	9.07
		Co-CoLA-8B	36.35
Parsinlu Entailment	Held Out	Aya-13B	42.64
		PersianMind-7B	4.45
		Mistral-7B	4.93
		Dorna-8B	3.32
		Ava-8B	15.04
		Ava-LoRA-8B	7.18
		Co-CoLA-8B	**55.32**

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

their training data is unavailable, Dorna (PartAI, 2024) and Ava (Moghadam, 2024) are newly introduced models, 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, despite 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, are inferior to those achieved with Co-CoLA training, highlighting the effectiveness of our method.

5.2 Linguistic Evaluation

Our comprehensive linguistic evaluation aimed to further substantiate the effectiveness of Co-CoLA in handling the nuances of the Persian language, compared to the baseline model Ava. The evaluation specifically focused on analyzing the models' capabilities in terms of coherence, relevance, and linguistic quality, which are critical for assessing the applicability of language models in real-world scenarios.

5.2.1 Evaluation Setup

The evaluation involved detailed analysis by language experts who assessed the output from both models based on predefined criteria. This approach ensures an unbiased evaluation of the models' performance in generating contextually appropriate and linguistically accurate content.

5.2.2 Evaluation Criteria

The linguistic outputs were evaluated based on three main criteria:

Coherence: This assesses the logical flow and connectivity of the text produced by the models.

Relevance: This measures how well the model's output adheres to the context provided in the input.

Linguistic Quality: This evaluates the grammatical accuracy, punctuation, and stylistic appropriateness of the text.

5.2.3 Evaluation Results

The evaluation results are summarized in Table 3, which provides a clear comparative analysis of the performance of the two models across all assessed criteria. The scores indicate that while Ava scored slightly higher in coherence, Co-CoLA outperformed Ava in relevance and linguistic quality, suggesting its superior ability to produce contextually accurate and linguistically refined outputs.

The higher scores of Co-CoLA in relevance and linguistic quality demonstrate its effectiveness in

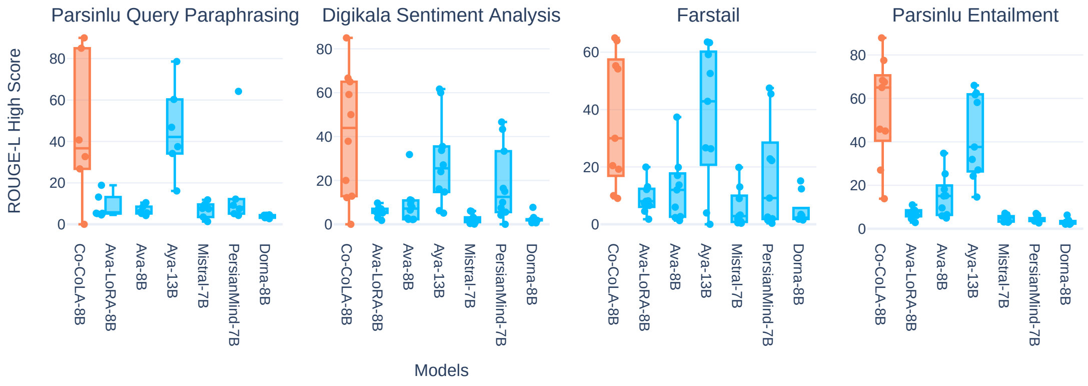

Figure 5: Comparative performance of different models on Persian language tasks using the ROUGE-L metric. The bar chart depicts the superior performance of *Co-CoLA* across multiple tasks, particularly excelling in the Parsinlu Entailment task.

Criteria	**Co-CoLA**	**Ava**
Coherence	4.2	**4.3**
Relevance	**3.7**	3.2
Linguistic Quality	**4.6**	4.0

Table 3: Average Scores from Linguistic Evaluation

producing not only grammatically correct but also contextually relevant outputs, which is essential for real-world applications. These results underscore the potential of Co-CoLA in enhancing the linguistic handling of Persian language tasks, setting a benchmark for future developments in language model applications.

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, which combines precision and recall, 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.

7 Conclusion

This study introduces significant advancements with FarsInstruct and Co-CoLA, addressing critical gaps in the processing and instruction-following capabilities for Persian, a low-resource language. FarsInstruct, with its diverse tasks ranging from text summarization to named entity recognition, has proven to enhance language model performance as shown through rigorous ROUGE evaluations and human assessments. This dataset not only enriches multi-lingual model training but also establishes a new standard for language model instruction tuning.

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. This allows for sustained model performance over diverse tasks while optimizing computational resources. Looking ahead, the focus will be on expanding the scope of these datasets to cover more tasks and modalities, thereby driving further innovations in cross-lingual language understanding and promoting AI inclusivity.

8 Limitations

This section delineates the principal limitations of our study, which, while providing substantial contributions to Persian NLP, presents challenges that could be addressed in future developments to 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., 2022b).

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

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*.

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.

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*.

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`.

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*.

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

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.

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.

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.

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.

PartAI. 2024. Dorna-Llama3-8B-Instruct model on hugging face. https://huggingface.co/PartAI/Dorna-Llama3-8B-Instruct. Accessed: 2024-06-15.

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.

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*.

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*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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. 2023. Multilora: Democratizing lora for better multi-task learning. *arXiv preprint arXiv:2311.11501*.

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. 2022. 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, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. 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, Transformers (Wolf et al., 2020) and Accelerate (Gugger et al., 2022) library. For efficient training, we randomly selected 25 prompt templates and applied them to their corresponding datasets. Consequently, for example, a dataset with two selected templates would be upsampled to twice its original size. We then sampled a minimum of 10,000 examples from each dataset, based on the specific template and dataset length, to create the current training data. The rehearsal hyper-parameter of Co-CoLA was set to 0.01. 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. Dataset 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 4: Detailed Overview of Datasets Utilized for Categorization and Generation Tasks. Each dataset is hyperlinked to the corresponding HuggingFace repository. 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.

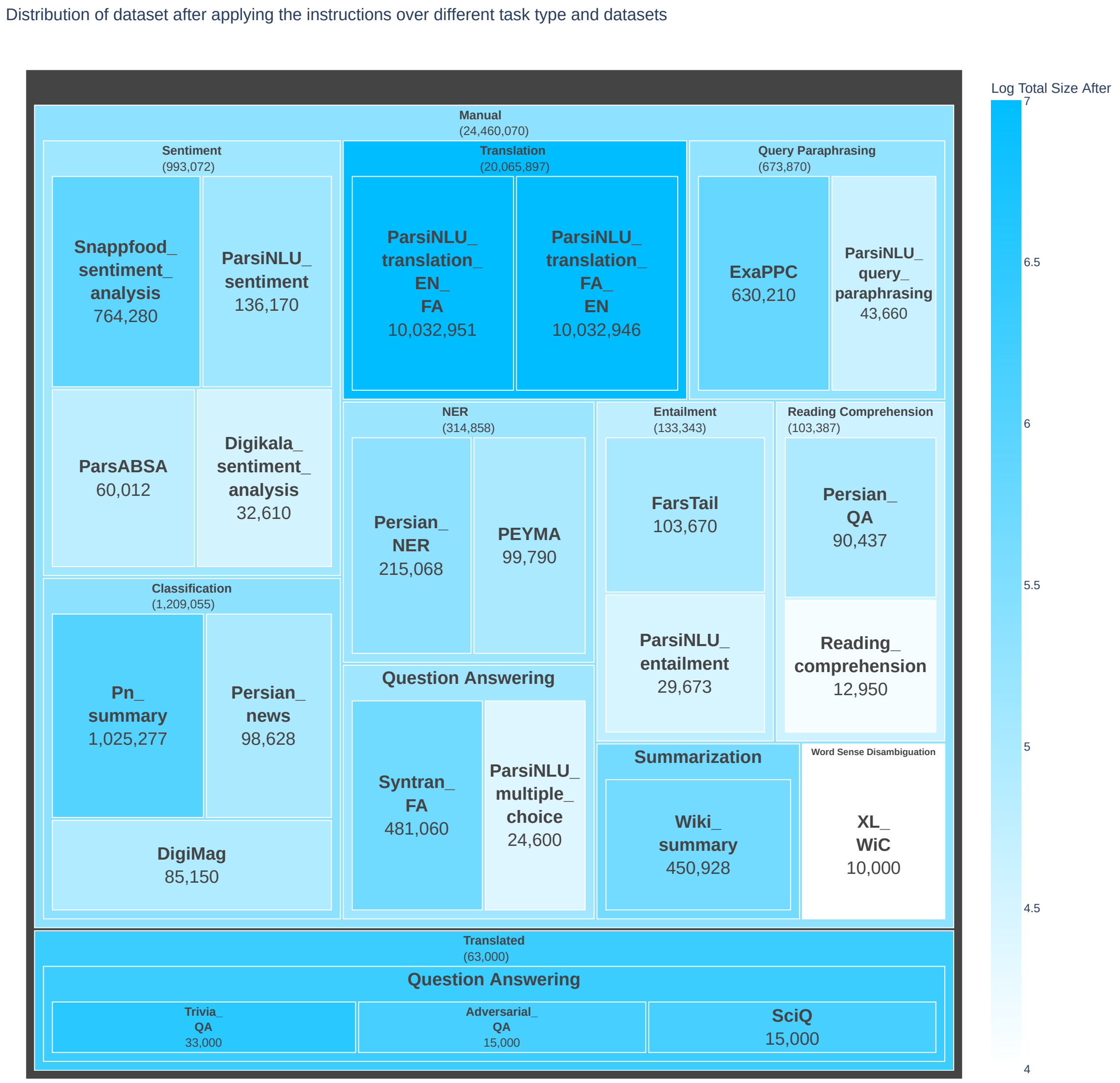

Figure 6: A treemap visualization that organizes datasets by task type, post-instruction application size, and data category (training vs. testing). Each primary rectangle represents a distinct task type within the natural language processing field, encompassing areas such as Question Answering, Classification, 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.

C. Prompts

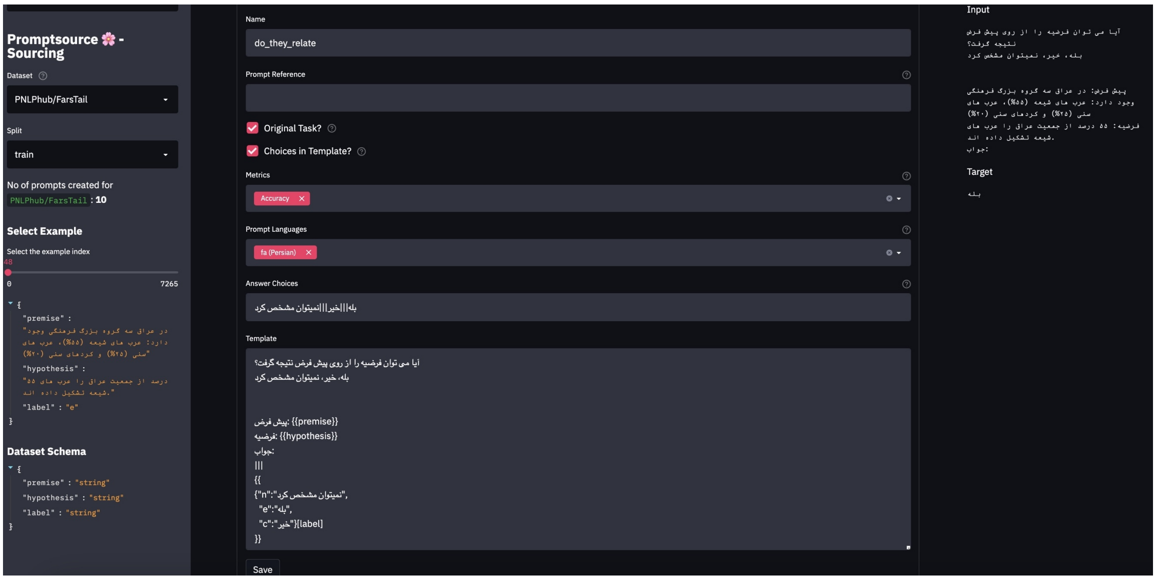

Figure 7: A prompt example shown in Promptsource environment. PromptSource is an advanced toolkit designed for creating, sharing, and utilizing natural language prompts. Prompts function as mappings that convert examples from datasets into natural language inputs and corresponding target outputs. In 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_id == 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$_i d == 0$%}
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$_i d == 0$%}
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
```