Comprehensive and Efficient Distillation for Lightweight Sentiment Analysis Models

Guangyu Xie[1*], **Yice Zhang**[1*], **Jianzhu Bao**[3,1],
Qianlong Wang[1], **Yang Sun**[1], **Bingbing Wang**[1], **and Ruifeng Xu**[1,2†]
[1] Harbin Institute of Technology, Shenzhen, China
[2] Peng Cheng Laboratory, Shenzhen, China
[3] Nanyang Technological University, Singapore
guangyuxie2001@gmail.com,zhangyc_hit@163.com,xuruifeng@hit.edu.cn

Abstract

Recent efforts leverage knowledge distillation techniques to develop lightweight and practical sentiment analysis models. These methods are grounded in human-written instructions and large-scale user texts. Despite the promising results, two key challenges remain: (1) manually written instructions are limited in diversity and quantity, making them insufficient to ensure comprehensive coverage of distilled knowledge; (2) large-scale user texts incur high computational cost, hindering the practicality of these methods. To this end, we introduce COMPEFFDIST, a comprehensive and efficient distillation framework for sentiment analysis. Our framework consists of two key modules: attribute-based automatic instruction construction and difficulty-based data filtering, which correspondingly tackle the aforementioned challenges. Applying our method across multiple model series (Llama-3, Qwen-3, and Gemma-3), we enable 3B student models to match the performance of 20x larger teacher models on most tasks. In addition, our approach greatly outperforms baseline methods in data efficiency, attaining the same performance level with only 10% of the data. All codes are available at https://github.com/HITSZ-HLT/COMPEFFDIST.

1 Introduction

Recent research shows that large language models (LLMs) possess robust sentiment analysis capabilities. Without requiring task-specific fine-tuning, these models can achieve exceptional performance across various sentiment-related tasks, including polarity determination (Zhang et al., 2024b), emotion recognition (Liu et al., 2024b), sarcasm detection (Yao et al., 2025), and stance detection (Zhang et al., 2024a). Furthermore, these models demonstrate strong abilities to reason and interpret sentiments in complex contexts (Fei et al., 2023; Zhang et al., 2024c).

Despite remarkable performance, the substantial parameter size of LLMs severely constrains their practical application. To address this limitation, extensive research (Zhong et al., 2024; Gu et al., 2024; Ko et al., 2024; Wu et al., 2024; Peng et al., 2024) has focused on leveraging knowledge distillation techniques (Hinton et al., 2015) to transfer knowledge and skills from large teacher models to more compact student models, thereby reducing deployment costs. Among these studies, targeted distillation (Liu et al., 2023; Kim et al., 2024; Zhou et al., 2024) emerges as a particularly promising and practical approach, enabling much smaller models to approximate the capabilities of LLMs across a broad range of applications.

Recent work (Zhang et al., 2025) explores target distillation for sentiment analysis. Their method employs sentiment-related *instructions* and *user texts* to prompt the teacher model, generating a corpus enriched with sentiment knowledge, which is then used to optimize the student model. We argue that the effectiveness of this process critically hinges on the comprehensiveness of instructions and the quantity of user texts. However, these requirements introduce two major challenges: (1) crafting sufficiently comprehensive instructions is labor-intensive, as it requires covering various perspectives for analyzing subjective content, such as polarity, emotion, lexicon, and rhetorical devices; (2) utilizing large-scale user texts incurs substantial computational costs throughout the distillation pipeline, which constrains the practical applicability of distillation-based methods.

To address these two challenges, this paper introduces a comprehensive and efficient distillation framework (COMPEFFDIST) for sentiment analysis. As illustrated in Figure 1, the framework comprises two key modules. The first is attribute-based automatic instruction construction. It iden-

[*] The first two authors contribute equally to this work.
[†] Corresponding Authors

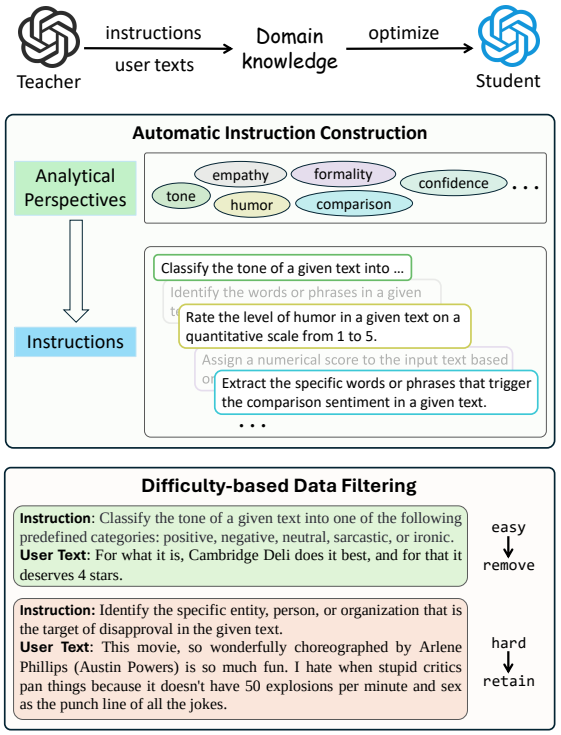

Figure 1: Illustration of our approach: (1) we extract sentiment knowledge from the teacher model through instructions and user texts and then utilize it to optimize the student model; (2) we generate diverse instructions based on various analytical perspectives to ensure comprehensive distillation; (3) we assess the difficulty of instructions and user texts and then reduce the proportion of simple samples to ensure efficient distillation.

tifies and enumerates a wide range of sentiment-related attributes from user texts, applies clustering techniques to group these attributes into distinct analytical perspectives, and subsequently generates diverse instructions based on these analytical perspectives. As such, the module constructs comprehensive instructions without requiring labor-intensive efforts.

The second module, termed difficulty-based data filtering, aims to filter out overly simple data to boost data efficiency. This module is motivated by the hypothesis that simple data contributes minimally to model optimization (Liu et al., 2024a). Specifically, we devise a ranking-based metric that employs the student model to assess the difficulty of instructions and user texts. These difficulty scores are then generalized into a proxy model, enabling efficient scoring. Finally, we apply a difficulty-prioritized sampling strategy to decrease the proportion of simple data, thereby reducing the computational cost of the distillation pipeline.

We conduct experiments across multiple model series (Llama-3, Qwen-3, and Gemma-3) and evaluate their sentiment analysis capabilities using SENTIBENCH (Zhang et al., 2025). The experimental results reveal that: (1) Our approach enables 3B student models to achieve performance on par with teacher models on most tasks, despite being up to 20x smaller in size. (2) Compared to baseline methods, our approach attains the same level of performance using only 10% of the distillation data. These results highlight the effectiveness and promising potential of our approach.

2 Preliminaries: Targeted Distillation

Targeted distillation aims to transfer domain knowledge from a teacher model $\mathcal{T}$ to a student model $\mathcal{S}$. The process generally consists of two stages. The first stage is to extract domain knowledge from the teacher model. Existing methods (Xu et al., 2023; Zhang et al., 2024d; Kim et al., 2024; Zhou et al., 2024) typically utilize a large collection of instruction-user text pairs (ins, x) to prompt the teacher model, generating responses $\hat{y}$:

$$\hat{y} \sim \mathcal{T}(y \mid ins, x). \tag{1}$$

The resulting triples $(ins, x, \hat{y})$ are considered to encode rich domain knowledge. In the second stage, the student model is fine-tuned on these triples using the language modeling objective, formulated as:

$$\max \sum_{ins,x,\hat{y}} \mathcal{S}(\hat{y} \mid ins, x). \tag{2}$$

Challenges. The effectiveness of the aforementioned process critically relies on the quality of the instructions and user texts employed. To ensure comprehensive coverage of distilled knowledge, the instruction set must capture a wide range of relevant perspectives. In sentiment analysis, these perspectives refer to various dimensions for analyzing subjective content. They may include sentiment polarity, emotion types, linguistic expressions, as well as higher-level aspects like rhetorical devices and contextual background. However, manually summarizing all such perspectives and crafting the corresponding instructions is extremely challenging and often impractical.

Simultaneously, the user text collection must span a broad spectrum of contextual variations. For example, in terms of linguistic expressions, it should include diverse patterns such as explicit

sentiment words, factual statements, comparisons, metaphors, and sarcastic utterances. Achieving such diversity typically depends on the size and richness of the user text corpus, consequently increasing the computational demands of teacher model prompting and student model optimization.

3 Comprehensive & Efficient Distillation

We introduce a comprehensive and efficient distillation framework (CompEffDist) to address the challenges of previous methods. The framework consists of two modules: (i) attribute-based automatic instruction construction, which generates diverse instructions to ensure comprehensive distillation; and (ii) difficulty-based data filtering, aiming to reduce the proportion of simple data during distillation, thereby enhancing data efficiency.

3.1 Attribute-based Automatic Instruction Construction

As discussed in Section 2, it is essential for instructions to cover a diverse range of analytical perspectives. Inspired by Lou et al. (2024), we adopt an approach in which the teacher model identifies sentiment-related attributes from user texts, which are then used to generate specific instructions. As illustrated in Figure 2, the process comprises four steps: attribute enumeration, attribute clustering, task generation, and instruction generation.

Attribute Enumeration. We observe that real-world user texts inherently contain a sufficiently diverse range of sentiment-related attributes. Consider the following example:

> "I wish I could give it zero stars. Whoever thinks this smells like a lemon, needs help. This is the most disgusting, repulsive, overwhelming, stinky cleaner I ever had the displeasure of using."

This text exhibits a high degree of emotional intensity and conveys strong negative sentiments such as frustration and disgust. Besides, it employs sarcasm[1] as a rhetorical device within the context of product dissatisfaction. In summary, this example involves the following attributes: emotional intensity, frustration, disgust, sarcasm, and product dissatisfaction.

[1]The rationale of sarcasm is that "need help" is not a genuine suggestion to seek medical attention, but rather a sarcastic critique of the other person's absurd judgment.

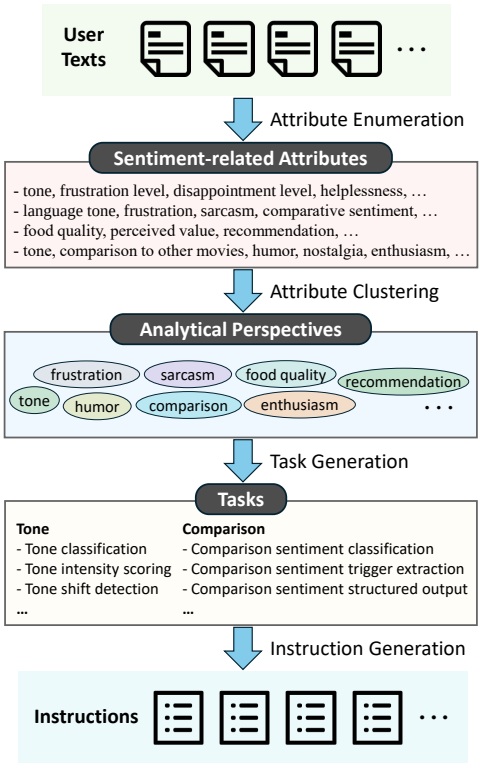

Figure 2: Illustration of attribute-based automatic instruction construction.

Building on the above observation, we prompt the teacher model to identify and enumerate sentiment-related attributes present in user texts. A total of 20K user texts are used in this step. After normalizing the collected attributes, we obtain approximately 1,800 distinct attributes. The complete prompt and implementation details are provided in Appendix A.1.

Attribute Clustering. Many attributes are semantically equivalent, but differ only in form or phrasing. For example, terms such as 'tone', 'language tone', 'tone of language', and 'tone of voice' all convey the same underlying meaning. We therefore employ clustering techniques to consolidate semantically similar attributes. Specifically, we first employ `UAE-Large-V1`[2] (Li and Li, 2024), an embedding model, to project the textual attributes into vector representations. We then apply the affinity propagation algorithm (Frey and Dueck, 2007) to cluster these representations.

We ultimately obtain 180 clusters. Moving forward, we refer to these clusters as analytical perspectives and use the most frequently occurring attribute within each cluster as its name. The repre-

[2]Available at https://huggingface.co/WhereIsAI/UAE-Large-V1.

sentative analytical perspectives include tone, comparison, humor, and food quality, covering expression styles, linguistic phenomena, and concrete aspects. A detailed presentation and analysis can be found in Section 4.

Task & Instruction Generation. For each analytical perspective, we prompt the teacher model to brainstorm a series of tasks, where each task consists of a task name and a brief description. To guide the task generation process, we provide the teacher model with predefined task types, including classification, regression, extraction, structured output, and open-ended generation. For example, under the analytical perspective of 'tone', the generated tasks comprise: (i) tone classification, (ii) tone intensity scoring, (iii) tone shift detection, (iv) tone comparison to neutral, (v) tone-related entity extraction, (vi) tone profiling, and (vii) tone-based summarization.

Subsequently, we instruct the teacher model to synthesize complete instructions by enriching the task description, incorporating specific requirements, and generating a set of demonstrations. During the demonstration generation process, we make a special effort to balance the class distribution. Detailed implementation is described in Appendix A.2. Figure 3 presents an example of the resulting instruction.

Task Description: Create a tone profile for the given text by analyzing its tone category, intensity, and emotional undertones.

Requirements: Represent the tone profile in a structured JSON format with the following fields:

* `tone_category`: a string label representing the tone category (e.g., "positive", "negative", "neutral", "sarcastic", etc.)
* `intensity`: a numerical score between 0 and 1 representing the intensity of the tone
* `emotional_undertones`: a list of strings representing the emotional undertones (e.g., ["anger", "fear", "joy", etc.])

The tone profile should be represented in the following JSON format:
```
{
  "tone_category": "",
  "intensity":,
  "emotional_undertones": []
}
```

Demonstrations:
Input: u think u can just slide into my dms and ask for my wifi password? no way
Output:
```
{
  "tone_category": "sarcastic",
  "intensity": 0.7,
  "emotional_undertones": ["irritation", "defensiveness"]
}
```

Input: {Input Text}
Output:

Figure 3: Generated instruction for tone profiling.

After obtaining a large and diverse set of instructions, we randomly pair them with user texts to construct a comprehensive collection of instruction-user text pairs. These pairs are then used to prompt the teacher model for response generation. An alternative approach is to match instructions with user texts based on their associated attributes. In our experiments, we find that both methods achieve comparable performance. Detailed results are presented in Appendix D.

3.2 Difficulty-based Data Filtering

As mentioned in Section 2, targeted distillation requires a large amount of user texts to ensure effectiveness, increasing the computational cost. To address this challenge, we assess the difficulty of instruction-user text pairs and adopt a difficulty-prioritized sampling strategy to reduce the proportion of simple data. In terms of difficulty assessment, we first evaluate each pair through a ranking-based metric. The resulting difficulty scores are optionally used to train a proxy model that enables more efficient difficulty estimation.

Ranking-based Difficulty Metric. We evaluate the difficulty of a triplet (ins, x, y) by assessing how well the student model $\mathcal{S}$ can reproduce the response y given the input (ins, x). Most existing methods (Xie et al., 2024; Li et al., 2024) rely on perplexity-based metrics for this purpose:

$$\text{PPL} = -\frac{1}{|y|}\sum_{t=1}^{|y|}\log P_{\mathcal{S}}(y_t \mid ins, x, y_{<t}). \quad (3)$$

However, perplexity is unsuitable for evaluating sentiment analysis samples. This is because sentiment analysis tasks typically involve categorical outputs rather than free-form text generation, and full-vocabulary probability distributions do not provide reliable indicators of correctness. Therefore, we devise a ranking-based scoring metric.

We first curate a small subset of the distillation data to warm up the student model. Subsequently, this warmed-up model is used to assess the difficulty of a given triplet (ins, x, y). For each token y_t, we estimate the size of the relevant label space using top-p sampling, denoted as N_t. We then determine the ranking position of y_t within the top-p distribution, denoted as r_t. Based on these values, we define the difficulty score for y_t as follows:

$$d(y_t) = \begin{cases} \dfrac{r_t - 1}{N_t} & \text{if } r_t \le N_t, \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

The overall difficulty of the response y is computed as the average of the token-level difficulty scores. However, to avoid bias from format tokens (e.g.,

punctuation marks such as [or "), we exclude tokens whose scores fall below a threshold ε_d during the averaging process. Detailed implementations are presented in Appendix A.3.1.

Proxy Model. The aforementioned difficulty metric requires access to the teacher response y, implying that this method can only reduce the optimization cost of the student model while leaving the prompting cost of the teacher model unaffected. We therefore explore training a proxy model that does not require teacher responses, thereby allowing for data filtering before the teacher model prompting. The proxy model $\mathcal{P}$ is an autoregressive model with a regression head. It takes an instruction ins and a user text x as input and outputs a difficulty score $\hat{d}$:

$$\hat{d} = \mathcal{P}(ins, x). \tag{5}$$

We optimize the proxy model using the mean-squared error (MSE) loss, formulated as:

$$\mathcal{L} = \mathrm{MSE}(\hat{d}, d), \tag{6}$$

where d denotes the score derived from the ranking-based metric.

While using the proxy model can reduce both the teacher model's prompting cost and the student model's optimization cost, its performance is inferior due to the absence of teacher responses. Detailed experimental results are presented in Section 5.4. Consequently, the proxy model is treated as an optional component within the data filtering module.

Difficulty-Prioritized Sampling Strategy prioritizes more challenging samples while reducing the proportion of easily-learned samples. For a group of M samples sharing the same instruction, we estimate their difficulty scores using the ranking-based metric or the proxy model. These samples are then sorted in ascending order of their scores, and each is assigned a sampling probability of $\frac{\rho-0.5}{M}$, where ρ denotes its rank. We perform stochastic sampling based on these probabilities. As a result, 50% of the samples are expected to be retained. We also explore two variants: (i) global sampling, which ranks samples across all instructions based on their difficulty scores and adopts a unified sampling procedure; and (ii) hard-only sampling, which retains only the most difficult 50% of samples for each instruction. Both variants yield suboptimal performance. Detailed results and analysis are provided in Section 5.4.

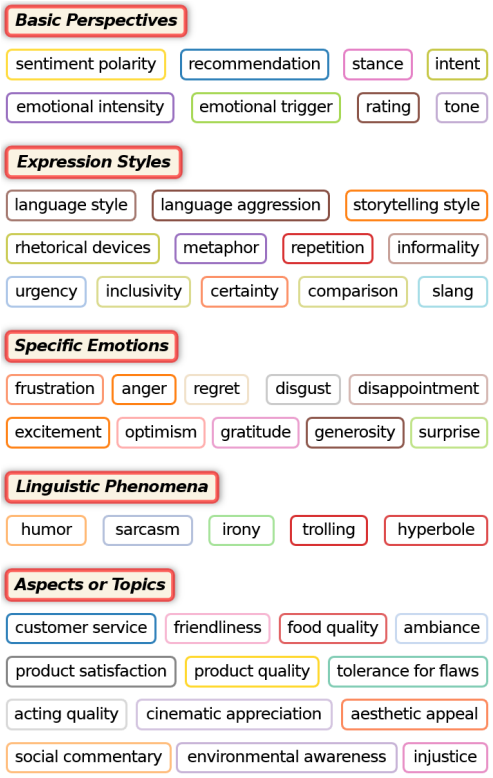

Figure 4: Visualization of the representative analytical perspectives. A more complete visualization is provided in Figure 9 of Appendix B.

4 Data Analysis

Visualization of Analytical Perspectives. The generated analytical perspectives span a broad semantic range. As shown in Figure 4, they can be broadly categorized into five main categories: (i) basic analytical perspectives, such as sentiment polarity, emotional intensity, and tone; (ii) expression styles, such as language style, and rhetorical devices; (iii) specific emotions, such as frustration, anger, excitement, and optimism; (iv) linguistic phenomena, such as humor, sarcasm, and trolling; and (v) concrete aspects or topics, such as customer service, food quality, acting quality, and social commentary. Moreover, a certain degree of overlap or nesting among analytical perspectives can also be observed. For example, 'product quality' is a sub-aspect of 'product satisfaction'. We believe that this level of redundancy is potentially beneficial, as it enhances the comprehensiveness of the perspective set and supports the generation of more diverse tasks.

Data Statistic. We obtain a total of 3,707 tasks and 50K samples. Figure 5 presents the distribution of task types. There are five task categories, with

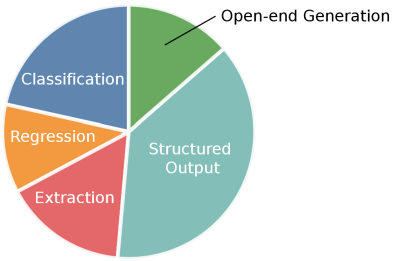

Figure 5: Type distribution of generated tasks.

structured-output tasks comprising a notably high proportion. This is because LLMs tend to generate complex and composite tasks. We also offer the length distribution of instructions, user texts, and responses in Figure 8 of Appendix B.

5 Experiments

5.1 Experimental Setup

Implementation Details. We conduct extensive experiments across multiple LLM series, namely Llama-3 (Grattafiori et al., 2024), Qwen-3 (Yang et al., 2025), and Gemma-3 (Kamath et al., 2025). The specific teacher-student setups are (Llama-3.1-70B-instruct, Llama-3.2-3B-instruct), (Qwen-3-32B, Qwen-3-4B), and (Gemma-3-27B-it, Gemma-3-4B-it). If not specified, analytical experiments are conducted on the Llama-3 series models.

The user text corpus for the distillation process is collected from IMDb, Yelp, Amazon, and Twitter. We use 20K user texts to generate instructions, resulting in 3,707 distinct instructions. An additional 100K user texts are then paired with these instructions to form 100K instruction-user text pairs. These pairs will be filtered based on difficulty, retaining only 50% of them. The remaining pairs are used for distillation. Hyperparameter settings are detailed in Appendix A.4. After distillation, we evaluate the student models on SENTIBENCH (Zhang et al., 2025), with dataset statistics presented in Appendix C.

Baselines. We select three categories of baselines for comparison. The first is two generic distillation methods that fine-tune the student model using Alpaca-data (Taori et al., 2023) and Lamini-data (Wu et al., 2024). The second is KNOW & ICLDIST (Zhang et al., 2025), a two-stage approach leveraging manually written instructions to distill knowledge for sentiment analysis. The third is EmoLlama (Liu et al., 2024b), which fine-tunes Llama models using a multi-task affective analysis instruction dataset. In addition, we provide results of GPT-3.5 and GPT-4o[3] as reference.

5.2 Main Results

Table 1 presents the comparison results between our approach and the baseline methods. These results suggest that our approach enables the student model to attain performance close to that of the teacher model on most tasks and consistently outperforms all baseline methods, demonstrating its effectiveness.

Furthermore, we make the following observations. Firstly, the performance gains from the two generic distillation methods are minimal, indicating their inefficiency in transferring specific capabilities. Secondly, our approach surpasses the previous method relying on manually constructed instructions (*i.e.*, KNOW & ICLDIST). This improvement demonstrates that our approach reduces human effort while achieving superior performance. Thirdly, there is a notable performance gap between the student and teacher models on the FSA tasks, and our approach successfully narrows this gap. We attribute this improvement to the diverse structured output tasks in the automatically generated instruction set. Finally, the performance gains on Qwen-3 and Gemma-3 are relatively modest compared to those observed on Llama-3. This can be attributed to their extensive use of knowledge distillation during training process (Yang et al., 2025; Kamath et al., 2025). As these models already incorporate sophisticated distillation techniques in their training pipelines, the incremental benefits of our method are naturally reduced.

5.3 Effect of Instruction Comprehensiveness

One of the key claims of this work is that the diversity of instruction sets is a critical factor in ensuring effective distillation. We conduct exploratory experiments to assess the impact of instruction variety. Results in Figure 6 show that the student model performance gradually improves as the variety of instructions increases, providing strong support for our hypothesis. Besides, we compare our instruction set with previous work under different data budgets. As presented in Table 2, our approach achieves results comparable to the prior method using only 20K samples—less than 10% of 300K samples used in the prior method. This substan-

[3]Available at https://chat.openai.com/. The specific models used are `gpt-3.5-turbo-0125` and `gpt-4o-2024-11-20`.

Models	BSA				MSA				FSA				Avg
	IMDb	Yelp2	SST2	Twitter	Irony	Emoti.	Stance	Intim.	ATSA	ACSA	ASQP	SSA	
EmoLlama-chat-7B	91.27	97.10	93.46	64.84	70.28	70.48	74.52	41.77	40.19	50.31	20.20	23.18	61.47
EmoLlama-chat-13B	93.63	97.87	94.31	59.87	65.94	70.75	73.65	45.28	48.31	58.92	23.07	30.83	63.53
EmoLlama-3-3B	90.86	96.50	91.96	66.25	71.51	74.88	74.33	48.55	47.09	48.72	16.55	22.84	62.50
GPT-3.5	93.70	98.30	96.31	60.15	78.64	75.61	79.99	52.63	56.43	66.67	30.30	44.01	69.40
GPT-4o	93.91	98.18	97.13	70.84	77.58	76.66	85.22	53.29	56.72	72.64	34.75	51.46	72.37
Llama-3-70B	95.30	98.10	97.14	68.75	83.99	75.87	85.21	53.68	59.48	74.13	32.14	50.11	72.83
Llama-3-8B	94.17	98.07	95.90	66.58	82.63	73.00	75.86	49.85	59.00	65.53	23.49	34.90	68.25
+ **CompEffDist (Ours)**	93.56	98.07	96.23	68.81	85.89	74.56	82.35	53.36	63.01	70.08	30.37	45.57	71.82(+3.57)
Llama-3-3B	92.57	96.53	93.59	61.45	64.00	68.88	71.43	33.32	52.74	53.23	14.33	23.56	60.47
+ Distill. *w/* Alpaca-data	92.37	97.37	93.92	57.70	66.59	64.47	72.05	28.70	50.96	54.11	21.16	28.58	60.66(+0.19)
+ Distill. *w/* Lamini-data	92.80	97.33	94.91	62.07	70.10	65.61	72.49	40.28	53.73	56.46	19.99	27.25	62.59(+2.12)
+ KNOW & ICLDIST	94.30	98.17	95.41	69.57	85.25	77.47	75.10	48.24	53.07	65.22	24.61	36.17	68.55(+8.08)
+ **CompEffDist (Ours)**	93.67	97.12	94.78	68.17	82.86	76.29	78.77	54.32	58.17	67.22	31.34	35.98	69.89(+9.42)
Qwen-3-32B	94.37	97.87	94.86	62.80	83.06	73.12	79.28	52.87	60.36	72.94	33.80	49.51	71.24
Qwen-3-4B	91.87	97.70	94.45	69.46	79.51	68.27	73.09	48.17	57.43	67.55	28.16	42.95	68.22
+ Distill. *w/* Alpaca-data	92.90	97.93	94.76	68.20	82.15	69.81	74.10	45.63	57.30	66.92	29.08	44.22	68.58(+0.36)
+ Distill. *w/* Lamini-data	92.43	98.17	94.16	68.50	84.02	64.78	72.89	50.43	56.11	66.10	28.93	38.21	67.89(-0.33)
+ KNOW & ICLDIST	93.40	98.03	95.82	69.07	82.23	75.53	76.87	50.49	59.21	71.31	30.44	37.81	70.02(+1.80)
+ **CompEffDist (Ours)**	92.57	97.83	94.67	67.98	84.86	73.09	77.08	54.03	60.05	71.47	32.24	42.46	70.69(+2.47)
Gemma-3-27B	93.57	98.27	96.80	68.68	82.70	75.59	83.25	61.28	62.89	73.72	33.00	54.00	73.64
Gemma-3-4B	92.10	97.00	93.81	62.20	63.00	73.30	75.68	51.37	55.01	61.01	23.74	44.97	66.10
+ Distill. *w/* Alpaca-data	92.50	97.40	94.09	60.09	75.83	74.46	74.99	44.60	54.34	64.08	24.94	41.98	66.61(+0.51)
+ Distill. *w/* Lamini-data	93.40	97.73	94.91	63.58	80.37	71.93	75.22	49.99	55.03	63.43	20.27	38.04	66.99(+0.89)
+ KNOW & ICLDIST	93.10	97.87	95.04	68.12	72.52	75.41	76.83	53.82	58.02	69.82	32.91	38.93	69.37(+3.27)
+ **CompEffDist (Ours)**	92.19	97.33	94.45	64.34	78.56	75.30	77.72	54.49	57.45	66.60	26.53	48.59	69.46(+3.36)

Table 1: Comparison results on SENTIBENCH (F_1-score, %). BSA, MSA, and FSA denote basic sentiment analysis, multi-faceted sentiment analysis, and fine-grained sentiment analysis, respectively. KNOW & ICLDIST is trained using 300K samples, while our method uses 50K samples. EmoLlama-3-3B refers to a Llama-3-3B model fine-tuned on the same instruction dataset as EmoLlama model.

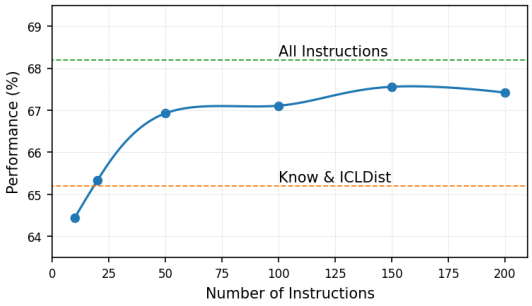

Figure 6: Performance trend of the student model with varying numbers of instructions (%). The distillation dataset size is 20K, and data filtering is not applied.

tial reduction in data requirements highlights that the comprehensiveness of the instruction set also enhances the overall efficiency of the distillation process.

Models	Avg-F1	Δ
Llama-3-3B	60.47	-
+ KNOW & ICLDIST (300K)	68.55	+8.08
+ OURS (20K)	68.19	+7.72
+ OURS (100K)	70.17	+9.70

Table 2: Performance comparison between our instruction set and that of the previous method (%).

5.4 Analysis of Data Filtering

Figure 7 illustrates the impact of data quantity on student model performance. As expected, performance improves steadily with increasing data size, highlighting the importance of sufficient distillation data. Moreover, our data filtering methods substantially enhance data efficiency, such that using only 50K filtered data can match the performance of 90K original data. This result demonstrates the effectiveness of the proposed filtering methods. Furthermore, the proxy model performs worse than direct

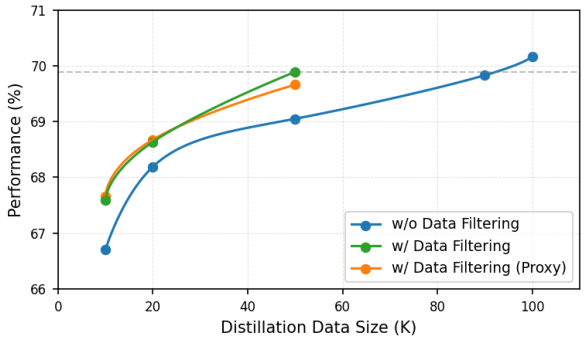

Figure 7: Performance trend of the student model with increasing data quantity (%).

scoring. However, it reduces both the prompting cost of the teacher model and the optimization cost of the student model.

Filtering Methods	Avg-F1	Δ
OURS	69.89	-
Difficulty Metric Ablations		
Perplexity	69.09	-0.80
IFD	69.08	-0.81
User Text Length	69.21	-0.68
Sampling Strategy Ablations		
Global Sampling	69.29	-0.60
Hard-only Sampling	69.23	-0.69

Table 3: Ablation studies of data filtering methods (%). For difficulty metric ablations, we use difficulty-prioritized sampling strategy. For sampling strategy ablations, we use the ranking-based difficulty metric.

We conduct ablation experiments on the filtering methods. Firstly, we compare different metrics for difficulty assessment: perplexity, IFD (Li et al., 2024), and text length. Results in Table 3 show that these metrics perform worse than our ranking-based metric. Secondly, we explore variants of our sampling strategy. We find that both global sampling and hard-only sampling result in suboptimal performance. We attribute the poor performance of global sampling to the large variation in instruction difficulty, which can lead to over-selection of certain instruction types and thus reduce diversity. As for hard-only sampling, we believe that restricting the distillation data to only difficult samples hinders the learning of the student model.

5.5 Further Analysis

We have the following further analyses in the Appendix:

- Analysis of Instruction-User Text Pairing.
- Data Filtering on Other Baselines.
- Results on Complex Contexts.
- Case Study of Difficulty Assessment.

6 Related Work

Targeted Distillation. Knowledge distillation techniques have been widely applied to develop more accessible and compact models (Taori et al., 2023; Chiang et al., 2023; Wu et al., 2024). Targeted distillation, which focuses on transferring LLMs' capabilities in specific applications, has recently gained significant attention. Existing methods can be broadly categorized into two paradigms. The first (Ding et al., 2023; He et al., 2024; Xu et al., 2023; Zhou et al., 2024) treats the LLM as an annotator, generating large-scale task-specific pseudo-labels for training a smaller model. This method typically employs limited instructions and is effective mainly for narrowly defined tasks. The second (Zhang et al., 2025; Kim et al., 2024) constructs a broader set of instructions to transfer LLMs' capabilities across a targeted domain. While offering stronger effectiveness and broader generalization, it also imposes higher demands on the quantity and diversity of instructions.

Instruction Generation has emerged as a key research direction, due to its critical role in improving the coverage of distilled knowledge. Existing methods can be broadly categorized into two types. The first (Wang et al., 2023; Honovich et al., 2023; Xu et al., 2024) adopts a bootstrap strategy, generating new instructions based on existing ones. However, this method requires a large seed instruction set and often suffers from limited diversity. The second is attribute-based methods (Wu et al., 2024; Lou et al., 2024), generating instructions by specifying topics, entities, or text segments. Its main challenge lies in developing a high-quality and diverse attribute set. To address this, we identify a large number of attributes from user texts and employ clustering algorithms to group them into meaningful analytical perspectives.

Data Selection has been extensively studied, especially as model sizes continue to grow, leading to prohibitively high fine-tuning and inference costs. The main criteria guiding data selection include diversity, quality, and difficulty. A few studies explore manual curation of instruction data (Köpf et al., 2023; Zhou et al., 2023), but such methods are labor-intensive and less scalable. More recent

efforts have therefore focused on automatic selection methods. For diversity, techniques such as vocabulary coverage, semantic tagging, and clustering are employed (Cao et al., 2024; Lu et al., 2024; Ge et al., 2024). For quality, filtering based on advanced LLMs is a common practice (Chen et al., 2024; Lian et al., 2023). For difficulty, most existing methods rely on the student model's uncertainty, with ongoing efforts aimed at developing more robust and reliable difficulty metrics (Li et al., 2024; Kung et al., 2023). In this paper, we highlight that current difficulty metrics are not well-suited for sentiment analysis tasks. To address this, we propose a ranking-based metric.

7 Conclusions

To develop lightweight sentiment analysis models, we introduce CompEffDist, a comprehensive and efficient distillation framework. This framework automatically generates a large and diverse set of instructions via an attribute-based method and applies difficulty-based data filtering to boost data efficiency. Leveraging this framework, we construct a dataset containing 3,707 distinct tasks and 50K samples. Applying it to knowledge distillation, we enable 3B student models to achieve performance comparable to that of 20x larger teacher models on most tasks. Furthermore, our approach attains results on par with baseline methods using only 10% of the data, demonstrating its superior data efficiency.

Limitations

We discuss potential limitations of this work:

- CompEffDist does not include task-level deduplication or filtering operations. The large number of generated tasks inevitably contains overlaps and some low-quality instances. Introducing task-level deduplication and quality-based filtering could increase the proportion of high-quality, long-tail tasks, thereby improving data efficiency in the distillation process. However, identifying task overlaps and assessing instruction quality remain challenging.

- CompEffDist does not incorporate quality control for the teacher model's responses. Teacher models can generate incorrect or biased outputs, which can be transferred to the student model and affect its performance. Incorporating quality assurance techniques, such as reflection, reasoning, or consistency checks, has the potential to improve the effectiveness and reliability of knowledge distillation. However, this would also introduce additional computational costs. Balancing the trade-off between cost and performance improvement is an important direction for future research.

We believe that these limitations point to promising directions for future research.

Ethics Statement

Large language models for sentiment analysis have enabled progress in areas such as public health and commercial applications; yet their reliance on large-scale pretraining corpora raises ethical concerns, including risks of privacy violations, cultural and annotator subjectivity, and systematic harms to marginalized groups (Mohammad, 2021). While knowledge distillation substantially improves efficiency and deployability, prior work shows that it can also transfer and intensify existing biases, exacerbating disparities across sentiment classes and demographic subgroups.

Accordingly, ethical evaluation of distilled sentiment models should not only emphasize improvements in overall performance but also recognize the risks of propagating biases and exacerbating disparities across categories and social subgroups (Sabbagh et al., 2025). Therefore, the community should place greater emphasis on assessing subgroup- and category-level fairness, accompanied by clearer documentation of risks and limitations. In addition, exploring fairness-aware distillation methods and developing practical guidelines could help mitigate potential misuse in sensitive or high-stakes applications.

Acknowledgments

This work was supported by the National Natural Science Foundation of China 62176076 and 62576120, Natural Science Foundation of Guang Dong 2023A1515012922, the Major Key Project of PCL2023A09, CIPSC-SMP-ZHIPU Large Model Cross-Disciplinary Fund ZPCG20241119405 and Key Laboratory of Computing Power Network and Information Security, and Ministry of Education under Grant No.2024ZD020.

References

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2024. Instruction mining: Instruction data selection for tuning large language models. In *First Conference on Language Modeling*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. Reasoning implicit sentiment with chain-of-thought prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1171–1182, Toronto, Canada. Association for Computational Linguistics.

Brendan J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Mahong Xia, Zhang Li, Boxing Chen, Hao Yang, Bei Li, Tong Xiao, and JingBo Zhu. 2024. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 464–478, Miami, Florida, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit San-

gani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. AnnoLLM: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 165–190, Mexico City, Mexico. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

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

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. Distillm: towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Po-Nien Kung, Fan Yin, Di Wu, Kai-Wei Chang, and Nanyun Peng. 2023. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations – democratizing large language model alignment. *Preprint*, arXiv:2304.07327.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7595–7628, Mexico City, Mexico. Association for Computational Linguistics.

Xianming Li and Jing Li. 2024. AoE: Angle-optimized embeddings for semantic textual similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839, Bangkok, Thailand. Association for Computational Linguistics.

Wing Lian, Guan Wang, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification.

Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. 2023. Tinygsm: achieving >80% on gsm8k with small language models. *Preprint*, arXiv:2312.09241.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024a. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Zhiwei Liu, Kailai Yang, Qianqian Xie, Tianlin Zhang, and Sophia Ananiadou. 2024b. Emollms: A series of emotional large language models and annotation tools for comprehensive affective analysis. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5487–5496, New York, NY, USA. Association for Computing Machinery.

Renze Lou, Kai Zhang, Jian Xie, Yuxuan Sun, Janice Ahn, Hanzi Xu, Yu Su, and Wenpeng Yin. 2024. MUFFIN: Curating multi-faceted instructions for improving instruction following. In *The Twelfth International Conference on Learning Representations*.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2024. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *The Twelfth International Conference on Learning Representations*.

Saif M. Mohammad. 2021. Ethics sheet for automatic emotion recognition and sentiment analysis. *CoRR*, abs/2109.08256.

Hao Peng, Xin Lv, Yushi Bai, Zijun Yao, Jiajie Zhang, Lei Hou, and Juanzi Li. 2024. Pre-training distillation for large language models: A design space exploration. *Preprint*, arXiv:2410.16215.

Kamil Sabbagh, Hadi Salloum, Rafik Hachana, Marko Pezer, and Manuel Mazzara. 2025. Impact of data distillation on fairness in machine learning models. *Preprints*.

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

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

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.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2024. LaMini-LM: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 944–964, St. Julian's, Malta. Association for Computational Linguistics.

Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2024. Efficient continual pre-training for building domain specific large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10184–10201, Bangkok, Thailand. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Yichong Xu, Ruochen Xu, Dan Iter, Yang Liu, Shuohang Wang, Chenguang Zhu, and Michael Zeng. 2023. InheritSumm: A general, versatile and compact summarizer by distilling from GPT. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13879–13892, Singapore. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Ben Yao, Yazhou Zhang, Qiuchi Li, and Jing Qin. 2025. Is sarcasm detection a step-by-step reasoning process in large language models? *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25651–25659.

Bowen Zhang, Daijun Ding, Liwen Jing, Genan Dai, and Nan Yin. 2024a. How would stance detection techniques evolve after the launch of chatgpt? *Preprint*, arXiv:2212.14548.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. 2024b. Sentiment analysis in the era of large language models: A reality check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906, Mexico City, Mexico. Association for Computational Linguistics.

Yice Zhang, Guangyu Xie, Jingjie Lin, Jianzhu Bao, Qianlong Wang, Xi Zeng, and Ruifeng Xu. 2025. Targeted distillation for sentiment analysis. *Preprint*, arXiv:2503.03225.

Yice Zhang, Guangyu Xie, Hongling Xu, Kaiheng Hou, Jianzhu Bao, Qianlong Wang, Shiwei Chen, and

Ruifeng Xu. 2024c. Distilling fine-grained sentiment understanding from large language models. *Preprint*, arXiv:2412.18552.

Yifei Zhang, Bo Pan, Chen Ling, Yuntong Hu, and Liang Zhao. 2024d. ELAD: Explanation-guided large language models active distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4463–4475, Bangkok, Thailand. Association for Computational Linguistics.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. 2024. Revisiting knowledge distillation for autoregressive language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10900–10913, Bangkok, Thailand. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations*.

Organization of Appendices

We organize the appendix into four sections:

- Appendix A presents additional implementation details of our method;
- Appendix B provides more comprehensive data visualization;
- Appendix C describes the evaluation setup and dataset statistics;
- Appendix D offers further analysis of our method.

A Further Implementation Details

A.1 Attribute Enumeration and Clustering

We leverage the teacher model to identify and enumerate sentiment-relevant attributes from user texts. The complete prompt used for this step is shown in Table 4. By parsing the model responses, we obtain a large number of attributes, which are then standardized to construct an attribute pool. Attributes that appear fewer than or equal to 10 times are removed, resulting in a total of 1,785 distinct attributes.

These attributes are subsequently mapped into a vector space using UAE embeddings. We apply affinity propagation clustering to group the vectors. The hyperparameters are set as follows: percentile_rate = 0.5 and damping = 0.9. To incorporate attribute frequency into the clustering process, we first map the frequency counts x using the following transformation:

$$y = 1 + \log(1 + x), \tag{7}$$

and then replicate each attribute y times before performing clustering.

A.2 Task and Instruction Generation

For each analytical perspective, we prompt the teacher model to generate two types of tasks: open-ended generation tasks and constrained tasks. The corresponding prompts are provided in Table 5.

For each constrained task, we further guide the model to synthesize complete instructions by enriching the descriptions and adding specific requirements. The detailed prompt for this step is shown in Table 6. In addition, we generate 32 demonstrations for each task, using the prompting templates

Instruction: Given the following input, what kind of sentiment-related attributes does it have?

Requirements:

1. Please brainstorm as many related attributes as possible.
2. Be creative. Any interesting perspectives are welcome!
3. Each attribute should typically reflect a particular characteristic of the input text.
4. Each attribute should be followed with Attribute Description (a brief description of what the attribute represents) and Attribute Value (the corresponding attribute value as reflected in the text).
5. Feel free to ignore the tedious and specific content. Just focus on some general textual attributes!

Input: {Input Text}

Attribute:

Table 4: The prompt for attribute enumeration.

Open-end Generation Task Generation

Please generate prompts for analyzing subjective texts such as product reviews or social media according to the following rules:
1. Each prompt should capture the core and commonalities of the following attribute categories and without relying on specific attribute: {Perspective}.
   - The explanation for {Attribution1} is {Brief Explaination of Attribution1}.
   - The explanation for {Attribution2} is {Brief Explaination of Attribution2}.
   - The explanation for {Attribution3} is {Brief Explaination of Attribution3}.
   - The explanation for {Attribution4} is {Brief Explaination of Attribution4}.
   - The explanation for {Attribution5} is {Brief Explaination of Attribution5}.
2. Ensure that each prompt is domain-general by using neutral references such as "this text" avoiding any specific domain indications.
3. Each prompt should be designed to help better understand subjective texts by deconstructing it based on the specified attribute categories.
4. Employ diverse strategies, which may include but are not limited to:
   - Open-ended deconstruction instructions
   - Diagnostic questions
5. Ensure that your responses are structured in ordered numbers.

Generated prompt:

Constrained Task Generation

I want you to focus on the following text attribute: **{Perspective}({Brief Explaination of Perspective})**, and systematically generate a diverse range of tasks that target a single text. Please make sure each task includes the following elements:
   - Task Name: a concise title that captures the core goal or theme of the task.
   - Task Description: an explanation of the problem this task aims to solve or the objective it aims to achieve.

The task types should be diverse, such as:
1. Classification
   - Closed-set categories classification
   - Open-ended categories classification
2. Scoring or Rating
   - Quantitative scales
3. Information Extraction
   - Keywords, key sentences, triggers
   - Root causes, contextual dependencies, and more
4. Structured Output
   - JSON, tables, or other machine-readable formats
   - Potentially includes multiple fields (roles, attribute values, etc.)

When designing these tasks, please follow these guidelines:
   - Clarity: Each task's goal should be described methodically.
   - Diversity: Aim for a wide range of creative ideas across classification, scoring, extraction, and extended analyses.
   - All tasks must target a single text. Therefore, do not generate tasks involving comparisons between two texts.

Based on the above requirements, please list several diverse tasks focused on **{Attribution}**.
Present your output in the following structured JSON format, ensuring that it can be directly parsed.

Table 5: The prompts for task generation.

Instruction Generation
Please rewrite the task based on the task name and description, making the task definition more standardized and normalized.

Task Name: {Task Name}
Task Description:{Task Description}

Below are the specific requirements and guidelines:
1. Avoiding Ambiguity: Ensure task description, requirement and constraint is precise, complete, and free of ambiguity. If the task contains two direction, specify one direction in the task description and requirments and you should NOT add any requirments in input.

2. Ensure the rewritten task is consistent with the original task description.

3. Task Elements: Ensure that each task definition includes the following components:
- Task Name: A concise title of the task.
- Task Description: A detailed explanation of the task and should contain the following parts:
    - Explicitly specifying the expected output format and requirements (e.g., classification label, numerical score, structured JSON, Python list).
    - If the task is a classification task or contains classification task as subtask, for closed-set classification, you should explicitly list all allowed labels. For open-set classification, you should instruct the model to infer the appropriate labels from the input.
    - If the task is a annotation/extraction task, you should specify whether the extracted or annotated text must exactly match the original text or if modifications are allowed.
    - If the task requires structured output, specify the exact structure (for example, a JSON schema or Python list format) and enumerate all required fields.
- Task Examples: You should provide at least EIGHT concrete examples, each including:
    - Task Input: Formatted according to the input specifications.
    - Task Output: Formatted according to the output specifications.

Table 6: The prompts for instruction generation.

Demo Generation
Generate two instances for the following task. The text part in the samples needs to refer to the style, vocabulary, and themes in the Reference Texts. Carefully read the task description to ensure the correct labeling in the generated samples.

Reference Texts:

{Reference Text1}

{Reference Text2}

Task Description:

{Task Description}

Give your response in the following format:
Input: {}
Output: {}

Table 7: The prompts for demo generation.

listed in Table 7. During demonstration generation, we provide reference texts to enhance diversity. After generation, we analyze the distribution of demonstration categories. If the distribution is imbalanced, we generate additional examples for underrepresented categories to ensure a more balanced composition.

A.3 Difficulty Assessment

A.3.1 Detailed Calculation of Difficulty Metric

We compute the difficulty of a sample using the ranking-based metric. To adapt the student model to the data distribution, we first perform a warm-up phase using 5,000 distillation samples. For each token in the response, we estimate the size of the label space using top-p sampling, with p empirically set to 0.95. When aggregating the scores across tokens, we exclude those tokens whose scores are below a threshold $\varepsilon_d = 1 \times 10^{-6}$. However, to avoid division by zero, we ensure that at least one token is retained for each sample.

The following example illustrates the detailed calculation of our ranking-based difficulty metric, including the estimation of N_t and the overall calculation process. Given a triplet $(instr, x, y)$, the ranking-based difficulty score for each target token y_t is calculated through the following steps:

- **Instruction**: "Classify the sentiment of the following review as Positive, Negative, or Neutral."
- **Input** (x): "This product is a complete waste of money. I regret buying it."
- **Ground-truth label** (y): "Negative"

Step 1: Model Output Distribution
The model generates the following probability distribution over candidate label tokens:

Token	Probability
Pos	0.45
Neu	0.40
Neg	0.11
Mixed	0.02
Other tokens	0.02

Table 8: Model probability distribution over candidate label tokens

Step 2: Top-p Sampling ($p = 0.95$)
We calculate cumulative probabilities in descending order:

- 'Pos': 0.45
- 'Pos' + 'Neu': 0.45 + 0.40 = 0.85
- 'Pos' + 'Neu' + 'Neg': 0.85 + 0.11 = 0.96

Since the cumulative probability first exceeds 0.95 with the inclusion of 'Neg', the candidate set contains three tokens: {'Pos', 'Neu', 'Neg'}, resulting in $N_t = 3$. After sorting by predicted probability, the ground-truth token 'Neg' receives rank $r_t = 3$.

Step 3: Difficulty Score Calculation
Since the target token appears in the candidate set ($r_t \leq N_t$), we apply the first case of the formula:

$$d(y_t) = \frac{r_t - 1}{N_t} = \frac{3-1}{3} = \frac{2}{3} \approx 0.67. \quad (8)$$

To illustrate the maximum difficulty scenario, consider a case where the ground-truth token 'Neg' does not appear in the top-95% probability mass. In this situation, $r_t > N_t$, and the difficulty score becomes:

$$d(y_t) = 1. \quad (9)$$

This maximum score indicates that the correct label is not among the model's most probable predictions, representing the highest level of prediction difficulty.

The ranking-based difficulty metric provides an intuitive measure of prediction difficulty:

- **Lower scores** (closer to 0): The correct token has high predicted probability and low rank, indicating easier prediction.
- **Higher scores** (closer to 1): The correct token has low predicted probability and high rank, indicating more difficult prediction.
- **Maximum score** (exactly 1): The correct token is not among the top-p candidates, representing maximum difficulty.

A.3.2 Proxy Model

The proxy model is implemented as an autoregressive model with an additional regression head. It is initialized from the student model, *i.e.*, Llama-3.2-3B-instruct. We train the proxy model on a dataset of 50K samples using LoRA (Hu et al., 2022), with hyperparameters specified in Table 9.

Hyper-parameter	Value
Batch Size	3
Learning Rate	1.5e-4
Training Epoch	3
Learning Rate Deacy	Linear
Rank	64
Alpha	16
Target Module	k_proj,q_proj,v_proj,o_proj

Table 9: Hyperparameters for the proxy model's optimization.

A.4 Knowledge Distillation

In the process of constructing distillation samples, each instruction is paired with multiple randomly sampled user texts. Furthermore, we randomly sample 1 to 8 demonstrations from the demonstration pool. The instruction, selected demonstrations, and user text are then fed into the teacher model to generate a response. The resulting samples are subsequently used to optimize the student model, with the maximum sequence length set to 2048. The optimization hyperparameters for the three student models are listed in Tables 10, 11, and 12, respectively.

Hyper-parameter	Value
Batch Size	128
Learning Rate	5e-5
Training Epoch	4
Learning Rate Deacy	Cosine
Warmup Step Ratio	0
Weight Decay	0.1
Adam β_1	0.9
Adam β_2	0.999

Table 10: Hyperparameters for Llama-3.2-3B-instruct.

Hyper-parameter	Value
Batch Size	128
Learning Rate	2e-5
Training Epoch	4
Learning Rate Deacy	Cosine
Warmup Step Ratio	0.05
Weight Decay	0.1
Adam β_1	0.9
Adam β_2	0.999

Table 11: Hyperparameters for Qwen-3-4B.

B Further Data Analysis

We visualize all the obtained perspectives in Figure 9. Besides, we provide length statistics for the 50K samples in Figure 8.

Hyper-parameter	Value
Batch Size	128
Learning Rate	1e-5
Training Epoch	4
Learning Rate Deacy	Cosine
Warmup Step Ratio	0.05
Weight Decay	0.01
Adam β_1	0.9
Adam β_2	0.999

Table 12: Hyperparameters for Gemma-3-4B-it.

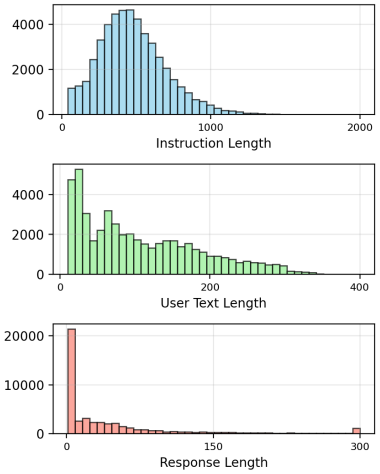

Figure 8: Length distribution in the distillation dataset.

C Evaluation Settings

Following the previous work (Zhang et al., 2025), we evaluate the models on SENTIBENCH using an in-context learning setup. The dataset statistics are shown in Table 13. The number of demonstrations is fixed at 4. We select demonstrations from the validation set using three different random seeds and report the average result of three runs. The prompts used are the same as those in Zhang et al. (2025), except for four datasets under the FSA category. For these datasets, we refine the prompts and update the performance of the baseline models accordingly. The refined prompts are presented in Table 18.

D Further Analysis

Analysis of Instruction-User Text Pairing. We compare two strategies for pairing instructions and user texts: (i) random pairing and (ii) attribute-based matching. As shown in Table 15, both methods achieve similar performance, with random pairing even showing a slight advantage. We attribute

Task	Dataset	Train	Dev	Test	#Class
BASIC SENTIMENT ANALYSIS					
Document-level sentiment classification	IMDb	3000	300	1000	2
	Yelp2	3000	300	1000	2
Sentence-level sentiment classification	SST2	3000	300	1821	2
	Twitter17	3000	300	1000	3
MULTIFACETED SENTIMENT ANALYSIS					
Irony detection	Irony18	3000	300	784	2
Emotion recognition	Emotion20	3000	300	1421	4
Stance detection	P-Stance	3000	300	2157	3
Intimacy analysis	MINT-English	1287	300	396	3
FINE-GRAINED SENTIMENT ANALYSIS					
Aspect term sentiment analysis	Rest16	1600	400	676	-
Aspect category sentiment analysis	Rest16	1600	400	676	-
Aspect sentiment quad prediction	Rest16	1264	316	544	-
Structured sentiment analysis	Opener	1744	249	499	-

Table 13: dataset statistics of SENTIBENCH.

Models	BSA				MSA				FSA				Avg
	IMDb	Yelp2	SST2	Twitter	Irony	Emoti.	Stance	Intim.	ATSA	ACSA	ASQP	SSA	
Llama-3-3B	92.57	96.53	93.59	61.45	64.00	68.88	71.43	33.32	52.74	53.23	14.33	23.56	60.47
+ KNOW & ICLDIST	94.30	98.17	95.41	69.57	85.25	77.47	75.10	48.24	53.07	65.22	24.61	36.17	68.55(+8.08)
+ Data Filtering	94.70	98.20	95.66	69.91	83.93	77.78	74.76	45.91	53.10	65.56	26.90	32.22	68.22(+7.75)

Table 14: Performance comparison of the KNOW&ICLDIST baseline trained on the full 300k dataset versus 150k filtered dataset.

Models	Avg-F1	Δ
Llama-3-3B	60.47	-
+ DIST *w/* Random-Pairing	68.19	+7.72
+ DIST *w/* Attribute-Matched-Pairing	67.61	+7.14

Table 15: Comparison between two instruction-user text pairing methods (%).

this outcome to the fact that random pairing leads to a more balanced class distribution in the resulting dataset, whereas attribute-based matching tends to introduce an excessive number of positive samples. For example, in the sarcasm detection task, attribute-based matching results in an overrepresentation of sarcastic samples and an underrepresentation of non-sarcastic ones. Based on these analyses, we adopt the random pairing strategy in our final framework.

Data Filtering on Other Baselines. We apply our data filtering method to the KNOW&ICLDIST baseline to investigate its generalizability. The results in Table 14 demonstrate the effectiveness and robustness of our method. Notably, performance degradation is minimal even when the dataset is reduced by 50%.

Models	TSA-R14	TSA-L14	ASA-R16	ASA-L16
IMPLICIT SENTIMENT SAMPLES				
GPT-3.5	43.11	30.73	52.75	29.25
Llama-3-70B	50.08	42.89	63.39	44.30
Llama-3-3B	22.65	21.45	40.46	17.13
+ OURS	37.93(+15.28)	30.28(+8.83)	53.33(+12.87)	27.94(+10.81)
MULTIPLE SENTIMENTS SAMPLES				
GPT-3.5	48.35	35.07	52.23	32.54
Llama-3-70B	54.40	49.31	60.13	44.37
Llama-3-3B	28.32	20.45	36.22	14.96
+ OURS	43.47(+23.02)	35.73(+15.28)	51.00(+14.78)	22.30(+7.34)

Table 16: Experimental results in complex contexts (F_1-score, %).

Results on Complex Contexts. Complex contexts refer to texts that contain implicit sentiment and express multiple sentiment polarities simultaneously. We evaluate the impact of distillation on the student model's ability to perform sentiment analysis in complex contexts. The evaluation is conducted on the dataset introduced by Zhang et al. (2024c),

under an in-context learning setup with 4 demonstrations. The results in Table 16 reveal the following: (1) Llama-3-3B performs significantly worse than Llama-3-70B on both types of complex context; (2) Our approach leads to substantial improvements in the performance of the Llama-3-3B model, with average gains of 11.95% and 15.11% across the two settings. These findings demonstrate that our approach can effectively enhance the student model's capability to handle complex contextual understanding.

Case Study of Difficulty Assessment. We present two representative examples of difficulty assessment in Table 17. Based on these cases, we make the following observations. Firstly, perplexity is not an effective indicator of a sample's true difficulty. As shown in the table, two samples with similar perplexity scores exhibit noticeably different levels of difficulty. Secondly, for relatively easier samples, both the ranking-based metric and the proxy model assign low difficulty scores, suggesting that their estimations are reasonably accurate in such cases. Thirdly, for more complex tasks, the proxy model tends to overestimate the difficulty. This is because the proxy model does not have access to the teacher model's response and thus cannot accurately determine whether it can replicate the teacher's output. In summary, effectively and efficiently estimating the difficulty of a sample remains a challenging problem. We believe this is a promising direction for future research.

Instruction: Classify the level of satisfaction expressed in a given text into one of the following predefined categories: Very Satisfied, Satisfied, Neutral, Dissatisfied, Very Dissatisfied. The output should be a single classification label.
Expected Output Format: A single string label from the following set: ["Very Satisfied", "Satisfied", "Neutral", "Dissatisfied", "Very Dissatisfied"]
Input: The first time we watched this movie we were all sitting in a ball on the couch! Over all a very nice horror movie, if you want to get scared! We all know the scary sound of Kayako's throat sound! My son's bedroom door creaks and sounds like it and creeps him out! I think It's one of the best horror movies we own! 5 *'s!
Output: Very Satisfied
Perplexity: 1.0009
Ranking-based Metric: 0
Proxy Model: 0.0479

Instruction: Analyze the input text and provide a JSON output containing the sentiment analysis results. The output should include the following fields:...
Input: Input: Award winning bakery indeed!!! I was searching for key lime pie in the Orlando area and read multiple reviews regarding Yvette's story. Impressed by all the awards she won as a new baker lead me to give her sweets a try. HANDS DOWN, Her key lime pie is the BEST! Sweet, creamy, zest filled, homemade crust goodness will keep you coming back for more!
Output:

```
{
  "sentiment": "Positive",
  "sentiment_intensity": 5,
  "sentiment_triggers": [
    "Award winning",
    "HANDS DOWN",
    "the BEST",
    "Sweet, creamy, zest filled, homemade crust goodness"
  ]
}
```

Perplexity: 1.0814
Ranking-based Metric: 0.1488
Proxy Model: 0.5234

Table 17: Representative examples for difficulty assessment.

FSA - ATSA - Rest16

Please perform Aspect Term Sentiment Analysis task. Given the sentence, extract all aspect terms and their corresponding sentiment polarities.

Return your answer in JSON format as an array of objects, each with the fields:

- "aspect_term": the extracted aspect
- "sentiment": one of "positive", "negative" or "neutral"

Example output format:

[{"aspect_term": "aspect_term", "sentiment": "sentiment"}]

FSA - ACSA - Rest16

Please perform aspect-level sentiment analysis task. Given the sentence, tag all aspect categories and their corresponding sentiment polarities.

Aspect category should be selected from ["ambience general", "drinks prices", "drinks quality", "drinks style_options", "food prices", "food quality", "food style_options", "location general", "restaurant general", "restaurant miscellaneous", "restaurant prices", "service general"], and sentiment should be selected from ["negative", "neutral", "positive"].

Return your answer in JSON format as an array of objects, each with the fields:

- "aspect_category": the selected aspect category
- "sentiment": the sentiment polarity

If there are no aspect-sentiment pairs, return an empty list.

Example output format:

[{"aspect_category": "aspect_category", "sentiment": "sentiment"}]

FSA - ASQP - Rest16

Please perform Aspect Sentiment Quad Prediction task. Given the sentence, extract all (aspect term, aspect category, opinion, sentiment polarity) quadruples.

1. Aspect category should be selected from ["ambience general", "drinks prices", "drinks quality", "drinks style_options", "food general", "food prices", "food quality", "food style_options", "location general", "restaurant general", "restaurant miscellaneous", "restaurant prices", "service general"].
2. Sentiment polarity should be selected from ["negative", "neutral", "positive"].
3. If there is no aspect term, use "NULL" as the aspect term. Only aspect term can be "NULL", aspect category, opinion and sentiment polarity CANNOT be "NULL".

Return your answer in JSON format as an array of objects, each with the fields:

- "aspect_term": the extracted aspect term (or "NULL")
- "aspect_category": the selected aspect category
- "opinion": the expressed opinion
- "sentiment": the sentiment polarity

Example output format: [{"aspect_term": "aspect_term", "sentiment": "sentiment", "opinion": "opinion", "sentiment": "sentiment"}]

FSA - SSA - Opener

Please perform the Structured Sentiment Analysis task. Given a sentence, extract all opinion tuples in the format (holder, target, sentiment expression, sentiment polarity).

Each tuple should contain:

- Holder: The entity expressing the sentiment, if there is no explicit holder, use "NULL" as the holder.
- Target: The entity being evaluated, if there is no explicit target, use "NULL" as the target.
- Sentiment Expression: The phrase conveying the sentiment, if there is no sentiment expression, use "NULL".
- Sentiment Polarity: The polarity of the sentiment, either positive, negative, or neutral, if there is no sentiment expression, use "NULL".

Follow these rules:

1. If there is no sentiment expression, return "NULL" for all fields.
2. Return your answer in JSON format as an array of objects, each with the fields:
   - "holder"
   - "target"
   - "sentiment_expression"
   - "sentiment_polarity"

Example output format: [{"holder": "holder", "target": "target", "sentiment_expression": "sentiment_expression", "sentiment_polarity": "sentiment_polarity"}]

Table 18: The refined prompts for fine-grained sentiment analysis (FSA) task.

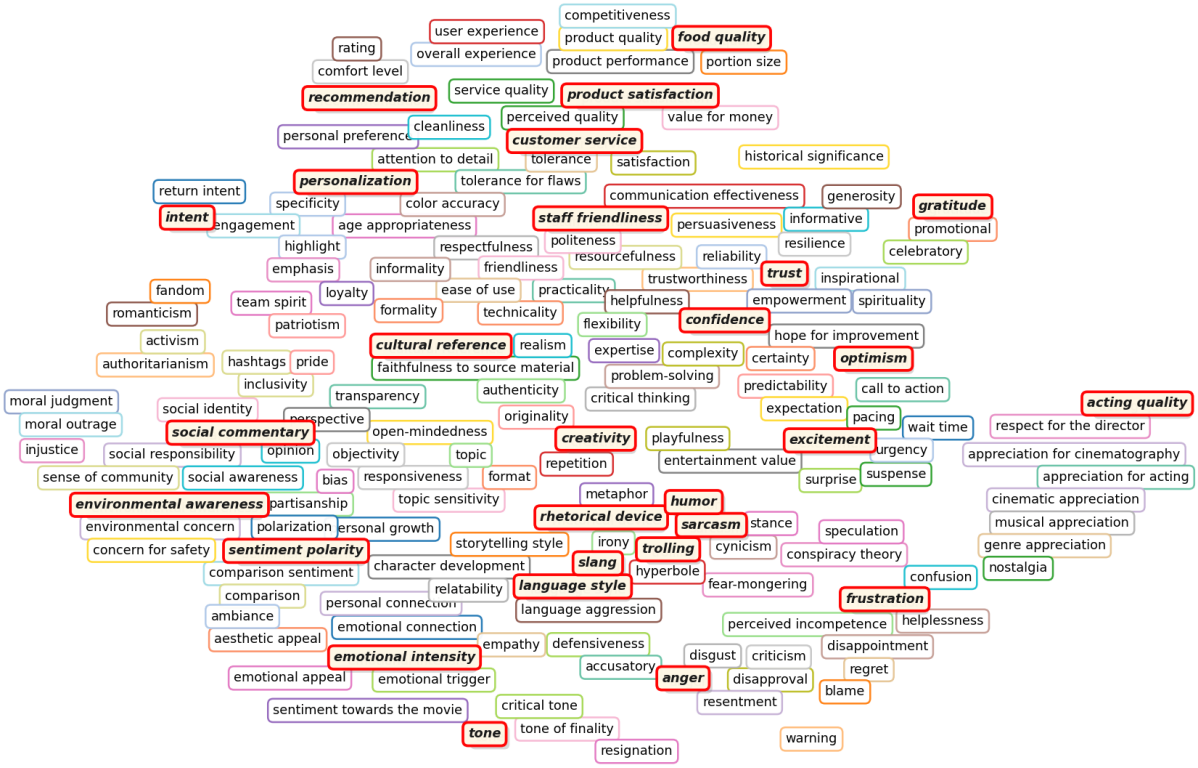

Figure 9: A t-SNE (van der Maaten and Hinton, 2008) visualization of the generated analytical perspectives using the UAE embeddings (Li and Li, 2024). Representative perspectives are highlighted with red bounding boxes. For clarity, overly long names have been appropriately shortened (e.g., *sense of helplessness* → *helplessness*).