FlexTag: Instruction Tagging for Multi-Perspective Understanding with Small Language Models

Anonymous ACL submission

Abstract

Instruction tagging maps free-form user requests into structured metadata, serving as a core component for LLM routing, data auditing, and retrieval indexing. Existing tagging methods typically rely on flat semantic keyword generation, which struggles to generalize across template variations and fails to capture multi-perspective tag quality. In practice, high-quality tags should simultaneously exhibit diversity, complementarity, and synergy across semantic facets. We introduce FlexTag, a 1B-scale instruction tagging framework that supports both open-ended multi-perspective tag generation and tag selection. FlexTag addresses the limitations of prior approaches through three key components: (1) a multi-template data augmentation pipeline that improves robustness to prompt and template variations, (2) a multi-objective training strategy that optimizes multi-perspective tag quality, and (3) a template-adaptive inference pipeline that enhances tagging reliability. Experiments on four public datasets show that FlexTag outperforms all comparable same-scale baselines on open tag generation, surpasses GPT-4o zero-shot performance on tag selection, and maintains strong generalization under out-of-distribution settings. We release FLEXTAG at https://anonymous.4open.science/r/FlexTag-A0DD.

1 Introduction

In recent years, large language models (LLMs) such as GPT-4 (Achiam et al., 2023) have become core infrastructure for real-world systems, powering code generation, question answering, and writing assistance. A key enabling technique behind these applications is instruction tagging, which maps free-form natural language requests into structured metadata labels. Such tags support a broad range of downstream uses: LLM routing can select an appropriately sized model to balance quality and cost; data analysis can audit training distributions and reveal coverage gaps; and retrieval can build fine-grained indices that improve search and navigation.

Most prior work on tag generation (*eg.,* InsTagger (Lu et al., 2023), TAGROUTER (Chen et al., 2025)) targets open-domain keyword tagging: it leverages strong LLMs to produce arbitrary keyword tags and then distills this behavior into smaller models for large-scale analysis. TagGPT (Li et al., 2023) instead performs zero-shot tag generation with general-purpose LLMs, using frequency statistics and semantic matching to induce tag sets. In a different direction, TagLLM (Shen et al., 2024) treats tags as controllable inputs via learnable embeddings to steer model behavior, rather than generating human-readable, structured metadata.

While large models such as GPT-4O (Li et al., 2023) can perform instruction tagging via zero-shot prompting, production systems are highly sensitive to latency and cost, making 1B-scale small language models a more practical deployment choice. However, the limited capacity of small models amplifies three deeply interconnected limitations of existing approaches. First, existing methods are designed primarily for open-domain keyword tagging and cannot meet production demands for schema-constrained multi-facet annotations that directly drive routing, auditing, and retrieval pipelines. Second, even when output format is enforced, standard fine-tuning under low-resource conditions fails to guarantee multi-perspective quality: models collapse toward high-frequency label values, produce redundant predictions across facets, and fail to generate complementary evidence across dimensions. Third, small models trained under narrow task-specific supervision struggle to retain broad instruction-following competence, causing sharp degradation under minor changes in input format or output schema. Together, these challenges constitute the core obstacles to building high-quality

small-scale taggers, and must be addressed jointly across data, training, and inference. To this end, we propose FLEXTAG, which tackles them from three complementary angles: multi-template data augmentation, multi-objective training, and template-adaptive inference.

To address these three challenges, we propose FLEXTAG, a 1B-scale instruction tagging framework supporting both tag selection and open-ended multi-perspective tag generation, tackling them jointly from three complementary angles: *(1)* a low-resource multi-template data augmentation pipeline that rewrites each sample into four template categories to mitigate template rigidity; *(2)* a multi-objective training strategy with span-weighted cross-entropy, facet coverage, cross-facet redundancy, and synergy terms to ensure multi-perspective quality; and *(3)* a template-adaptive inference pipeline to guarantee output validity. Our main contributions are as follows:

- We propose a low-resource multi-template data augmentation pipeline that augument each sample into four instruction formats, systematically mitigating template rigidity for small-scale taggers.
- We introduce a multi-objective training strategy comprising span-weighted cross-entropy, facet coverage, cross-facet redundancy, and synergy, jointly ensuring multi-perspective annotation quality under low-resource conditions.
- Extensive experiments across four public datasets demonstrate that FLEXTAG outperforms GPT-4o zero-shot on tag selection, surpasses all comparable-scale baselines on open generation, and maintains strong generalization on out-of-distribution datasets.

2 Related Work

Instruction Tagging. Instruction tagging aims to map free-form user requests into structured metadata. INSTAG (Lu et al., 2023) leverages ChatGPT to generate open-ended keyword tags for analyzing SFT data diversity and complexity. TAGROUTER (Chen et al., 2025) uses tags for LLM routing to balance cost and performance. TAGGPT (Li et al., 2023) explores zero-shot multimodal tagging via prompt engineering. These methods focus on flat keyword generation and do not support multi-dimensional schema-constrained annotations or multi-perspective quality guarantees. Beyond instruction tagging, TAGLLM (Shen et al., 2024) repurposes tags as learnable embeddings to control model behavior, and LLM4TAG (Tang et al., 2025) targets industrial-scale content tagging via graph-based recall and LLM-based selection, both representing orthogonal directions to structured metadata annotation. Work on instruction tuning (Wei et al., 2022; Wang et al., 2022, 2023) and data augmentation (Xu et al., 2025) further demonstrates that diverse supervision signals are critical for generalization. FLEXTAG builds on these directions and targets multi-perspective tagging supporting both tag selection and open generation.

Small Language Models. Language models are highly sensitive to prompt wording and formatting, where semantically equivalent variants can cause substantial performance fluctuations (Razavi et al., 2025; Sclar et al., 2024; Zhuo et al., 2024). This challenge is particularly acute for SLMs, which are attractive for deployment due to low latency and cost (Ding et al., 2023; Gunasekar et al., 2023) but struggle to retain broad instruction-following competence under narrow task-specific supervision (Gudibande et al., 2023). In schema-constrained multi-template tagging, this limitation is further amplified: the model must maintain stable outputs across diverse instruction formats while satisfying multi-perspective quality requirements under low-resource conditions. Template-based augmentation (Longpre et al., 2023; Mishra et al., 2022) is a common mitigation strategy, and post-training methods such as SFT, DPO (Rafailov et al., 2023), and GRPO (Shao et al., 2024) expand the adaptation toolbox. However, optimizing SLMs for structured multi-template tagging remains underexplored. FLEXTAG addresses this gap by combining multi-template augmentation with structure-aware training objectives.

3 Preliminaries

Notations. Let $\mathcal{X}$ denote the space of natural-language query and $\mathcal{T} = \{\tau_1, \ldots, \tau_K\}$ the set of tagging templates. Each template τ specifies an input/output formatting protocol (*e.g.,* field order, candidate list rendering) and template-dependent label vocabularies, inducing an output space $\mathcal{Y}_\tau$. We further define the valid (format- and value-compliant) output set $\mathcal{Y}_\tau^{\text{val}} \subseteq \mathcal{Y}_\tau$. We view $\mathcal{Y}_\tau$

as the space of all template-typed structured outputs, and $\mathcal{Y}_\tau^{\text{val}}$ as the subset satisfying the template-defined format/value constraints. The dataset is grouped by template as $\mathcal{D} = \bigcup_{\tau\in\mathcal{T}} \mathcal{D}_\tau$, where $\mathcal{D}_\tau = \{(x_i, y_i)\}_{i=1}^{N_\tau}$ and $y_i \in \mathcal{Y}_\tau^{\text{val}}$. A template τ can be viewed as a set of facets $\mathcal{K}_\tau = \{1, \ldots, M_\tau\}$, where each facet k has its own value space $\mathcal{V}_{\tau,k}$, and $\mathcal{Y}_\tau = \prod_{k\in\mathcal{K}_\tau} \mathcal{V}_{\tau,k}$. In our experiments, $M_\tau = 5$ with four single-valued facets and one set-valued facet (topics), covering both (i) *tag selection* from finite candidates and (ii) *open-tag generation*.

Tagging objective. Given an instruction x and a template τ, a single tagger produces a structured prediction $\hat{y} = f_\theta(x; \tau) \in \mathcal{Y}_\tau$. We adopt the following unified empirical objective:

$$\min_\theta \; \mathbb{E}_{\tau\sim p(\tau)} \left[\frac{1}{|\mathcal{D}_\tau|} \sum_{(x_i,y_i)\in\mathcal{D}_\tau} \mathcal{L}_{\text{NLL}}(f_\theta(x_i;\tau),\, y_i) \right], \tag{1}$$

where $\mathcal{L}_{\text{NLL}}$ is the token-level negative log-likelihood loss, and $p(\tau)$ is the template sampling distribution (uniform over $\mathcal{T}$ by default).

Limitations of tagging optimization. Most prior taggers implicitly assume a fixed template τ^* shared by training and inference, optimizing primarily for label accuracy under that template. Beyond the template generalization gap, minimizing Eq. (1) alone neither encodes template validity (format/value compliance) nor considers *multi-perspective* desiderata, especially preventing facet collapse and reducing redundant dependence across facets.

4 Method

4.1 Multi-template Data Augumentation

Existing instruction tagging datasets typically cover a single template, causing models to entangle the annotation schema with a fixed input format and degrade under template shifts. We therefore build a multi-template augmentation pipeline that projects the same semantic supervision signal across heterogeneous template formats(Details in Appendix C). The pipeline has three steps. *(i)* We aggregate raw samples from multiple public instruction sources and prompt GPT-4o to produce multi-facet structured labels, yielding a base annotation set $\mathcal{D}_{\text{base}}$ that serves as the semantic anchor for template projection. *(ii)* We filter $\mathcal{D}_{\text{base}}$ by discarding samples with extremely low-frequency tag values and removing samples with missing facet fields. *(iii)* We apply multi-template rewriting to each sample in $\mathcal{D}_{\text{base}}$, producing a set of template instances $\{(x^{(\tau)}, y^{(\tau)})\}_{\tau\in\mathcal{T}(x)}$. The rewriting covers four template categories (open-tag generation, tag selection, zero-shot, and few-shot), each with multiple variants, and we verify the projected supervision satisfies $y^{(\tau)} \in \mathcal{Y}_\tau^{\text{val}}$. The resulting dataset is grouped by template as $\mathcal{D} = \bigcup_{\tau\in\mathcal{T}} \mathcal{D}_\tau$.

4.2 Tag-focused Supervision

In multi-template tagging, model outputs typically consist of structured tag fields interleaved with auxiliary text such as explanations or structural tokens. Applying uniform token-level NLL across all positions dilutes the training signal with low-information-density spans, an effect that becomes more pronounced when templates contain lengthy auxiliary text. We therefore propose a span-weighted supervision strategy that assigns an amplified weight $\alpha > 1$ to tag-value spans and weight 1 to all remaining tokens:

$$\mathcal{L}_{\text{wNLL}} = -\sum_{t=1}^{|y|} m_t \log p_\theta(y_t \mid y_{<t}, x, \tau)\,, \tag{2}$$

where $m_t = \alpha$ if position t falls within a tag-value span and $m_t = 1$ otherwise. This weighting relies solely on the parseable boundaries of tag fields, requiring no facet-specific semantic knowledge, and thus applies uniformly across all template variants. Intuitively, $\mathcal{L}_{\text{wNLL}}$ redistributes the gradient budget away from verbose auxiliary text toward high-information-density label spans, improving tag field prediction quality across templates.

4.3 Multi-perspective Generation

While $\mathcal{L}_{\text{wNLL}}$ improves supervision on tag fields, token-level likelihood training alone leaves three desiderata unaddressed for multi-facet structured generation templates $\mathcal{T}_{\text{gen}} \subseteq \mathcal{T}$: *(i) coverage*—predictions over enumerable facets should preserve marginal coverage rather than collapsing toward high-frequency values; *(ii) de-redundancy*—facets should not become mutually predictable and degenerate into redundant restatements of the same semantic signal; *(iii) synergy*—facets should provide complementary evidence such that joint conditioning reduces prediction uncertainty on a held-out facet more than any single-facet conditioning. Inspired by partial information decomposition (PID) theory (Venkatesh et al., 2023), we introduce three lightweight regularizers targeting these goals.

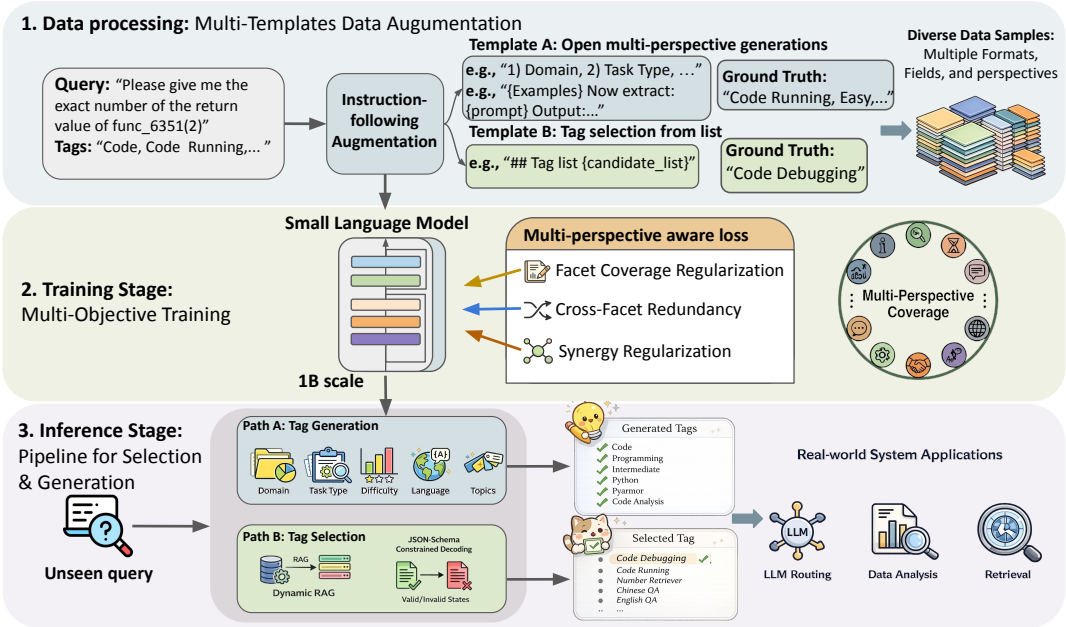

Figure 1: **FlexTag** overview. Semantically aligned multi-template augmentation yields diverse tagging templates. A single 1B tagger is trained with span-weighted supervision and multi-perspective regularizers (coverage, de-redundancy, synergy). At inference, open-generation templates decode multi-dimensional structured tags, while selection templates use RAG few-shot demonstrations and template-derived constrained decoding to ensure validity.

Facet Coverage. For facets with finite value spaces,

$$\mathcal{K}_{\text{enum}}(\tau) = \{k \in \mathcal{K}_\tau : |\mathcal{V}_{\tau,k}| < \infty\},$$

maximum-likelihood training tends to collapse predictions toward majority-class values. Let $\pi_{\tau,k}$ denote the empirical label distribution computed from the training corpus prior to training, and let $\hat{p}_{\tau,k}$ denote the batch-averaged predicted marginal for facet k, obtained by aggregating sample-level distributions $p_\theta(\cdot \mid x, \tau, k)$, where $p_\theta(\cdot \mid x, \tau, k)$ is the model-induced prediction distribution over $\mathcal{V}_{\tau,k}$ under input x and template τ. We penalize deviation from the empirical prior via KL divergence (Kullback and Leibler, 1951):

$$\mathcal{L}_{\text{cover}} = \sum_{k \in \mathcal{K}_{\text{enum}}(\tau)} \text{KL}(\pi_{\tau,k} \,\|\, \hat{p}_{\tau,k}) \,. \quad (3)$$

By anchoring to the empirical prior rather than a uniform distribution, this regularizer penalizes only collapse beyond the natural label skew already present in the data, while leaving open-ended facets unconstrained.

Cross-Facet Redundancy. Even with adequate per-facet coverage, the model may learn shortcut cross-facet dependencies: if one facet's prediction is largely determined by another, the multi-facet output degenerates into redundant restatements. For each sample x and facet k, we define the facet distribution as the average predictive distribution over token positions assigned to facet k:

$$\hat{p}_k(x) = \frac{1}{|\mathcal{P}_k(x)|} \sum_{t \in \mathcal{P}_k(x)} p_\theta(\cdot \mid x_{<t}, \tau) \quad (4)$$

where $\mathcal{P}_k(x)$ is the set of token positions assigned to facet k. Facets in $\mathcal{K}_{\text{main}}$ each correspond to a single tag token, while open-ended facets span multiple tokens and yield stable distributional estimates. We therefore penalize the cosine similarity between each $k \in \mathcal{K}_{\text{main}}$ and each open-ended facet $k' \in \mathcal{K}_\tau \setminus \mathcal{K}_{\text{main}}$. Let $\rho_k(x) = \cos(\hat{p}_k(x),\, \hat{p}_{k'}(x))$, then:

$$\mathcal{L}_{\text{redund}} = \sum_{x:\, \tau(x) \in \mathcal{T}_{\text{gen}}} \sum_{k \in \mathcal{K}_{\text{main}}} \max(0,\ \rho_k(x)). \quad (5)$$

This soft constraint discourages vocabulary co-movement across facets without imposing strict statistical independence.

Synergy. To operationalize synergy, at each training step we randomly hold out one anchor facet and compare the prediction loss under joint conditioning versus single-facet conditioning. Concretely, we sample a target facet $s \sim \text{Unif}(\mathcal{K}_\tau)$ and a source facet $k \sim \text{Unif}(\mathcal{K}_\tau \setminus \{s\})$. Let $\mathcal{L}_{\text{anchor}}^{(\text{all}\to s)}$ denote the span-level NLL for predicting facet s under joint conditioning (all other facets provided as

Algorithm 1 Template-adaptive constrained inference for selection templates.

Require: Input instruction x, selection template $\tau \in \mathcal{T}_{\text{sel}}$, example library $\mathcal{B}_\tau$, tagger f_θ, $\mathbf{CD}(f_\theta, \mathbf{P}, \mathcal{C})$: constrained decoding of f_θ under prompt $\mathbf{P}$ and constraints $\mathcal{C}$
Ensure: Structured prediction $\hat{y} \in \mathcal{Y}_\tau^{\text{val}}$
1: Retrieve demonstrations $\mathcal{E}_\tau(x)$ from $\mathcal{B}_\tau$
2: Construct $\mathbf{P} \leftarrow \text{Prompt}(x, \tau, \mathcal{E}_\tau(x))$
3: Derive $\mathcal{C}_\tau$ from template rules (structure + admissible candidates)
4: $\hat{y} \leftarrow \mathbf{CD}(f_\theta, \mathbf{P}, \mathcal{C}_\tau)$
5: **return** $\hat{y}$

context), and let $\mathcal{L}_{\text{anchor}}^{(k\to s)}$ denote the same loss when only facet k is retained as context. We impose the following margin constraint:

$$\begin{aligned} \Delta_{s,k} &= \text{sg}(\mathcal{L}_{\text{anchor}}^{(k\to s)}) - \mathcal{L}_{\text{anchor}}^{(\text{all}\to s)}, \\ \mathcal{L}_{\text{syn}} &= \mathbb{E}_{s,k}\big[\max(0,\ \gamma - \Delta_{s,k})\big]. \end{aligned} \tag{6}$$

where $\gamma > 0$ is a margin hyperparameter and $\text{sg}(\cdot)$ denotes stop-gradient.

4.4 Objective Overview

We optimize a tag-focused likelihood term for all templates, and additionally apply multi-perspective regularizers for multi-facet structured generation templates $\tau \in \mathcal{T}_{\text{gen}}$. The per-sample objective is:

$$\begin{aligned} \mathcal{L}(x, y; \tau) &= \mathcal{L}_{\text{wNLL}}(x, y; \tau) + \mathbf{1}_{\tau\in\mathcal{T}_{\text{gen}}}\, \lambda\, \mathcal{R}(x, y; \tau), \\ \mathcal{R}(x, y; \tau) &= \mathcal{L}_{\text{cover}} + \mathcal{L}_{\text{redund}} + \mathcal{L}_{\text{syn}}, \end{aligned} \tag{7}$$

where $\lambda \geq 0$ controls the overall strength of the multi-perspective regularization terms.

4.5 Template-Adaptive Constrained Inference

While training-time objectives improve tag prediction quality, a small language model can remain brittle at inference time under template shifts. This is most pronounced for *selection-style* templates, where the output must satisfy strict structure and candidate-set constraints (i.e., $\hat{y} \notin \mathcal{Y}_\tau^{\text{val}}$ leads to invalid predictions). We therefore design a template-adaptive inference procedure specifically for $\tau \in \mathcal{T}_{\text{sel}}$ that combines retrieval-augmented few-shot prompting with constrained decoding (Geng et al., 2023; Lewis et al., 2020) to ensure output validity. For each selection template τ, we maintain a demonstration library $\mathcal{B}_\tau$ of template-formatted input–output examples. Given (x, τ), we retrieve a small set of relevant demonstrations from $\mathcal{B}_\tau$ and prepend them to the prompt as few-shot context, improving template following and reducing formatting errors. We additionally derive a constraint set $\mathcal{C}_\tau$ from template rules, capturing structural constraints (field/layout) and value constraints (admissible candidate sets). During decoding, we mask invalid continuations that violate $\mathcal{C}_\tau$, guaranteeing $\hat{y} \in \mathcal{Y}_\tau^{\text{val}}$ for selection templates. Algorithm 1 summarizes the inference pipeline.

5 Experiment

5.1 Settings

Training Data. We construct FlexTag-Bench as our training corpus, aggregated from 18 publicly available instruction sources, and annotated by GPT-4o with subsequent quality filtering. The final dataset comprises approximately 28K samples for supervised fine-tuning and 235 samples for preference optimization. Details of the data construction pipeline and statistics are provided in Appendix B.

Evaluation Datasets. To ensure fair evaluation and validate generalization, we evaluate on four public datasets. For all datasets, we use GPT-4o to generate ground-truth annotations with multi-perspective semantic tags, following a unified annotation and evaluation protocol (see Appendix D). `InfiniteBench` (Zhang et al., 2024) is a long-context benchmark spanning 12 diverse domains, from which we sample 50 instances per domain (600 total). `Helmet` (Yen et al., 2024) covers 7 task categories, including retrieval-augmented generation, summarization, and long-document QA (100 instances per category, 700 total). `RouterBench` (Hu et al., 2024) aggregates prompts from mainstream benchmarks including MBPP (Austin et al., 2021), GSM-8k (Cobbe et al., 2021), MMLU (Hendrycks et al., 2020), and MT-Bench (Bai et al., 2024), covering code generation, mathematical reasoning, and commonsense QA (1K instances). `OASST1` (Köpf et al., 2023) is a crowd-sourced multilingual conversation corpus spanning 35 languages, representing real-world human-written instructions (1K instances). For inputs exceeding the model's context window, we truncate by retaining the first and last 300 tokens.

Baselines. We select baselines along four dimensions for a comprehensive comparison: *(1) Task-specialized taggers*: INSTAGGER (Lu et al., 2023)[7B] and TAGROUTER (Chen et al., 2025)[0.5B], representing mainstream approaches specifically fine-tuned for instruction tagging. *(2) General-purpose small language models*: MISTRAL-v0.1-Instruct[7B] (Jiang

Table 1: Open-generation tagging results on `InfiniteBench`, `Helmet`, `RouterBench`, and `OASST1`. We compare against prior tagging-specific methods and pretrained small language models. We report semantic accuracy (Recall, Precision, and F1) in %. The best results are in **bold**, and the second-best results are underlined.

Method	InfiniteBench			Helmet			RouterBench			OASST1		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
Mistral-v0.1-Instruct [7B]	18.4	21.1	18.1	19.8	14.4	13.6	18.9	15.6	14.9	29.3	19.7	21.4
Phi-3.5-vision-instruct [4.2B]	13.7	14.7	12.8	14.1	10.9	10.6	8.1	9.8	7.7	23.7	20.7	19.8
Phi-3.5-mini-instruct[3.8B]	19.9	20.7	18.9	21.6	15.7	16.6	11.1	8.2	8.5	35.3	25.0	27.1
Llama-3.2-Instruct [3B]	19.3	20.2	17.9	23.0	15.4	16.6	25.9	22.1	21.9	25.5	19.3	20.2
Gemma3 [1B]	17.6	13.7	13.7	19.3	15.5	15.3	13.2	15.3	12.1	28.1	21.8	22.8
Qwen3.5 [0.8B]	9.8	21.4	12.8	16.1	**19.6**	16.0	8.9	18.2	11.1	11.5	14.1	11.6
Qwen3 [0.6B]	6.1	12.5	7.9	13.4	18.2	13.4	6.8	26.4	9.3	13.4	18.0	14.0
Qwen2.5-Instruct [0.5B]	4.5	12.9	6.1	8.6	13.0	9.0	2.9	9.7	3.8	12.4	19.1	13.2
Instagger [7B]	17.8	**30.0**	21.0	10.3	11.6	9.8	21.3	**35.4**	24.8	7.9	12.1	8.5
TagRouter [0.5B]	9.1	17.7	11.3	8.6	13.4	9.8	14.2	31.9	18.7	9.0	15.0	10.7
FlexTag w/ DPO [1B]	20.4	23.2	20.6	24.0	19.4	19.0	38.3	29.4	31.6	36.6	25.2	28.1
FlexTag w/ GRPO [1B]	21.0	24.5	21.5	24.6	**19.6**	**19.5**	**38.7**	29.2	**31.7**	36.6	25.1	28.1
FlexTag w/ DPO [4B]	**24.5**	24.9	23.2	25.3	17.4	19.1	37.5	29.1	31.2	36.6	25.3	28.1
FlexTag w/ GRPO [4B]	**24.5**	25.2	**23.7**	**25.9**	17.9	**19.5**	37.5	29.2	31.2	**37.1**	**25.4**	**28.3**

Table 2: Tag selection performance (Exact Match, %) on `InfiniteBench` and `Helmet`. Higher is better. The best results are in **bold**, and the second-best results are underlined.

Method	InfiniteBench	Helmet
GPT-4o w/ RAG [>100B] (Skyline)	91.6	66.9
GPT-4o [>100B]	82.3	42.7
Mistral-v0.1-Instruct [7B]	74.2	44.7
Phi-3.5-vision-instruct [4.2B]	75.0	31
Phi-3.5-mini-instruct[3.8B]	74.3	44.2
Llama-3.2-Instruct [3B]	52.7	31.6
Gemma3 [1B]	33.3	32.3
Qwen3.5 [0.8B]	30.0	3.1
Qwen3 [0.6B]	24.1	27.1
Qwen2.5-Instruct [0.5B]	12.5	6.9
Instagger [7B]	6.9	0*
TagRouter [0.5B]	0*	0*
FlexTag w/ DPO [1B]	**87.9**	55.0
FlexTag w/ GRPO [1B]	87.6	53.1
FlexTag w/ DPO [4B]	87.1	55.9
FlexTag w/ GRPO [4B]	85.9	**56.1**

*: A value of 0 indicates that the model produced no correct predictions under the corresponding evaluation metric.

et al., 2023), PHI-3.5-vision-instruct[4.2B] and PHI-3.5-mini-instruct[3.8B] (Abdin et al., 2024), LLAMA-3.2-Instruct[3B] (Grattafiori et al., 2024), GEMMA3 (Team et al., 2025) [1B], QWEN3.5 [0.8B], QWEN3[0.6B], and QWEN2.5-Instruct[0.5B] (Yang et al., 2025), evaluated in a zero-shot setting. These models are comparable in scale to FLEXTAG and serve to validate the necessity of task-specific training. *(3) Skyline*: GPT-4O (Achiam et al., 2023) and GPT-4O w/ RAG represent the performance ceiling. For tag selection, GPT-4O w/ RAG incorporates retrieval-augmented few-shot demonstrations. All baselines are evaluated using identical inference prompts to ensure fair comparison.

Evaluation Metrics. We consider two evaluation settings: open-ended tag generation and candidate-constrained tag selection.

(i) Open Generation. Since exact string matching is overly strict and cannot accommodate benign lexical variations, we adopt semantic soft matching based on PHRASEBERT (Wang et al., 2021): a ground-truth tag is considered matched if its cosine similarity with any predicted tag exceeds the threshold $\sigma = 0.8$. Based on this rule, we compute sample-level Recall, Precision, and F1, and report macro-averaged scores over the test set. Beyond semantic correctness, we further evaluate whether predicted tags satisfy the quality requirements of multi-facet tagging through two additional metrics. Coverage measures whether the predicted tag distribution aligns with the ground-truth distribution rather than collapsing to a few dominant values:

$$\text{Coverage} = 1 - \frac{\text{JSD}(\hat{p} \| p^*)}{\log 2} \tag{8}$$

where $\hat{p}$ and p^* denote the predicted and ground-truth distributions, respectively, and JSD (Lin, 2002) is normalized to $[0, \log 2]$. Higher Coverage indicates better distributional alignment. Redundancy measures the degree of semantic overlap among predicted tags:

$$\text{Redundancy} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{2}{k_s(k_s - 1)} \sum_{i<j} \cos(\mathbf{e}_i, \mathbf{e}_j) \tag{9}$$

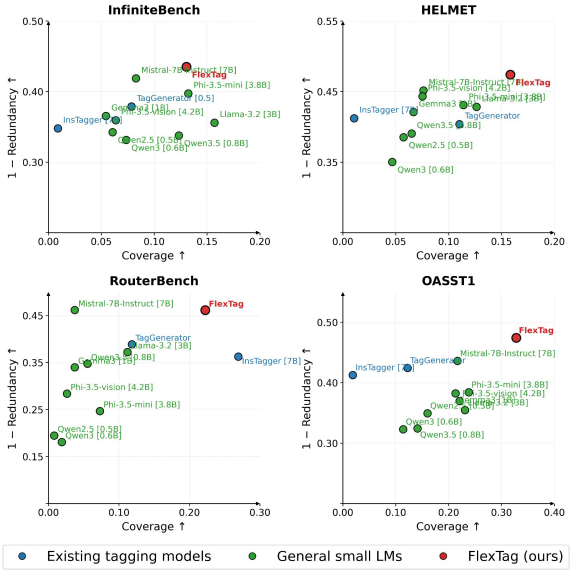

Figure 2: Comparison of multi-perspective performance across models. X-axis: Coverage (↑); Y-axis: 1-Redundancy (↑). The upper-right region indicates better performance. Colors denote model categories: blue = existing tagging models, green = general small LMs, red = FLEXTAG (ours).

where $\mathcal{S}$ is the set of samples with $k_s \geq 2$ predicted tags, and $\mathbf{e}_i$ denotes the PHRASEBERT (Wang et al., 2021) embedding of tag t_i. Lower redundancy indicates that the model generates complementary rather than repetitive labels.

(ii) Tag Selection. We use Exact Match (EM), which measures the proportion of samples whose predicted labels exactly match the ground-truth labels:

$$\text{EM} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\hat{y}_i = y_i] \tag{10}$$

where $\hat{y}_i$ and y_i denote the predicted and ground-truth labels for the i-th sample, respectively.

5.2 Main Results

Open Tagging Generation Performance. Table 1 presents open-generation results across four datasets. FLEXTAG at 1B scale outperforms nearly all baselines across all four datasets. On RouterBench, FLEXTAG w/ GRPO (Shao et al., 2024) achieves 31.7% F1, while the same-scale GEMMA3 [1B] obtains only 12.1% F1. On two OOD test sets, FLEXTAG maintains a substantial advantage with 21.5% F1 on InfiniteBench and 19.5% F1 on HELMET. Notably, task-specialized taggers perform poorly on open generation: despite having 7× more parameters, INSTAGGER [7B] achieves below 25% F1 across all datasets, and TAGROUTER [0.5B] remains below 19% F1, as these methods are designed for coarse-grained keyword tagging and fail to capture multi-perspective quality requirements. General-purpose small models including GEMMA3 [1B] and the QWEN series perform even weaker under zero-shot settings with F1 scores generally in the 10%–15% range, indicating that pretraining alone is insufficient for multi-perspective tag generation. In contrast, FLEXTAG effectively bridges this gap through multi-template training and multi-objective optimization.

Tag Selection Performance. Table 2 presents tag selection results on InfiniteBench and HELMET, two OOD test sets. FLEXTAG at 1B scale significantly outperforms GPT-4o on both datasets: FLEXTAG w/ DPO achieves 87.9% EM on InfiniteBench, surpassing GPT-4o by 5.6 percentage points, and 55.0% EM on HELMET, exceeding GPT-4o by 12.3 percentage points. Compared to GPT-4o w/ RAG, FLEXTAG trails by only 3.7% on InfiniteBench and 11.9% on HELMET. Given that FLEXTAG has approximately 1% of GPT-4o's parameters, these results demonstrate that targeted training can effectively compensate for the parameter gap on specific tasks. FLEXTAG also exhibits stronger cross-dataset stability: PHI-3.5-vision-instruct [4.2B] achieves 75.0% on InfiniteBench but drops sharply to 31.0% on HELMET, whereas FLEXTAG maintains consistently high performance across both datasets. Task-specialized taggers suffer severe generalization failure under template shift, with INSTAGGER [7B] obtaining only 6.9% EM on InfiniteBench and 0% on HELMET, and TAGROUTER [0.5B] failing to produce a single correct prediction on either dataset, directly corroborating the template rigidity problem we identified. Notably, DPO (Rafailov et al., 2023) slightly outperforms GRPO (Shao et al., 2024) on tag selection on both datasets, possibly because tag selection is a deterministic choice task where DPO's stable optimization is more suitable.

Multi-perspective Performance. Figure 2 compares model performance across Coverage and Redundancy dimensions. The X-axis represents Coverage (higher is better), and the Y-axis represents 1-Redundancy (higher is better). The upper-right region indicates optimal performance: models that generate diverse tag distributions while avoiding semantic redundancy. Across all four datasets, FLEXTAG consistently occupies the upper-right region, demonstrating the best multi-perspective perfor-

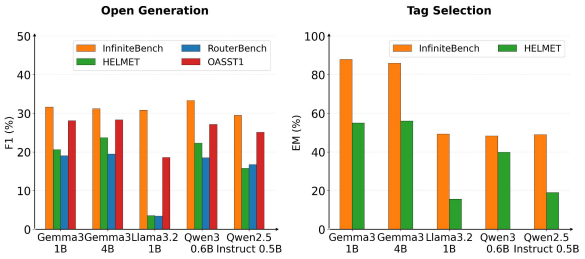

Figure 3: Sensitivity analysis of FLEXTAG across different backbone models over four datasets. Left: F1 score for open tag generation. Right: exact match (EM) for tag selection.

mance. On OASST1, FLEXTAG achieves the highest Coverage (0.33) while maintaining low Redundancy. Existing task-specialized tagging models exhibit severe Coverage deficiency: InsTagger [7B] achieves near-zero Coverage on InfiniteBench and HELMET, indicating that its outputs collapse to a few high-frequency labels and fail to generate diverse multi-perspective tags. This further corroborates that these models are not designed to meet the quality requirements of multi-perspective tagging. General-purpose small models such as the QWEN series cluster in the lower-left region, with low Coverage and high Redundancy. For example, QWEN3 [0.6B] achieves only 0.18 for 1-Redundancy on RouterBench, indicating substantial semantic overlap among its generated tags. In contrast, FLEXTAG effectively balances Coverage and Redundancy through multi-template training and multi-objective optimization, achieving strong multi-perspective performance across all datasets.

Ablation Studies Table 3 presents the ablation study of the three auxiliary loss terms. Overall, removing any single loss term leads to performance degradation on most benchmarks, with the effect more pronounced on long-context OOD datasets (InfiniteBench and HELMET) than on general datasets (RouterBench and OASST1). Removing $\mathcal{L}_{\text{red.}}$ causes the largest drop on InfiniteBench (14.2%), suggesting that cross-facet redundancy regularization contributes most to generalization on long-context inputs. Removing $\mathcal{L}_{\text{cov.}}$ yields mixed results: performance declines on HELMET and RouterBench, while InfiniteBench shows a marginal improvement, possibly reflecting the interaction between coverage regularization and the relatively narrow label space of that dataset. Removing $\mathcal{L}_{\text{syn.}}$ leads to consistent but modest degradation across benchmarks, indicating a stable marginal contribution. These results suggest that the three

Table 3: Ablation study of auxiliary loss terms in FLEXTAG. We report F1 (%) on open-generation tagging. **Bold** denotes the best result.

Ablations	`InfiniteBench`	`HELMET`	`RouterBench`	`OASST1`
FlexTag [1B]	**21.5**	**19.5**	**31.7**	**28.1**
w/o $\mathcal{L}_{\text{cov.}}$	20.1 ↓1.4	17.9 ↓1.6	30.2 ↓1.5	27.0 ↓1.1
w/o $\mathcal{L}_{\text{syn.}}$	18.4 ↓3.1	17.7 ↓1.8	31.3 ↓0.4	26.6 ↓1.5
w/o $\mathcal{L}_{\text{red.}}$	14.2 ↓7.3	17.2 ↓2.3	31.4 ↓0.3	25.8 ↓2.3

regularizers are complementary, and that their benefits are more salient under distribution shift. We provide additional ablation results across different backbone models in Appendix E.

Sensitivity Analysis. Figure 3 presents FLEXTAG performance across different backbone models. For open generation, all four backbones perform comparably on RouterBench with F1 between 29.5% and 33.3%, but significant divergence emerges on OOD datasets: LLAMA3.2 (Grattafiori et al., 2024) [1B] suffers severe degradation on InfiniteBench and HELMET with F1 of only 3.5% and 3.4%, while GEMMA3 [1B] maintains robust performance across all four datasets. For tag selection, the gap is more pronounced: GEMMA3 [1B] leads substantially on both InfiniteBench and HELMET with 87.9% and 55.0% EM, whereas other backbones cluster around 48%–49%. This discrepancy likely stems from differences in instruction-following and format compliance capabilities established during pretraining.

6 Conclusion

We presented FLEXTAG, a 1B-scale instruction tagging framework addressing template rigidity and multi-perspective quality in production tagging systems. Under low-resource samples, FLEXTAG outperforms GPT-4o zero-shot on tag selection, surpasses all comparable-scale baselines on open generation, and maintains strong generalization on out-of-distribution schema. Several limitations remain. Evaluation focuses on general instruction tagging, with finer-grained scenarios such as hierarchical and ontology-based annotation yet to be systematically studied. Also, while we evaluate coverage and redundancy, establishing a comprehensive standardized protocol for multi-perspective tagging quality remains an open problem. Future work will investigate adapting FLEXTAG to specialized domains such as healthcare, law, and finance under scarcer annotation budgets, while preserving tag quality across diverse and evolving schema.

7 Limitations

Our work has several limitations. First, FLEXTAG is designed around a predefined multi-facet tagging schema, which means its effectiveness still depends on the quality and stability of the schema itself. When the target facets are redefined, expanded, or made more hierarchical, additional annotation and adaptation may be required. Second, although FLEXTAG improves robustness to template variation, our experiments mainly study template diversity at the instruction formatting level rather than deeper shifts in task semantics, user intent ambiguity, or domain-specific labeling conventions. Third, the full inference pipeline may introduce extra overhead when constrained decoding and retrieval-augmented few-shot prompting are used together, which could limit deployment in latency-sensitive settings. Finally, while our automatic metrics capture important aspects such as facet coverage and redundancy, they may not fully reflect whether the predicted tags are maximally informative or equally useful for all downstream applications.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and 1 others. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454.

Zhou Chen, Zhiqiang Wei, Yuqi Bai, Xue Xiong, and Jianmin Wu. 2025. Tagrouter: Learning route to llms through tags for open-domain text generation tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21539–21564.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, and 1 others. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured nlp tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *Preprint*, arXiv:2305.15717.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need. *Preprint*, arXiv:2306.11644.

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

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, and 1 others. 2023. Openassistant conversations-democratizing large language model alignment. *Advances in neural information processing systems*, 36:47669–47681.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Chen Li, Yixiao Ge, Jiayong Mao, Dian Li, and Ying Shan. 2023. Taggpt: Large language models are zero-shot multimodal taggers. *arXiv preprint arXiv:2304.03022*.

Jianhua Lin. 2002. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. *Preprint*, arXiv:2301.13688.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. *arXiv preprint arXiv:2308.07074*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. *Preprint*, arXiv:2104.08773.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models. In *European Conference on Information Retrieval*, pages 303–313. Springer.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *Preprint*, arXiv:2310.11324.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Junhong Shen, Neil Tenenholtz, James Brian Hall, David Alvarez-Melis, and Nicolo Fusi. 2024. Tag-llm: Repurposing general-purpose llms for specialized domains. *arXiv preprint arXiv:2402.05140*.

Ruiming Tang, Chenxu Zhu, Bo Chen, Weipeng Zhang, Menghui Zhu, Xinyi Dai, and Huifeng Guo. 2025. Llm4tag: Automatic tagging system for information retrieval via large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 4882–4890.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Praveen Venkatesh, Corbett Bennett, Sam Gale, Tamina Ramirez, Greggory Heller, Severine Durand, Shawn Olsen, and Stefan Mihalas. 2023. Gaussian partial information decomposition: Bias correction and application to high-dimensional data. *Advances in Neural Information Processing Systems*, 36:74602–74635.

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021. Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration. *Preprint*, arXiv:2109.06304.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. *Preprint*, arXiv:2212.10560.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, and 21 others. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *Preprint*, arXiv:2204.07705.

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

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2025. Wizardlm: Empowering

large pre-trained language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2024. Helmet: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and 1 others. 2024. Infinitebench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277.

Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. Prosa: Assessing and understanding the prompt sensitivity of llms. *Preprint*, arXiv:2410.12405.

A Mathematical Notation

Table 4 summarizes the key mathematical symbols used throughout the paper.

Symbol	Description
Problem Setup	
$x,\ \mathcal{X}$	Input query and its space
$\tau,\ \mathcal{T}$	A template and the set of all templates
$\mathcal{K}_\tau,\ \mathcal{V}_{\tau,k}$	Facet index set and value space for facet k under τ
$\mathcal{Y}_\tau,\ \mathcal{Y}_\tau^{\text{val}}$	Full output space and valid output subset for template τ
$\mathcal{D}_\tau,\ \mathcal{D}$	Training set for template τ; full dataset $\bigcup_\tau \mathcal{D}_\tau$
f_θ	Parameterized tagger model
Training Objectives	
α	Upweighting factor for tag-value token spans in $\mathcal{L}_{\text{wNLL}}$
$\mathcal{T}_{\text{gen}} \subseteq \mathcal{T}$	Multi-facet structured generation templates
$\mathcal{K}_{\text{enum}}(\tau)$	Enumerable facets (finite value spaces) under τ
$\mathcal{K}_{\text{main}}$	Main single-valued facets: domain, task, difficulty, language
$\pi_{\tau,k},\ \hat{p}_{\tau,k}$	Empirical and predicted marginal distribution for facet k
$\mathcal{P}_k(x)$	Token positions assigned to facet k for input x
$\rho_k(x)$	Cosine similarity between facet k and an open-ended facet
$s,\ k$	Target and source facet sampled for $\mathcal{L}_{\text{syn}}$
$\Delta_{s,k}$	Synergy gap: $\text{sg}(\mathcal{L}^{(k\to s)}) - \mathcal{L}^{(\text{all}\to s)}$
$\gamma,\ \lambda$	Margin for $\mathcal{L}_{\text{syn}}$; regularization coefficient for $\mathcal{R}$
$\text{sg}(\cdot)$	Stop-gradient operator
Evaluation & Inference	
σ	Cosine similarity threshold for semantic matching (σ=0.8)
$\mathbf{e}_i$	PhraseBERT embedding of predicted tag t_i
$\hat{p},\ p^*$	Predicted and ground-truth tag distributions for Coverage
$B_\tau,\ C_\tau$	Demonstration library and constraint set for template τ

Table 4: Mathematical notation used throughout the paper.

B FlexTag-Bench Pipeline

B.1 Data Sources

FlexTag-Bench is aggregated from 18 publicly available instruction datasets spanning 7 categories, yielding 25,900 raw samples. Table 5 summarizes the data sources and their corresponding sample counts.

B.2 Ground-Truth Annotation

We use GPT-4o to annotate all samples with multi-facet tags. To ensure annotation quality and safety, we design **category-specific prompt templates** tailored to each data category. All templates follow a unified five-facet annotation schema: Domain → Task Type → Difficulty → Language → Topics.

Category	Dataset	#Samples
Math	GSM8k	2,000
Code	MBPP	974
	DS1000	1,000
General Conversation	Lima	1,003
	Alpaca	2,000
	Dolly	2,000
	OASST1	2,000
	RouterBench	2,000
	OpenChat	2,000
	Unnatural	2,000
	WizardLM	2,000
PII	pii-masking-200k	2,000
Toxic	lmsys/toxic-chat	2,000
Company Confidential	Contract	423
National Security	ScaleAI/fortress_public	500
Harmful	LLM-LAT/harmful-dataset	2,000

Table 5: Data sources for FlexTag-Bench construction.

> You are a tagging system that labels the key intent of a user instruction. Below is an instruction:
> [begin]
> {instruction}
> [end]
>
> Follow this process: (1) Identify the main **domain** (knowledge field). (2) Identify the **task type** (what the user wants done). (3) Rate the **difficulty** (Easy / Intermediate / Hard). (4) Detect the **language** of the instruction. (5) Extract 2–4 specific **topics** (what the instruction is about).
>
> Each topic must be output as a separate tag object. Output tags in strict order: Domain → Task Type → Difficulty → Language → Topic(s). Your response must strictly follow this JSON format: [{"tag": str, "explanation": str}].

Table 6: GPT-4o prompt template for annotating general instruction tags.

For sensitive categories (PII, Toxic, Confidential, Security, Harmful), we add safety constraints to prevent the model from restating sensitive content.

The remaining category-specific templates (Math, Code, Toxic, Confidential, Security, Harmful) follow the same structure with appropriate modifications for their respective content types.

B.3 Quality Filtering

After annotation, we apply the following filtering steps: (1) **Format filtering**: Remove samples with JSON parsing failures or missing facet fields. (2)

You are a tagging system that labels the key intent of a **PII-related** user instruction. Below is an instruction:
[begin]
{instruction}
[end]

Follow this process: (1) Identify the main **domain**. (2) Identify the **task type**. (3) Rate the **difficulty** (Easy / Intermediate / Hard). (4) Detect the **language**. (5) Extract 2–4 specific **topics** (keywords related to privacy content).

If the instruction contains **personal-sensitive information**, do NOT reveal or restate that information.

Output tags in strict order: Domain → Task Type → Difficulty → Language → Topic(s). Your response must strictly follow this JSON format: [{"tag": str, "explanation": str}].

Table 7: GPT-4o prompt template for annotating PII-related instruction tags.

Frequency filtering: Remove samples with tag values below a frequency threshold to avoid long-tail noise. (3) **Completeness filtering**: Ensure each sample contains all five facet annotations. After filtering, 23,271 high-quality samples are retained.

B.4 Preference Optimization Data

From an initial pool of 17,690 samples, we evaluate using the SFT-trained model and filter out samples with accuracy above 85% (too easy) or below 30% (too noisy), retaining 235 samples for DPO/GRPO training.

C Training Details

To mitigate template rigidity, we rewrite each sample into multiple instruction formats covering four template categories: (1) **Open generation**: Open-ended multi-facet tag generation. (2) **Tag selection**: Selection from a candidate list. (3) **Zero-shot**: Direct inference without examples. (4) **Few-shot**: In-context learning with demonstrations. Each category contains multiple template variants with different instruction phrasings, facet combinations, and output formats (e.g., with or without explanations). In total, we design diverse template variants across the four categories. Below we present one representative example from each category to illustrate the key characteristics.

Open Generation Template. The model generates complete multi-facet tags without candidate list constraints.

You are a helpful assistant. For the query below, generate tags in this order: 1) Domain, 2) Task Type, 3) Difficulty, 4) Language, 5) Topics (can be multiple). Explain each tag briefly. Output must be JSON: {"tag": str, "explanation": str}.

Query: {prompt}

Table 8: Open generation template example.

Tag Selection Template. The model selects the best matching tag from a given candidate list.

You are a helpful assistant. Please identify the single best tag **from the following list** that matches the user query. Provide an explanation for your choice. Please respond in the JSON format {"tag": str, "explanation": str}.

Tag list
{candidate_list}

Query: {prompt}

Table 9: Tag selection template example.

Zero-shot Template. Only one example is provided as a format reference, and the model must perform direct inference.

{examples}

Input: {prompt}
Output:

Table 10: Zero-shot template example.

Few-shot Template. Multiple examples are provided as in-context learning references.

By randomly selecting different template types during training, the model learns to perform the same tagging task across multiple instruction formats, thereby maintaining robustness to template variations at inference time.

D Evaluation Protocol

D.1 Ground-Truth Generation and Evaluation Pipeline

For all four evaluation datasets, we follow a unified annotation generation and evaluation pipeline as illustrated in Figure 4.

Ground-Truth Generation. We use GPT-4o as the annotator with a multi-perspective tagging prompt. The prompt instructs the model to generate high-level semantic tags that capture the key heuristics of the input text, emphasizing that tags should be abstract semantic labels useful for categorization or routing decisions, rather than keywords or entities extracted from the text. Raw annotations from GPT-4o are processed through a four-stage normalization pipeline adapted from InsTag: (i) *Frequency Filtering*, which removes tags that appear fewer than 2 times across the dataset; (ii)

{task_description}

{examples}

Now extract:
Input: {prompt}
Output:

Table 11: Few-shot template example.

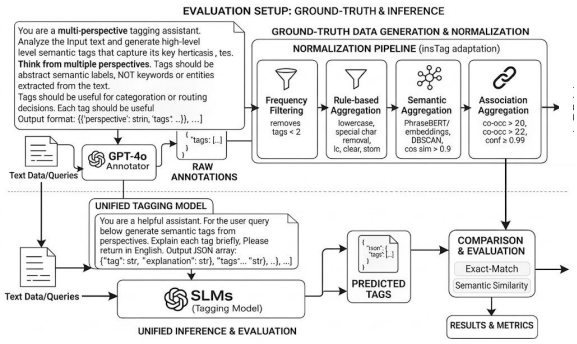

Figure 4: Evaluation pipeline for ground-truth generation, normalization, inference, and comparison.

Rule-based Aggregation, which applies lowercase conversion, special character removal, and stemming to standardize tag surface forms; (iii) *Semantic Aggregation*, which computes PhraseBERT embeddings for all tags and then clusters them using DBSCAN with a cosine similarity threshold greater than 0.9 to merge semantically equivalent tags; and (iv) *Association Aggregation*, which merges highly co-occurring tag pairs (co-occurrence > 20, confidence ≥ 0.99) into canonical forms.

You are a helpful assistant. For the user query below, generate semantic tags from multiple perspectives. Explain each tag briefly. Please return in English. Output JSON array: [{"tag": str, "explanation": str}, ...]

Query: {truncated_context}

Table 12: Unified prompt template for open generation evaluation.

D.2 Inference Prompts

All models are evaluated using identical prompts to ensure fair comparison. For open generation evaluation, we use a unified prompt that instructs the model to generate multi-perspective semantic tags with explanations (Table 12). For tag selection evaluation, we provide dataset-specific candidate lists: InfiniteBench uses 12 task-oriented tags covering code, QA, math, and retrieval tasks (Table 13), while HELMET uses 7 tags corresponding to its task categories (Table 14).

E Ablations

You are a helpful assistant. Please identify the best tag from the list below that matches the user query. Provide an explanation for your choice. Respond in JSON: {"tag": str, "explanation": str}.

Tag list
- Tag name: Code Debugging, Description: Asking for which function in code causes an error.
- Tag name: Code Programming, Description: Asking for the return value in code.
- Tag name: English Multiple Choice, Description: Multiple choice questions using English.
- Tag name: English Question Answering, Description: Question answering using English.
- Tag name: Chinese Question Answering, Description: Question answering using Chinese.
- Tag name: Summarization, Description: Summarization of a text.
- Tag name: Character Identification, Description: Identify speakers in dialogue.
- Tag name: Math Calculation, Description: Asking for results from mathematical calculations.
- Tag name: Math Finding, Description: Find the integer in a list.
- Tag name: Number Retrieval, Description: Finding a number from repeated text.
- Tag name: PassKey Retrieval, Description: Asking for what is a Pass Key in a text.
- Tag name: Key Value Retrieval, Description: Finding the corresponding value from a JSON dictionary.

Query: {truncated_context}

Table 13: Tag selection prompt template for InfiniteBench evaluation.

You are a helpful assistant. Please identify the best tag from the list below that matches the user query. Provide an explanation for your choice. Respond in JSON: {"tag": str, "explanation": str}.

Tag list
- Tag name: Synthetic recall, Description: Identify the words that appear most frequently in a long synthetic list.
- Tag name: Retrieval augmented generation, Description: Question answering with retrieved passages.
- Tag name: Generation with citations, Description: Generate an answer grounded in the provided evidence with citation support.
- Tag name: Passage re-ranking, Description: Rank the retrieved passages by their relevance to the query.
- Tag name: Summarization, Description: Novel and legal document summarization.
- Tag name: Many-shot in-context learning, Description: Classification with many in-context examples.
- Tag name: Long Document Question Answering, Description: Answer the question based on the given documents.

Query: {truncated_context}

Table 14: Tag selection prompt template for HELMET evaluation.

Table 15: Ablation study of auxiliary loss terms in FLEXTAG. We report F1 (%) on open-generation tagging. **Bold** denotes the best result.

Ablations	`InfiniteBench`	`HELMET`	`RouterBench`	`OASST1`
FlexTag w/ DPO [1B]	**20.6**	**19.0**	**31.6**	**28.1**
w/o $\mathcal{L}_{\text{cov.}}$	20.3 ↓0.3	17.9 ↓1.1	30.2 ↓1.4	27.0 ↓1.1
w/o $\mathcal{L}_{\text{syn.}}$	18.4 ↓2.2	17.9 ↓1.1	31.4 ↓0.2	24.7 ↓3.4
w/o $\mathcal{L}_{\text{red.}}$	14.2 ↓6.4	17.2 ↓1.8	31.4 ↓0.2	25.8 ↓2.3
FlexTag w/DPO [4B]	**23.2**	19.1	**31.2**	**28.1**
w/o $\mathcal{L}_{\text{cov.}}$	20.6 ↓2.6	**22.8** ↑3.7	29.2 ↓2.0	27.0 ↓1.1
w/o $\mathcal{L}_{\text{syn.}}$	20.0 ↓3.2	18.1 ↓1.0	30.3 ↓0.9	24.4 ↓3.7
w/o $\mathcal{L}_{\text{red.}}$	19.3 ↓3.9	16.9 ↓2.2	29.5 ↓1.7	25.5 ↓2.6
FlexTag w/GRPO [4B]	**23.7**	19.5	**31.2**	**28.3**
w/o $\mathcal{L}_{\text{cov.}}$	20.6 ↓3.1	**22.9** ↑3.4	29.2 ↓2.0	26.9 ↓1.4
w/o $\mathcal{L}_{\text{syn.}}$	20.2 ↓3.5	18.2 ↓1.3	30.7 ↓0.5	24.3 ↓4.0
w/o $\mathcal{L}_{\text{red.}}$	19.9 ↓3.8	17.4 ↓2.1	29.7 ↓1.5	25.4 ↓2.9