A Simple-Yet-Efficient Instruction Augmentation Method for Zero-Shot Sentiment Classification

Yang Zhao[1], **Masayasu Muraoka**[1], **Issei Yoshida**[1],

Bishwaranjan Bhattacharjee[2], **Hiroshi Kanayama**[1]

[1]IBM Research - Tokyo, 19-21 Nihonbashi Hakozaki-cho, Chuo City, Tokyo, 103-8510, Japan,
[2]IBM Research, Yorktown Heights, New York 10598, USA

Abstract

Instruction tuning significantly enhances the performance of large language models in tasks such as sentiment classification. Previous studies have leveraged labeled instances from sentiment benchmark datasets to instruction-tune LLMs, improving zero-shot sentiment classification performance. In this work, we propose a simple-yet-efficient instruction augmentation method which does not rely on any actual labeled sentiment instances. With just 240 pseudo instruction instances, the proposed method significantly improve the classification performance across several LLMs on 12 benchmark datasets, increasing scores by 30 points and outperforming LLMs that utilize more complex instruction tuning methods by 5.1 points. Surprisingly, the models tuned with 240 pseudo-instructions even outperform those tuned with actual domain-specific instruction instances. Despite method's simplicity, our further analysis suggests that the probability shift toward the positive and negative classes and its generalization ability may be the primary driver of the improvement. Our instruction data and code is released[1] for reproduction and future research.

1 Introduction

Sentiment analysis has long been an established area of research in Natural Language Processing (NLP). With recent advancements in Large Language Models (LLMs), impressive zero-shot performance in sentiment analysis was achieved by instruction-tuned LLMs (Deng et al., 2023; Wang et al., 2023a; Scaria et al., 2023). A typical sentiment **Instruction Instance** is a tuple with three components *(T, I, O)*:

- **Instruction Text (T)**: *Classify the following sentence into either positive, neutral or negative sentiment.*
- **Input (I)**: *A movie journey worth taking.*
- **Output (O)**: *The sentiment is positive.*

where the instruction text (**T**) refers to the user instruction. It usually specifies desired outputs; the input (**I**) refers to the input sentence or document for the sentiment task; the output (**O**) refers to the ground-truth answer corresponding to the instruction text.

Previously, many sentiment analysis studies have utilized actual training instances in sentiment benchmark datasets as Input (**I**) and corresponding labels as Output (**O**) for instruction tuning. For example, Wei et al. (2021) instruction-tuned LLMs across various NLP tasks, including four sentiment datasets, while Chung et al. (2022) further expanded this approach to over 1,800 NLP tasks. Considering sentiment classification spans diverse domains such as finance, restaurants, movies, and politics, obtaining a large number of domain-specific labeled instances for instruction tuning is labor-intensive and inefficient.

To enhance this aspect, we propose a simple-yet-efficient instruction augmentation method to construct sentimental adjective-based pseudo instructions which do not rely on any training instances in sentiment benchmark datasets. Subsequently, we instruction-tune Llama2-7b,13b,70b base models (Touvron et al., 2023) and the Falcon base model (Almazrouei et al., 2023) and evaluate their zero-shot performance on 12 sentiment benchmark datasets (7 general sentiment analysis datasets and 5 aspect-based sentiment analysis datasets). The results show that this method significantly outperforms the base models by 30 points and surpasses other instruction-tuned models by an average of 5.1 points. Surprisingly, models tuned with 240 pseudo instructions even outperform those tuned with actual domain instruction instances. Despite the method's simplicity, further analysis suggests that the improvement is primarily driven by a prob-

[1]https://github.com/code4coling/code

Proceedings of the 31st International Conference on Computational Linguistics, pages 1585–1599
January 19–24, 2025.

ability shift of the sentiment polarity classes. Our contribution are three-fold:

- We proposed a simple-yet-efficient sentiment instruction augmentation method significantly boosting the zero-shot performance across 12 sentiment benchmark datasets.
- We conducted ablation study, domain analysis, and offered a perspective of probability shift to demonstrate effectiveness and advantage of the proposed method.
- We released constructed instruction instances and experimental code publicly for future research.

2 Sentimental Adjective Instruction Construction

We herein describe the steps to construct pseudo instances using sentimental adjectives. Section 2.1 outlines the process for collecting diverse sentiment instruction text (**T**) from various corpora. Section 2.2 details the steps to construct instruction using sentimental adjectives for Input (**I**) and Output (**O**).

2.1 Instruction Text (T) Collection

User instructions exhibit a wide variety of paraphrasing. For instance, *Please classify the sentence into either positive, negative, or neutral* can be also expressed like *Please determine whether the sentence is positive, negative, or neutral*. To increase the diversity, we collect sentiment instruction text (**T**) from five widely-used instruction datasets written by either human annotators or LLMs, as follows: (1) SuperNI (Wang et al., 2022), which contains 96k instructions written by humans covering 1600+ NLP tasks. (2) Alpaca[2] (Taori et al., 2023), which contains 52k instructions generated by GPT-3 (davinci-003). (3) Self-instruct (Wang et al., 2023b), which contains 82k instructions generated by GPT-3 (vanilla). (4) Unnatural Instructions (Honovich et al., 2023), which contains 68k instructions generated by GPT-3 (davinci-002). (5) Baize (Xu et al., 2023), which contains 210k instruction instances created by prompting ChatGPT and letting it converse with itself.

We extracted all the instruction text (**T**) from these datasets and designed several heuristics to filter out only sentiment classification-related instruction. Please see Appendix A for details on instruction filtering. As this work focuses on sentiment classification, we retained instruction texts only if they contain the terms 'sentiment', 'positive', 'negative', and 'neutral'. Finally, 110 diverse sentiment instruction text (**T**) are yielded, and we empirically determine to use 80 for training and 30[3] for testing during instruction tuning. For the aspect-based sentiment classification task, we add *with respect to the TARGET* to the instruction text and replace TARGET with the specific aspect. Table 10 in Appendix shows 10 samples out of 110 instruction texts.

2.2 Sentimental Adjective based (I, O) Pair

Inspired by the concept of evaluative adjectives in linguistics, we describe the four steps to automatically collect pairs of instruction input (**I**) and output (**O**). Evaluative adjectives often express value judgments and convey opinions, emotions, or subjective interpretations. For instance, adjectives like *beautiful* imply a positive sentiment, while *awful* suggests a negative one. We refer to our collected adjectives as *sentimental adjectives*.

Step 1. Collect sentimental adjective candidates. We start by collecting adjectives from SentiWordNet 3.0[4] (Baccianella et al., 2010) where each sense of an adjective word w is assigned two scores: a positive score (S_{pos}) and a negative score (S_{neg}) where $0 \leq S_k \leq 1$ and $k \in \{pos, neg\}$. The selection criteria is:

1. Choose all words where at least one of its senses meets the criteria: $S_{pos} \geq r$ and $S_{neg} = 0.0$ to compile positive word list L^1_{pos}
2. Choose all words where at least one of its senses meets the criteria:: $S_{neg} \geq r$ and $S_{pos} = 0.0$ to compile negative word list L^1_{neg}
3. Choose all words where at least one of its senses meets the criteria: $S_{pos} = 0.0$ and $S_{neg} = 0.0$ to compile neutral word list L^1_{neu}

We empirically determine the threshold r to trade off between the number and quality of adjectives. Please see Table 11 in Appendix for L^1.

Step 2. Align with sentiment word sense. This step aims to refine the adjective lists in Step 1.

[2]https://github.com/gururise/AlpacaDataCleaned/

[3]A larger test set usually leads to more solid results. By increasing the test set to 30, e.g., 70b model would need to infer 54,630 instances. Considering experimental cost, we stopped increasing the test set size and set it to 30.

[4]https://github.com/aesuli/SentiWordNet. It is under CC BY-SA 4.0 license.

For instance, one sense of the word 'fresh' meets the criteria $S_{neg} \geq 0.75$ and $S_{pos} = 0.0$, this word is therefore included in the negative list L^1_{neg}. However, 'fresh' often conveys a non-negative meaning, typically referring to something new or unused. including this word in the negative list may confuse the model during instruction tuning. To address this, we utilize pre-defined positive (V_{pos}) and negative (V_{neg}) vocabularies in Hu and Liu (2004). Words in lists L^1_{pos} and L^1_{neg} are excluded if they do not appear in V_{pos} and V_{neg}, respectively. Words in L^1_{neu} are removed if they appear in either V_{pos} or V_{neg}. This process results in three refined lists: L^2_{pos}, L^2_{neg}, and L^2_{neu}. Please see Table 12 in Appendix for L^2.

Step 3. Rank word by frequency.
This step focuses on selecting more domain-agnostic words by leveraging frequency information. We use English Wikipedia[5] to obtain word frequency for ranking adjectives in each list in descending order based on their frequency. If an adjective in L^2_{pos}, L^2_{neg}, and L^2_{neu} is not in the wiki frequency list, its frequency would be set to zero. After ranking, frequent words such as *best*, *great*, and *important* appear at the top of the positive list, whereas the original words in the list are *legendary*, *solid*, and *gallant*. We note the ranked lists as L^3_{pos}, L^3_{neg}, and L^3_{neu}. Please see Table 13 in Appendix for L^3.

Step 4. Add negation words.
This step helps LLMs to better handle sentences containing negation words, which is common in sentiment classification. We add the negation word *not* directly before adjectives (e.g., *not beautiful*) for $X\%$ of instances in only L^3_{pos} and L^3_{neg}. Subsequently, adjectives with negation from the positive list are transferred to the negative list and vice versa. This process yields the final lists: L^4_{pos}, L^4_{neg}, and L^4_{neu} (where $L^4_{neu} = L^3_{neu}$). Please see Table 14 in Appendix for L^4.

After completing steps 1 to 4, we take the first instruction text **T** from 80 instruction texts in Section 2.1, the first adjective from L^4_{pos} and *positive* to form the first tuple (**T**, **I**, **O**); Continue this process until the 80th instruction text is taken. Then, we obtained 80 tuples for the positive class, 80 tuples for the negative class, 80 tuples for the neural class respectively. Please refer to Table 9 in Appendix for examples of constructed tuple for each class.

[5]https://jwsmythe.com/tools/wordlist/wikipedia-word-frequency-master/results/enwiki-2023-04-13.txt

3 Experiment

3.1 Experimental Setup

The constructed 240 tuples are split into 80% for the training set and 20% for the development set. We set the threshold r in SentiWordNet 3.0 in Step 1 to 0.75, and negation word percentage X to 10%, according to performance on the development set. For training, we follow Touvron et al. (2023) by utilizing an auto-regressive objective and zeroing out the loss on tokens from the user prompt, including instruction text and input, while backpropagating only on instruction output. Of the 110 instruction texts, we use 80 for model training and development, and remaining 30 for testing. This approach better replicates real-world scenarios and ensures the instruction texts are unseen, as users' instructions are inherently unpredictable.

During training, we employ the efficient parameter tuning technique, LoRA (Hu et al., 2021), with a LoRA rank of 8 and LoRA alpha of 32. We set learning rate to 2e-4 and batch size to 2. During inference, we follow Dettmers et al. (2022) to load models in the 8-bit mode which significantly speeds up the inference and has negligible impact on the final performance. We set the maximum number of generated tokens to 20. All the experiments are conducted using one A100 GPU. For the largest model, Llama2-70B, only 0.024% of the parameters are trainable, and training took approximately 2 GPU hours. For testing, evaluating the 70B model across 12 datasets requires approximately 8 GPU hours. Other smaller models require less time.

3.2 Evaluation Metric

Since all the instruction texts we collected explicitly specify the output space as positive, negative, or neutral label, we adopt the following metric for calculating instance-wise accuracy: 1) Score 1 if the output string contains the ground-truth label and does not contain other classes' ground-truth labels (case insensitive); 2) Score 0, otherwise. To verify the reliability of this metric, we asked a human annotator to rate two hundred Llama2-7b's outputs from the SST-2 dataset. The annotator observed that the output was either a single label like *positive* or a sentence like *The sentiment is positive.*; in minor cases, the output was nonsensical. The anno-

tator then assigned a score of 1 if the model output expressed the same sentiment as the ground-truth label, and 0 otherwise. We calculated the Pearson correlation coefficient between the two sequences of zero-one labels (one from the human annotator and one from the automatic metric), which yielded a value of 1.0. Observing such a high correlation score between the human annotator and our automatic metric, we decided to use this metric for all datasets.

3.3 Dataset

We experiment with 7 general sentiment classification datasets, i.e., SST-2 (Socher et al., 2013), IMDB, Yelp, Amazon datasets from (Kotzias et al., 2015), Airline[6], Debate[7], financial phrasebank (Malo et al., 2014a) as well as 5 aspect-based sentiment classification datasets[8] from the Workshop on Semantic Evaluation (SemEval) in 2014, 2015, and 2016. Please refer to related work for detailed description of datasets.

Dataset	Domain	Size	# Class	Aspect
SST-2	Movie	1,821	2	no
Yelp	Restaurant	1,000	2	no
Amazon (Amaz)	Product	1,000	2	no
IMDB	Movie	1,000	2	no
Airline	Operation	1,000	3	no
Debate (Deba)	Politics	1,000	3	no
PhraseBank (PB)	Finance	970	3	no
SemEval-14lap	Laptop	543	3	yes
SemEval-14res	Restaurant	994	3	yes
SemEval-15res	Restaurant	485	3	yes
SemEval-15hot	Hotel	215	3	yes
SemEval-16res	Restaurant	514	3	yes

Table 1: Statistics of sentiment classification datasets.

Table 1 shows the statistics of each dataset. We paired each sentence from the sentiment benchmark datasets with 30 instruction texts for testing. For instance, in the case of SST-2, this resulted in 1,821 × 30 = 54,630 instances used for testing instruction-tuned models. The same procedure was applied to the other datasets.

3.4 Models

We instruction-tuned Llama2 base model[9] (Touvron et al., 2023), and falcon-40b base model (Almazrouei et al., 2023) using our constructed 240 instruction tuples (**T**, **I**, **O**), noted as **base+ours**. In addition, we consider the following comparison methods:

base+ours w/o adjective Previous works, such as Kung and Peng (2023), have pointed out that some instruction-tuned models do not fully utilize instructions, and that the impressive performance gains from instruction tuning may stem from models learning superficial patterns, such as the output space and format. To verify this, we replaced the sentimental adjectives with empty strings to ablate the input, while keeping the instruction text and output format unchanged.

lexicon-match baseline We add a sentiment lexicon match-based model (Gilbert, 2014), which directly utilizes the presence of positive (e.g., *great*, *good*, and *nice*) and negative words (e.g., *sad*, *bad*, and *worse*) to determine the sentiment polarities. This aims to determine if good performance can be achieved through simple sentimental word matching, without injecting these sentimental adjectives via instruction tuning.

llama2 chat model The Llama2 chat model began supervised fine-tuning with converted instructions from 1.8K NLP tasks (Chung et al., 2024). The model was further fine-tuned on 27,540 annotated instructions and millions of human preference data via reinforcement learning. We believe this provides a powerful baseline, even for our sentiment classification task.

falcon chat model It is also known as the Falcon-40B-Instruct model[10], which is fine-tuned on hundreds of thousands of QA and dialog instances from Quora, Stack Overflow, and MedQuAD questions.

4 Result and Analysis

4.1 Overall performance

Table 2 shows comparison results and our observations are as follows:

[6]https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment. We only use 1k instances given the dataset is relatively large.

[7]https://www.kaggle.com/datasets/crowdflower/first-gop-debate-twitter-sentiment. We only use 1k instances.

[8]https://github.com/kevinscaria/InstructABSA/tree/main/Dataset

[9]we chose Llama 2 and Falcon base because they provide a gradual size increase from 7B and 13B to 40B (Falcon) and 70B. This allows us to investigate how model size affects our data augmentation method and the latest versions like Llama 3 do not include a 13B size for this purpose.

[10]https://huggingface.co/tiiuae/falcon-40b-instruct

Dataset	SST-2	Yelp	Amaz	IMDB	Deba	Airline	PB	14hap	14res	15res	15hot	16res	Ave.
△ Lexicon-match baseline	59.2	64.7	69.6	69.0	55.4	63.6	56.0	68.9	81.5	74.4	74.9	78.4	67.9
#1 llama2-7b-base	49.4	51.2	40.8	46.7	34.7	37.4	27.3	40.0	61.5	53.8	61.5	61.6	47.2
#2 llama2-7b-chat	78.8	88.8	83.5	86.2	61.6	**69.9**	**60.5**	75.5	83.6	78.9	71.5	75.7	76.2
#3 base+ours w/o adjective	38.9	35.5	32.3	37.9	35.2	35.6	29.9	18.3	13.8	24.9	16.4	13.4	27.7
#4 base+ours	**89.5**	**96.1**	**94.1**	**94.4**	**62.0**	67.6	53.5	**82.1**	**88.4**	**86.3**	**84.4**	**89.4**	**82.3**
$1 llama2-13b-base	47.0	52.5	43.4	49.3	36.1	40.8	41.7	46.1	61.2	56.7	58.0	57.9	49.2
$2 llama2-13b-chat	71.2	79.0	75.0	77.9	62.9	69.5	59.1	**68.9**	76.9	71.4	63.1	65.4	70.0
$3 base+ours w/o adjective	49.6	50.4	44.4	50.7	38.7	43.1	26.3	28.0	35.1	41.9	33.9	24.9	38.9
$4 base+ours	**80.5**	**88.4**	**75.9**	**86.1**	**63.1**	**69.8**	**62.0**	62.9	**81.6**	**78.1**	**73.0**	**77.2**	**74.9**
&1 llama2-70b-base	55.8	42.7	43.7	48.1	34.5	39.1	31.6	44.9	45.1	47.0	44.3	54.8	44.3
&2 llama2-70b-chat	81.9	90.0	87.6	88.6	**64.8**	**72.6**	**68.8**	74.5	80.9	77.1	72.3	67.8	77.2
&3 base+ours w/o adjective	72.4	80.5	75.8	77.4	43.6	51.1	29.4	64.3	77.1	72.3	71.1	67.0	65.2
&4 base+ours	**92.5**	**97.9**	**95.8**	**96.3**	63.0	71.1	55.3	**80.4**	**89.0**	**85.6**	**88.3**	**85.0**	**83.4**
♢1 falcon-40b-base	69.9	72.1	61.8	63.1	36.6	42.5	27.5	50.1	65.8	66.3	60.7	67.2	57.0
♢2 falcon-40b-instr.	78.9	89.2	80.0	83.2	51.5	55.2	40.3	74.7	**86.3**	**81.3**	**83.3**	85.3	74.1
♢3 base+ours w/o adjective	63.6	58.7	46.4	53.4	36.0	38.9	23.8	35.7	56.5	51.7	51.0	53.2	47.4
♢4 base+ours	**92.0**	**91.2**	**87.8**	**88.1**	**55.0**	**62.0**	**43.2**	**77.8**	84.1	80.6	80.3	85.3	**77.3**

Table 2: Accuracy of zero-shot sentiment classification on 12 benchmark datasets. Best results associated with the same base model are in bold.

(1) Our instruction-tuned models (**base+ours**) outperform all base models by 30 points and even all chat models by 5.1 points on average, validating the effectiveness and efficiency of our method given that our models used only 240 instruction instances for tuning. Moreover, our instruction-tuned Llama2-70B model achieves the best average performance, while our instruction-tuned Llama2-7B model is also highly competitive. This suggests that model size remains an important factor in the effectiveness of instruction tuning.

(2) The results of **base+ours w/o adjective** show significant performance degradation for Llama2-7B (#3), Llama2-13B ($3), and Falcon-40B (♢). While the "empty-input" instruction tuning boosts Llama2-70B's performance to some extent (&3), combining it with our sentimental adjectives achieves the best performance (&4). Comparison between *base+ours* and *base+ours w/o adjective* verifies that the performance improvements are largely not attributed to learning the output space formats, such as positive and negative labels.

(3) To investigate whether our *base+ours* models simply memorize sentimental adjectives for making predictions, we added a sentiment lexicon match-based model for comparison. The results show that our models significantly outperform this baseline (△), indicating that incorporating sentimental adjectives into LLMs through instruction tuning equips the models to handle not only straightforward sentiment lexicon-based cases but also more challenging cases without explicit sentiment lexicons.

(4) All models, whether instruction-tuned or not, struggle with the finance dataset (PB) compared to other domains. We investigated vocabulary overlap between domains, as shown in Figure 2, which shows that the finance domain is significantly different from other domains. This suggests a high necessity for LLMs to undergo domain adaptation to further enhance domain-specific zero-shot performance.

Furthermore, we also investigated how the number of instruction tuples affects the performance of the proposed method. We experimented with 50, 100, 150, and 200 instruction tuples, balancing each class, using the llama2-7b model. The results in Table 8 in Appendix show that 240 tuples provide a good trade-off between performance enhancement and computational efficiency.

4.2 How each step contributes to performance?

To investigate how each step in the word selection process in Section 2.2 impacts the model's instruction-tuning performance, we conducted ablation studies and have the following finding on results in Figure 1: (1) Overall, regarding the average performance, each step contributes to perfor-

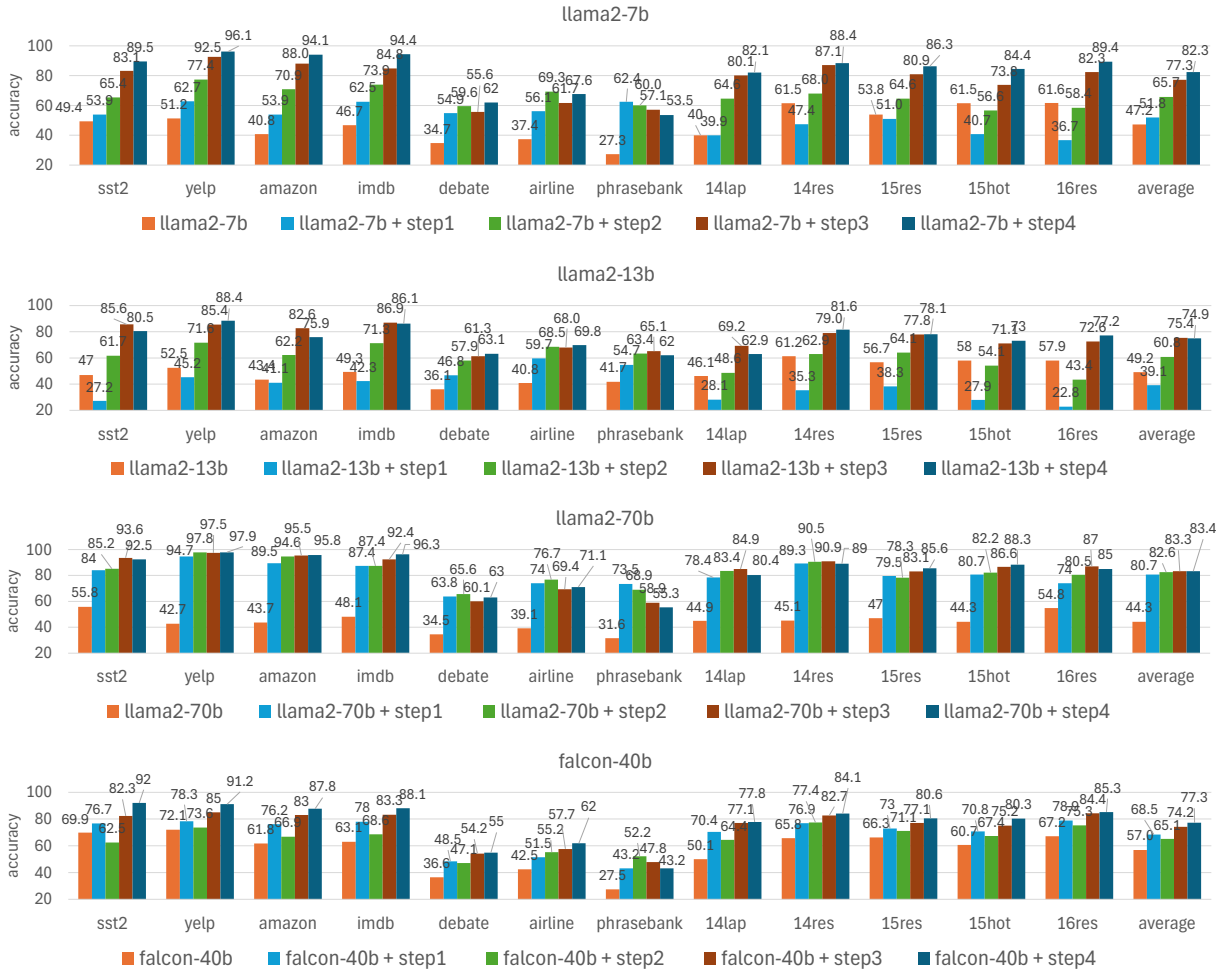

Figure 1: An ablation study was conducted for each step. The llama2-7b+step1 refers to the fine-tuning of the llama2-7b model using data selected by step1. The llama2-7b+step2 involves fine-tuning the llama2-7b model with data selected from step1 to step2. The llama2-7b+step3 pertains to fine-tuning the llama2-7b model using data from step1 to step3. The llama2-7b+step4 describes the fine-tuning of the llama2-7b model with data spanning from step1 to step4.

top-500 frequent word overlap (%)	SST-2 movie	Yelp resto	Amazon product	IMDB movie	Debate politics	Airline oper.	PB finance	14lap laptop	14res resto	15res resto	15hot hotel	16res resto	average
SST-2 (movie)	100	25	27	51	25	20	9	21	20	21	17	18	29
Yelp (resto)	25	100	36	33	24	31	12	23	45	45	32	43	38
Amazon (product)	27	36	100	33	25	29	16	36	25	25	26	23	33
IMDB (movie)	51	33	33	100	29	24	10	25	23	26	22	23	33
Debate (politics)	25	24	25	29	100	29	12	18	15	18	16	14	27
Airline (oper.)	20	31	29	24	29	100	16	21	17	20	25	18	29
PB (finance)	9	12	16	10	12	16	100	14	8	8	13	8	19
14lap (laptop)	21	23	36	25	18	21	14	100	18	19	19	17	28
14res (resto)	20	45	25	23	15	17	8	18	100	38	25	39	31
15res (resto)	21	45	25	26	18	20	8	19	38	100	22	37	32
15hot (hotel)	17	32	26	22	16	25	13	19	25	22	100	25	28
16res (resto)	18	43	23	23	14	18	8	17	39	37	25	100	30

Figure 2: Vocabulary overlap (%) of 12 sentiment datasets. *resto* stands for restaurant and *oper.* stands for operation. Vocabularies for each domain are created by considering the top 500 most frequent words (excluding stopwords) in each dataset. Red color indicates a high degree of vocabulary overlap, while blue color indicates a low degree of vocabulary overlap.

Dataset	SST-2 movie	Yelp resto	Amaz product	IMDB movie	Deba politics	Airline oper.	PB finance	14hap laptop	14res resto	15res resto	15hot hotel	16res resto	Ave.
Model+SST-2	**97.2**	87.0	79.0	78.4	52.2	52.5	40.1	66.5	76.6	71.9	72.9	71.3	70.5
Model+Yelp	87.1	92.3	88.0	91.2	62.0	58.6	48.1	80.1	**90.1**	84.6	85.2	87.7	79.6
Model+Amaz	91.4	89.5	**98.0**	75.7	48.9	54.4	30.3	75.8	84.5	77.1	80.3	78.2	73.7
Model+IMDB	79.1	88.2	82.1	**96.2**	52.4	53.6	39.3	68.8	76.0	71.8	72.4	72.5	71.0
Model+Debate	43.2	56.8	44.1	51.2	58.7	59.4	62.4	31.1	35.6	41.0	25.7	23.8	44.4
Model+Airline	61.7	78.9	62.6	64.6	59.0	**73.6**	65.5	49.2	70.9	68.3	58.5	62.6	64.6
Model+PB	54.0	60.4	44.7	51.1	59.9	59.1	**83.7**	33.8	32.4	41.4	35.3	26.1	48.5
Model+14lap	79.1	90.4	80.7	81.7	61.5	69.4	68.7	**82.9**	82.8	79.1	70.3	80.8	77.3
Model+14res	69.5	84.6	76.9	78.3	60.7	71.0	65.3	70.9	81.2	75.8	63.1	75.2	72.7
Model+15res	77.8	91.7	86.4	82.2	55.3	59.0	45.0	74.3	79.3	81.7	72.9	78.6	73.7
Model+15hot	91.2	78.6	68.1	70.7	49.6	52.2	40.7	59.5	69.0	64.2	**87.1**	63.9	66.2
Model+16res	88.5	93.8	89.3	89.9	**62.4**	61.4	49.7	76.6	85.0	82.5	77.7	**90.0**	78.9
ours	89.5	**96.1**	94.1	94.4	62.0	67.6	53.5	82.1	88.4	**86.3**	84.4	89.4	**82.3**

Table 3: Performance of 12 instruction-tuned llama2-7b models on domain-specific ground-truth tuple. We used one dataset for training and all datasets for testing. All other experimental and hyperparameter settings were kept the same as in the proposed method. *resto* stands for restaurant and *oper.* stands for operation. Best results are in bold.

mance improvement, confirming the necessity of implementing all four steps; (2) Steps contribute differently to the average performance. In small models like Llama2-7B, each step consistently improves performance. In contrast, for larger models like Llama2-70B, Step 1 leads to significant improvements, indicating that larger models are more effective at learning from the unrefined sentimental adjective list (L_1 data). Nevertheless, smaller gains (2.7 points) were also observed after implementing Steps 2, 3, and 4.

4.3 Our domain-agnostic v.s. domain-specific ground-truth instruction instance

One question is what the performance would be when using real instruction instances in the sentiment benchmarks. To answer this, we compared our domain-agnostic adjective-based instruction instances with domain-specific ground-truth instances by replacing instruction input **I** and output **O** with ground-truth sentences and their sentiment labels, forming a new tuple *(T, I', O')*. For example, when using sentences from the Financial Phrasebank data, we refer to this model as a financial domain-specific instruction tuple.

For a fair comparison across each of the 12 benchmark datasets, we extracted 240 domain-specific instances, with 80 instances per sentiment class to construct new tuples *(T, I', O')*. For two-class sentiment classification, we selected 120 instances per class[11]. Additionally, since the llama2-7b model achieves nearly the best result (only 0.9 points behind the llama2-70b) but has significantly lower latency, we conducted our experiments using the llama2-7b model. Consequently, we trained 12 domain-specific instruction-tuned models. Table 3 shows the experimental result of instruction-tuning using domain-specific tuples. We have following observations:

(1) For 8 out of 12 sentiment datasets, domain-specific instruction-tuned models achieve the best performance within its own domain. If domains are similar, the improvement is also significant. For example, both the Yelp and SemEval-14res datasets pertain to the restaurant scenario. The best performance on the SemEval-14res dataset was observed when the model was instruction-tuned using the Yelp dataset. Interestingly, our model achieved the best average performance across 12 datasets, compared to each domain-specific model. This highlights a distinct advantage of our method: **our domain-agnostic pseudo instruction tuning avoids overfitting too much to specific domains, leading to better generalization across other domains.**

(2) Instruction tuning with the Yelp dataset resulted in the best average performance (79.6) across all datasets, while fine-tuning with the PB (finance) and Debate (politics) dataset resulted in low average performances. To investigate the underlying reasons, we examine domain overlap among the datasets, following the approach of Gururangan et al. (2020): We identified the 500 most frequent

[11] Since SemEval-15hot has only 215 instances, we leverage all the 215 instances for training.

P_neg_ave	SST-2	Yelp	Amaz	IMDB	Deba	Airline	PB	14hap	14res	15res	15hot	16res
llama2-7b-base	0.014	0.040	0.035	0.030	0.014	0.024	0.008	0.025	0.032	0.036	0.026	0.031
llama2-7b-base+ours	**0.016**	**0.045**	**0.042**	**0.032**	**0.016**	**0.027**	**0.012**	**0.028**	**0.034**	**0.039**	**0.032**	**0.034**

Table 4: Probability shift of negative adjectives of using llama2-7b-base and llama2-7b-base+ours.

P_pos_ave	SST-2	Yelp	Amaz	IMDB	Deba	Airline	PB	14hap	14res	15res	15hot	16res
llama2-7b-base	0.013	0.027	0.026	0.025	0.012	0.020	0.009	0.020	0.021	0.023	0.017	0.021
llama2-7b-base+ours	**0.015**	**0.031**	**0.031**	**0.026**	**0.014**	**0.025**	**0.011**	**0.023**	**0.023**	**0.025**	**0.020**	**0.024**

Table 5: Probability shift of positive adjectives of using llama2-7b-base and llama2-7b-base+ours.

non-stopwords[12] in each dataset as a representation of its domain. As illustrated in Figure 2, the last column shows the average vocabulary overlap with all other datasets which indicates how close a dataset is to all other datasets. We found that the finance domain is the furthest from other domains, the politics domain is the second furthest, while Yelp is the closest. This may explain why, in Table 3, instruction-tuning using Yelp achieves the best average performance, while instruction-tuning using the PhraseBank finance dataset achieves the lowest one.

Given the promising results using Yelp domain-specific instances, we increased the number of instances from 240 to 1,000. However, this resulted in performance drop across all datasets except Yelp. Please see Appendix C for details.

4.4 Why simple adjective-based tuning improves performance of LLMs?

To investigate why LLMs significantly improved zero-shot classification performance after instruction tuning with just 240 sentimental adjectives, we hypothesize that this is primarily due to the *probability shift* of sentimental words after our instruction tuning. More specifically, after instruction tuning, a LLM is more likely to associate, for example, negative sentences with many other negative words, leading to better prediction performance, even though those negative words never appear during instruction tuning.

To verify this, we take all negative sentences from 12 datasets respectively and append *The sentiment is X*[13] to each negative sentence, such as *The decor of the restaurant is terrible. The sentiment is X*. Then, we replace X with 1,000 negative adjectives collected in Step 2 of Section 2.2, such as "disappointing"[14], and sum the probabilities of 1,000 negative adjectives from the model's softmax layer for each sentence to get P_neg_sum. We average P_neg_sum over all negative sentences in each dataset to finally get P_neg_ave. For example, in the SST-2 dataset, there are 912 negative sentences and we replace X in each sentence with 1,000 negative adjectives, generating 912,000 test cases. Our assumption is that if the model is "good," the probabilities of these negative words in the X position should be higher than those of a "bad" model, because we know these sentences are negative. The result in Table 4 verifies our assumption: there are obvious increases in probabilities across all 12 datasets. We observe the same *probability shift* tendency of positive words and please refer to Table 5 in Appendix for details.

4.5 Performance on Hard Sentiment Cases

Hard sentiment cases refer to documents which do not contain explicitly sentimental words such as *terrible* or *excellent*; For example, *This is a must-to-watch movie.* This sentence conveys quite positive sentiment despite it does include any explicitly sentimental words. These cases are much harder for models to make judgment and we are thus interested in the following question: how does our data augmentation method perform on these hard sentiment cases?

To verify this, we utilize a pre-defined positive list (V_{pos}) and negative list (V_{neg}) in Step 2 of Section 2.2 to filter each dataset. More specifically, we remove sentences from each dataset if any positive or negative sentiment words in V_{pos} or V_{neg} appear in the sentences. Table 6 shows the number of hard cases in each dataset. We observe that even in hard cases, our instruction-tuned models achieve significant improvements, further validat-

[12] We follow Gururangan et al. (2020) to consider word form.

[13] For aspect-based sentiment classification, we use *The sentiment of TARGET is <mask>* where TARGET is the aspect.

[14] For subwords, we multiply the probability of each subword, such that p("disappointing") = p("dis") × p("appointing"), to obtain the probability of the entire word.

	SST-2	Yelp	Amaz	IMDB	Deba	Airline	PB	14hap	14res	15res	15hot	16res	Ave.
# of cases	62	74	65	66	150	119	111	46	105	63	21	23	–
llama2-7b-base	35.5	43.4	36.5	37.6	18.0	19.5	22.7	23.0	35.4	32.9	34.8	34.1	31.1
llama2-7b-base+ours	**84.5**	**92.2**	**92.7**	**90.6**	**36.0**	**39.8**	**43.0**	**47.2**	**48.3**	**45.2**	**38.4**	**42.3**	**58.4**

Table 6: Accuracy performance comparison between llama2-7b-base and llama2-7b-base+ours for implicit (hard) cases across each dataset.

ing the generalization capability of the proposed method, especially when only explicit sentiment words are used as input for instruction tuning.

5 Related Work

Sentiment analysis has been a long established area of NLP research. Over the years, various domain-specific sentiment benchmarks have been proposed. In 2013, Socher et al. (2013) introduced the Stanford Sentiment Treebank benchmark, training a Recursive Neural Tensor Network for movie review classification. In 2014, Malo et al. (2014b) annotated news articles and created a financial benchmark for sentiment classification. In 2015, Kotzias et al. (2015) extracted sentences from three real-world customer review data sources: Amazon (amazon.com), IMDB (imdb.com), and Yelp[15]. They manually labeled 1,000 sentences from each source, with 500 positive and 500 negative sentences. On the other hand, Aspect-Based Sentiment Analysis (ABSA) has become increasingly important in practice. Between 2014 and 2016, the SemEval workshop introduced several ABSA tasks, resulting in SemEval-14, SemEval-15, and SemEval-16, which encompassed domains such as restaurants, hotels, and laptop PCs. On the other hand, sentiment adjectives have been studied in a separate line of research. Wiebe et al. (2000); Glass (2024) explore subjective adjectives, while Wiegand et al. (2013) focus on predictive adjectives. Much of the early research in sentiment focused on adjectives (Taboada et al., 2011). Incorporating these adjectives into the instruction tuning data is one of the key differences between our work and theirs.

With recent advancement of LLMs, many LLM-based models have been proposed for sentiment analysis and aspect-based sentiment analysis. Deng et al. (2023) leverage LLMs to generate financial sentiment labels to train a smaller model for sentiment analysis on social media content. Wang et al. (2023c) demonstrate that ChatGPT exhibits impressive zero-shot capabilities in sentiment classification, although it still lags behind domain-specific fully-supervised SOTA models. Recently, Zhao et al. (2023) utilized generic responses in dialogue corpora to debias LLMs for zero-shot sentiment classification. Scaria et al. (2023) proposed an instruction learning paradigm for ABSA, introducing positive, negative, and neutral examples to each training sample and fine-tuning the model for ABSA subtasks. Kanayama et al. (2024) further incorporated multilingual lexicon knowledge in LLMs to enhance sentiment classification performance. Our work differs from these approaches by not utilizing any actual training instances from sentiment benchmarks in the instruction construction.

6 Conclusion

In this work, we construct a small number of pseudo instructions to instruction-tune LLMs. The experimental result demonstrates significant performance gains over the base models on a wide range of sentiment benchmarks. Despite its simplicity, our method is supported by extensive analysis and ablation studies that highlight its effectiveness and advantages. Notably, it does not rely on ground-truth training instances from sentiment benchmarks and demonstrates superior generalization across diverse domains compared to domain-specific sentiment models. In future, we would extend our method to more fine-grained emotion classification.

7 Limitations

As we are focusing on classification task, the output space in this study is discrete, comprising positive, negative, neutral categories, rather than continuous. This approach is not suitable for scenarios that require continuous scoring, such as sentiment regression tasks. Also, while our method effectively handles general and aspect-based sentiment classification tasks, its ability to enhance more fine-grained sentiment classifications, such as 5-class classification, remains unclear. Further investigation and adaptation are required.

[15]https://www.yelp.com/dataset

Acknowledgments

We sincerely thank the reviewers and conference chairs for their valuable comments and suggestions.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani, et al. 2010. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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

Xiang Deng, Vasilisa Bashlovkina, Feng Han, Simon Baumgartner, and Michael Bendersky. 2023. LLMs to the Moon? Reddit market sentiment analysis with large language models. In *Companion Proceedings of the ACM Web Conference 2023*, pages 1014–1019.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332.

Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

Lelia Glass. 2024. The red dress is cute: why subjective adjectives are more often predicative. *Corpus Linguistics and Linguistic Theory*, (0).

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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)*, Toronto, Canada.

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

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Hiroshi Kanayama, Yang Zhao, Ran Iwamoto, and Takuya Ohko. 2024. Incorporating syntax and lexical knowledge to multilingual sentiment classification on large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4810–4817.

Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 597–606.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1317–1328, Toronto, Canada. Association for Computational Linguistics.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014a. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014b. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

Kevin Scaria, Himanshu Gupta, Siddharth Goyal, Saurabh Arjun Sawant, Swaroop Mishra, and Chitta Baral. 2023. Instructabsa: Instruction learning for aspect based sentiment analysis. *arXiv preprint arXiv:2302.08624*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023a. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada.

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

Zengzhi Wang, Qiming Xie, Yi Feng, Zixiang Ding, Zinong Yang, and Rui Xia. 2023c. Is chatgpt a good sentiment analyzer? a preliminary study. *arXiv preprint arXiv:2304.04339*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Janyce Wiebe et al. 2000. Learning subjective adjectives from corpora. *Aaai/iaai*, 20(0):0.

Michael Wiegand, Josef Ruppenhofer, and Dietrich Klakow. 2013. Predicative adjectives: An unsupervised criterion to extract subjective adjectives. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*.

Yang Zhao, Tetsuya Nasukawa, Masayasu Muraoka, and Bishwaranjan Bhattacharjee. 2023. A simple yet strong domain-agnostic de-bias method for zero-shot sentiment classification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3923–3931, Toronto, Canada. Association for Computational Linguistics.

A Instruction Text Collection Details

We herein describe filtering steps as follows and please refer to the code in supplementary material for the implementation:

- Filter 1: we extract all the instruction text (**T**) from five instruction datasets for all NLP tasks and yield 509,901 in total.
- Filter 2: as this work focuses on sentiment classification with a discrete output space, we designed several heuristics to remove irrelevant instruction text. Specifically, we kept instruction text only if it contained 'sentiment', 'positive', 'negative', or 'neutral'. Please refer to the code for the implementation. This process resulted in 974 instruction texts.
- Filter 3: we removed any repeated, word-level, or multi-sentence sentiment instruction texts. Additionally, to exclude sentiment regression-related instructions, we removed any instruction text containing strings such as 'sentiment score,' 'rating,' '1 for,' '0 for,' '-1 for,' etc. This process finally left us with 110 instruction texts.

B How the number of constructed instruction tuple affects performance?

We experimented with 50, 100, 150, and 200 instances to observe the performance change (we balance each class) using llama2-7b model. Table 8 shows the results: we observed that as the number of tuples increased, there was a consistent performance improvement. However, the growth rate of improvement slowed when the number of instances exceeded 200. Considering that further increases in instances yielded only marginal improvements (less than 1 point, from 200 to 240 tuples), we determined that 240 tuples provided a good trade-off between performance enhancement and computational efficiency.

C Additional experimental result with Yelp domain specific instances

Considering the promising results from domain-specific real instances of the Yelp dataset, we wondered if leveraging more instances would enhance performance. To test this, we instruction-tuned the llama2-7b model using the entire Yelp dataset, which contains 1,000 instances. Table 7 shows that the model trained on the full Yelp dataset performs perfectly on its own domain. However, the model shows significant performance drops compared to the models trained with fewer Yelp instances. We speculate that this is due to overfitting to the specific domain (i.e., Yelp); For domains closer to Yelp (i.e., restaurant), the performance drop is smaller (SemEval-14res, 15res, 16res); for those further from the Yelp domain, such as debate (Deba) and finance (PB), the drop is larger. We believe this highlights a distinct advantage of our method: **our domain-agnostic, adjective-based instructions avoid overfitting too much to specific domains, leading to better generalization across other domains.**

Dataset	SST-2 movie	Yelp resto	Amaz product	IMDB movie	Deba politics	Airline oper.	PB finance	14hap laptop	14res resto	15res resto	15hot hotel	16res resto	Ave. Ave.
Yelp (240)	**87.1**	92.3	**88.0**	**91.2**	**62.0**	**58.6**	**48.1**	**80.1**	**90.1**	**84.6**	**85.2**	**87.7**	**79.6**
Yelp (1,000)	84.4	**100.0**	85.1	85.2	48.9	56.0	40.8	76.5	89.9	81.0	80.5	86.2	75.8

Table 7: Performance of instruction-tuned llama2-7b using different amount of Yelp data. Best results are in bold.

Dataset	SST-2	Yelp	Amaz	IMDB	Deba	Airline	PB	14hap	14res	15res	15hot	16res	Ave.
base model	49.4	51.2	40.8	46.7	34.7	37.4	27.3	40.0	61.5	53.8	61.5	61.6	47.2
base + 50 tuples	68.1	72.6	63.7	69.0	54.1	59.1	62.2	53.4	58.5	60.5	49.2	52.2	60.2
base + 100 tuples	71.0	78.8	70.3	72.9	51.7	58.5	55.0	59.1	68.4	66	57.1	59.9	64.1
base + 150 tuples	78.6	87.4	80.7	80.8	58.2	**67.9**	**66.2**	75.2	79.8	76.1	65.6	73.6	74.2
base + 200 tuples	87.5	**96.2**	**94.8**	90.7	58.5	63.6	49.5	**84.8**	**90.8**	85.6	84.1	87.9	81.2
base + 240 tuples	**89.5**	96.1	94.1	**94.4**	**62.0**	67.6	53.5	82.1	88.4	**86.3**	**84.4**	**89.4**	**82.3**

Table 8: Performance of instruction-tuned llama2-7b using different number of constructed instruction tuples. Best results are in bold.

mode	source: instruction text (**T**) + input (**I**)	target: output (**O**)
Training	*Classify the given text into positive, negative or neutral sentiment.* \n *beautiful.* \n *Answer:*	positive
Training	*Classify the given text into positive, negative or neutral sentiment.* \n *dangerous.* \n *Answer:*	negative
Training	*Classify the given text into positive, negative or neutral sentiment.* \n *general.* \n *Answer:*	neutral
Evaluation	*Categorize the following sentence into either positive, neutral, or negative sentiment.*\n *A movie journey worth taking.* \n *Answer:*	[*LLM's prediction*]

Table 9: Constructed sentimental adjective-based tuple for training and testing. Note that there is no overlap between training instruction texts and testing instruction texts to make the evaluation out-of-the-box.

#	Instruction text sample
1	Analyze the content of the following text to determine whether it has a positive, negative or neutral sentiment *[with respect to the TARGET]*.
2	Categorize the following sentence into either positive, neutral, or negative sentiment *[with respect to the TARGET]*.
3	Classify the given text into positive, negative or neutral sentiment *[with respect to the TARGET]*.
4	Detect if the given text is positive, negative, or neutral in sentiment *[with respect to the TARGET]*. output one of these three labels for each input.
5	Find out the sentiment of the given sentence *[with respect to the TARGET]*. positive, negative, neutral.
6	Given a sentence, detect its sentiment *[with respect to the TARGET]*. possible outputs include: positive, negative, neutral.
7	In this task, you are given a sentence. Your task is to determine whether the sentiment in the sentence conveys either positive, negative or neutral emotion *[with respect to the TARGET]*.
8	Perform sentiment analysis and produce a label indicating whether the sentiment of given sentence is positive, negative, or neutral *[with respect to the TARGET]*.
9	What is the sentiment of the given statement *[with respect to the TARGET]*? (you should respond with one of these: "positive", "negative", "neutral").
10	You are given a sentence. Your task is to identify if the statement is positive, negative, or neutral *[with respect to the TARGET]*.

Table 10: 10 samples out of 110 from our collection of sentiment classification-related instruction text (**T**). Please note that when it comes to aspect-based sentiment classification task, we add *[with respect to the TARGET]* and replace TARGET with the specific aspect in the sentence.

positive words	negative words	neutral words
sophisticated	contemptible	last-ditch
magna-cum-laude	bogus	alate
gorgeous	salt	floored
boss	unfree	quadrilateral
heaven-sent	hidden	forty
exhaustive	inhumane	french-speaking
superb	humble	combined
healthy	false	client-server
...	...	...

Table 11: **Step 1. Collect sentimental adjectives candidates.**

positive words	negative words	neutral words
sophisticated	contemptible	alate
gorgeous	bogus	quadrilateral
superb	inhumane	forty
healthy	false	french-speaking
meticulous	precarious	combined
perfect	upset	client-server
sweet	numb	trojan
coherent	indelicate	diagonal
...	...	...

Table 12: **Step 2. Align with sentiment word sense.**

positive words	negative words	neutral words
best	dead	new
great	poor	more
important	difficult	national
good	unable	most
better	bad	many
supreme	wild	american
golden	cold	early
greatest	offensive	high
...	...	...

Table 13: **Step 3. Rank word by frequency.**

positive words	negative words	neutral words
best	dead	new
great	poor	more
important	difficult	national
good	unable	most
better	bad	many
supreme	wild	american
golden	cold	early
not offensive	not greatest	high
...	...	...

Table 14: **Step 4. Add negation words.**