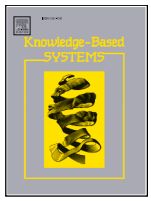
Knowledge-Based Systems 336 (2026) 115277

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Leveraging fine-tuning of large language models for aspect-based sentiment analysis in resource-scarce environments

Jakob Fehle [a,*], Udo Kruschwitz [b], Nils Constantin Hellwig [a], Christian Wolff [a]

[a] Media Informatics, University of Regensburg, University of Regensburg, Regensburg, 93040, Bavaria, Germany
[b] Information Science, University of Regensburg, University of Regensburg, Regensburg, 93040, Bavaria, Germany

ARTICLE INFO

Keywords:
Natural language processing (NLP)
Sentiment analysis (SA)
Aspect-based sentiment analysis (ABSA)
Instruction fine-tuning
Large language models (LLMs)
Low-resource settings

ABSTRACT

This study explores the use of fine-tuned open source large language models (LLMs) for Aspect-based Sentiment Analysis (ABSA), comparing their performance with state-of-the-art (SOTA) methods on English and German datasets with focus on low-resource scenarios. Results on the four ABSA subtasks Aspect Category Detection (ACD), Aspect Category Sentiment Analysis (ACSA), End-To-End-ABSA (E2E), and Target Aspect Sentiment Detection (TASD) show that fine-tuned LLMs handle limited training data scenarios better than current SOTA approaches, achieving consistent performance across various dataset sizes. Prompt formulation and hyperparameter tuning influence performance, though concise prompts often suffice when combined with effective fine-tuning. To assess generalizability, we conduct an ablation study across multiple languages, domains, and LLM architectures. The findings confirm that performance gains extend beyond the initial setting, supporting the robustness of fine-tuned LLMs over multiple different languages and domains. We establish new SOTA results on the Rest-16 and GERestaurant datasets and highlight the practical viability of fine-tuning LLMs for ABSA applications under limited training material.

1. Introduction

Aspect-based sentiment analysis (ABSA) is a crucial task in natural language processing (NLP) that focuses on identifying sentiments expressed towards specific aspects of entities within a text [1]. This granular approach to sentiment analysis provides more detailed insights compared to traditional sentiment analysis, which often considers the sentiment of an entire text as a whole [2].

ABSA plays a critical role across multiple application areas, including the analysis of customer feedback, product reviews, and social media data [3]. By extracting fine-grained sentiment information tied to specific aspects, ABSA enables businesses to better understand consumer opinions and identify areas for improvement [4,5]. Over the last years, the importance of ABSA in the business and industry domain has grown significantly [6], driven by its ability to provide valuable insights and enhance decision-making efficiency in increasingly competitive markets.

One significant challenge in ABSA is the availability of diverse, usable datasets for training robust models, especially in less researched languages such as German. While English benefits from numerous well-annotated datasets, progress in creating ABSA datasets for other languages has been much slower, with only a small number of new datasets emerging in recent years [7]. This limited availability poses challenges for both research, where it restricts the development and evaluation of advanced models, and practical applications, where it hinders the deployment of effective models for tasks like customer feedback analysis and market research. Traditional models often struggle in these environments due to their reliance on large, annotated datasets to capture the nuances of language and sentiment accurately [8]. To address this data scarcity, transfer learning has shown promise by leveraging knowledge from large, diverse datasets and applying it to specific tasks with limited data [9].

In recent years, with the rise of available large language models (LLMs) such as ChatGPT [10], the NLP research community has shifted towards using these models for various NLP tasks [11,12]. These models, when given well-crafted prompts, have demonstrated a high degree of adaptability and effectiveness, and ABSA is no exception [13,14]. LLMs such as GPT-4 [15], LLaMA 3 [16], or Mixtral [17] have revolutionized the field by offering powerful, pre-trained models that can be prompted or fine-tuned for specific tasks. In case of ABSA, fine-tuning allows these

* Corresponding author.
E-mail addresses: Jakob.Fehle@ur.de (J. Fehle), Udo.Kruschwitz@ur.de (U. Kruschwitz), Nils-Constantin.Hellwig@ur.de (N.C. Hellwig), Christian.Wolff@ur.de (C. Wolff).

https://doi.org/10.1016/j.knosys.2026.115277
Received 10 January 2025; Received in revised form 23 December 2025; Accepted 4 January 2026
Available online 8 January 2026

models to adapt more precisely to the specific aspects and sentiments within the target domain, enhancing their accuracy and effectiveness [18–20].

However, while commercially available LLMs such as ChatGPT offer robust performance and ease of use, they also come with several drawbacks, including concerns about data privacy and the high costs associated with extensive usage. In contrast, locally hosted, open source LLMs offer compelling advantages: greater control over sensitive data, full customization without external dependencies, and significantly lower long-term costs [21]. These benefits are particularly relevant in enterprise settings, where adaptability, cost efficiency, and data governance are critical. Recent work further highlights that open source LLMs (sometimes referred to as small language models when contrasted with commercial-scale systems)[1] are becoming increasingly competitive for real-world applications. Nevertheless, their reliability as replacements for proprietary LLMs and the need for systematic evaluation remain active areas of investigation [21]. While evaluations often focus on general NLP tasks [22,23], their performance in complex, structured problems like ABSA, remains largely underexplored.

Some recent studies have begun to address this gap: Šmíd et al. [20] fine-tuned LLaMA and Orca models for ABSA, focusing solely on the Target Aspect Sentiment Detection (TASD) task in English; Varia et al. [24] explored low-resource few-shot settings using a T5-based model; Gou et al. [25] proposed Multi-view-Prompting (MvP), a prompting-based method leveraging data augmentation and output template ordering for advanced ABSA (including low-resource scenarios); and Xu et al. [26] introduced DS^2-ABSA, a dual-stream data LLM-based synthesis framework with label refinement designed for few-shot ABSA. However, these studies are typically limited to English-language datasets and do not extend to cross-lingual or multilingual settings. Comprehensive evaluations of instruction-tuned open source LLMs for ABSA in genuinely low-resource scenarios, especially beyond English, remain scarce. This raises the question whether LLMs, given their extensive pretraining and generalization capabilities, can effectively address ABSA in various data-availability scenarios across different languages and task complexities.

Therefore, this means they could potentially transfer knowledge from their pre-training phase to new tasks and domains with even less additional data required. The ability of LLMs to understand and generate human-like text may make them more adaptable to different domains and languages, even with minimal fine-tuning data. This adaptability would be particularly advantageous in resource-scarce scenarios, where acquiring large, annotated datasets is impractical or impossible.

Motivated by these considerations, we aim to extend results of previous studies in fine-tuning LLMs by exploring how fine-tuning can enhance the performance of open source LLMs for ABSA, especially in less-resourced languages like German. Moreover, we investigate whether fine-tuned LLMs are less sensitive to smaller training dataset sizes than existing state-of-the-art (SOTA) methods by comparing results across various scenarios with established baseline models. Additionally, given the influence of the content of a prompt on the output produced by an LLM [27,28], we investigate three different prompting styles to determine the optimal formulation for different tasks and training dataset sizes.

To thoroughly evaluate the capabilities of fine-tuned LLMs, we utilize two datasets: SemEval-2016 restaurant reviews (EN-Rest) [29], a widely-used benchmark in ABSA research, and GERestaurant (DE-Rest) [30], which focuses on German-language restaurant reviews. These datasets provide annotations for aspect term, aspect category, and sentiment polarity, allowing us to focus on four widely recognized ABSA tasks: Aspect Category Detection (ACD), Aspect Category Sentiment Analysis (ACSA), End-To-End ABSA (E2E), and TASD [31]. By selecting these tasks, we ensure that our investigation aligns with the most established ABSA studies feasible with our datasets' annotations. To analyze the models' robustness under different resource conditions, we evaluate their performance across several data availability settings by limiting the total number of samples used during training and validation. Specifically, we experiment with the full dataset as well as subsets of 1,000, 500, and 50 labeled examples. These settings, especially the 50-samples condition, allow us to simulate data-scarce environments and assess the generalization ability of LLMs when supervision is minimal.

To further examine the generalizability and robustness of our findings, we conduct a complementary ablation study based on the most challenging 50-sample setting. This study extends our core experiments to additional languages, domains, and LLM architectures, allowing us to assess whether the observed trends hold under more diverse conditions and system setups.

Our study contributes a systematic investigation of fine-tuned open source LLMs for ABSA across four subtasks, two languages, and varying levels of data availability, including an extreme low-resource setting with only 50 labeled examples. We additionally compare performance with several baseline methods, including a few-shot prompting approach, and analyze training and inference efficiency to assess their suitability for real-world applications.

To structure our work, we focus on three research questions to provide a comprehensive evaluation of fine-tuned open source LLMs for ABSA:

RQ1: *How suitable are fine-tuned LLMs in solving the task of ABSA, and how can these results be placed in the context of current SOTA methods?*

RQ2: *Do fine-tuned LLMs adapt better to low-resource scenarios than existing SOTA methods?*

RQ3: *Does the type of prompt formulation have a significant impact on performance in fine-tuned LLMs?*

Our evaluation demonstrates that fine-tuned open source LLMs can deliver strong and robust performance across a range of ABSA subtasks and low-resource scenarios. Our approach establishes new SOTA results on the EN-Rest (Rest-16) dataset for ACSA (F1: 82.48) and E2E (F1: 81.77), and on the DE-Rest (GERestaurant) dataset for ACSA (F1: 85.45) and TASD (F1: 75.13). Through systematic hyperparameter tuning and prompt-style evaluation, we show that concise prompts are sufficient to reach high performance. In most settings, our approach delivers higher F1 scores than traditional and few-shot baselines and remains robust even under extreme data scarcity. These findings are further supported by an ablation study which focuses on extreme data scarcity and demonstrates that the advantages of fine-tuned LLMs hold across domains, languages, and model architectures, particularly for complex tasks where they outperform strong in-context learning (ICL) baselines. Our analysis on resource efficiency shows that fine-tuned models require only moderate resources for training and offer faster inference compared to baseline approaches, making them particularly suitable for real-time and high-throughput ABSA applications.

2. Related work

Recently, many LLMs like ChatGPT or Claude [32], along with open source alternatives such as LLaMA, Mixtral or Gemini [33] have emerged as powerful tools in NLP. The application of those LLMs across various NLP tasks has been a significant focus of recent research [11,12,34].

Commercial vs open source LLMs. A key distinction within the LLM landscape lies between commercial and open source models. Commercial models, such as OpenAI's ChatGPT, benefit from extensive computational resources, proprietary data, and frequent updates, offering ease

[1] Irugalbandara et al. [21] refer to models such as LLaMA as small language models (SLMs) relative to proprietary LLMs like GPT-4.

of use and broad applicability across multiple fields [35,36]. However, these systems are often closed, limiting transparency, reproducibility, and customizability while also raising concerns about data privacy. In contrast, open source models like LLaMA or Mixtral provide greater accessibility and flexibility, allowing researchers to fine-tune and adapt these models for specific applications while maintaining control over data and training workflows [37,38]. These benefits have made open source models increasingly popular in academic and applied research [39–41].

Evaluation of ChatGPT for sentiment analysis. However, LLMs, in general, both commercial and open source, have demonstrated broad applicability across various fields of NLP. This trend extends to sentiment analysis, where numerous studies have investigated the performance of these models across different scenarios, revealing both their potential and limitations. Several studies have evaluated ChatGPT for document-level sentiment analysis, finding that it achieves acceptable results but still lags behind SOTA methods [11,14,23,42]. Additionally, Zhang et al. [43] evaluated ChatGPT's sentiment extraction across six tasks, from document-level analysis to ABSA and Aspect Sentiment Quadruple Prediction (ASQP). They found that ChatGPT is effective in simpler tasks and resource-scarce scenarios due to their ICL capabilities. However, for complex tasks, ChatGPT was less effective compared to specialized fine-tuned models.

Language models in advanced sentiment analysis tasks. For more specialized applications, Wu et al. [44] focused on enhancing ASQP for Chinese by adapting English ASQP methodologies, achieving significant improvements. Similarly, Huang et al. [45] integrated multiple ABSA tasks into a unified generative framework, demonstrating robust performance across datasets in both fully supervised and few-shot learning settings. Additionally, Ding et al. [46] proposed a continual learning approach using LLMs for ABSA, achieving SOTA performance across 19 datasets by leveraging knowledge from multiple domains. Ahmed et al. [47] propose a DNN-driven Gradual Machine Learning (GML) approach for Aspect-term Sentiment Analysis (ATSA), which clusters automatically extracted features by sentiment orientation using an unsupervised neural network and models these clusters as factors in a factor graph, achieving SOTA results in both supervised and unsupervised setups. In a separate work, the authors further improved implicit ABSA by guiding BERT through auxiliary sentences derived from corpus semantics, leading to consistent performance gains across various ABSA benchmarks [48].

Instruction fine-tuning LLMs for ABSA. For instruction fine-tuning of LLMs, where the text input is formulated as a natural language task prompt, works like Scaria et al. [13] and Varia et al. [24] applied instruction fine-tuning on a T5 model, achieving SOTA performance in various ABSA subtasks. Building on this, Simmering and Huoviala [18] were first to fine-tune ChatGPT for ABSA, surpassing previous SOTA models such as InstructABSA [13]. Their work demonstrated that fine-tuning offers superior performance over few-shot approaches. They suggested that while prompt engineering is crucial for few-shot learning, it plays a lesser role in fine-tuning. They also highlighted the potential benefits of fine-tuning open source LLMs and investigating the impact of extended prompt engineering techniques like chain-of-thought (CoT) prompting. Extending this line of research, Šmíd et al. [20] evaluated LLaMA-based models for instruction fine-tuning and zero-/few-shot learning across various ABSA subtasks. They concluded that fine-tuned open source LLMs are highly effective in ABSA, often surpassing current SOTA approaches. However, they recognized limitations in their study due to focusing solely on datasets in English.

ABSA in resource-scarce scenarios. While low-resource environments have been studied to some extent in NLP [9,49] and sentiment analysis [50,51], relatively few works have specifically focused on ABSA under data-scarce conditions. In the early stages of ABSA research, many approaches in low-resource environments focused on creating new manually-annotated datasets [52–54], addressing the scarcity of annotated data needed for training robust models. More recently, however, the focus has shifted towards developing methods that can effectively operate with limited available data, leveraging advancements in generative models and prompting techniques. These include techniques such as self-consistency and the aggregation of multiple diverse prompts in few-shot learning scenarios [55,56], as well as data augmentation through varied output element orderings during fine-tuning [25,57]. For the fine-tuning of pre-trained models in low-resource settings, Hu et al. [57] explored the dynamic template orders Dataset-Level Order (DLO) and Instance-Level Order (ILO) for the ABSA elements of quadruplets (ASQP), demonstrating that diverse template configurations can significantly improve performance in resource-scarce scenarios. Similarly, Gou et al. [25] proposed MvP, a technique leveraging human-like problem-solving by aggregating sentiment elements generated in different orders, achieving SOTA performance across 10 ABSA datasets and excelling in low-resource conditions. Additionally, Hellwig et al. [58] utilized data augmentation and leveraged GPT-3.5-turbo and LLaMA-3-70B to generate annotated data for ABSA in resource-scarce settings. Their approach achieved notable F1 scores of 81.33 for ACD and 71.71 for ACSA in the restaurant domain, using only 25 human-annotated examples.

Summary. Recent advancements in generative approaches have brought significant improvements to sentiment analysis, showcasing the adaptability of LLMs across a range of tasks and scenarios. While these models perform well in simpler tasks and resource-scarce conditions, challenges persist for complex tasks like ABSA, where specialized fine-tuned models often outperform general-purpose LLMs. However, with advanced prompt engineering and fine-tuning, their performance can be significantly enhanced, making them valuable, particularly in resource-constrained environments. The recent success of fine-tuned open source models, such as those evaluated by Šmíd et al. [20], underscores the potential for LLMs to drive further advancements in ABSA, especially when expanded to include other languages, a broader range of tasks, and diverse application areas.

3. Methodology

In the context of this work, we analyze the suitability of a fine-tuned open source LLM for ABSA, evaluate which type of prompt formulation achieves the best performance, explore the impact of fine-tuning hyperparameters, and investigate the performance of LLMs under the conditions of limited training data, where the datasets are reduced to only 1,000, 500, or 50 examples using stratified sampling.

3.1. Our method

To provide a transparent overview of our approach, Fig. 1 illustrates the conceptual structure of our method. Starting from a labeled ABSA dataset, we generate prompts from the training split for instruction-based fine-tuning and prepare corresponding evaluation prompts from the evaluation split. Then, our approach, further referenced to as **LLaMA-FT-ABSA**, applies instruction-fine-tuning [59] and leverages sequence-to-sequence text generation in order to train a LLM on solving different ABSA tasks formulated as natural language instructions. Once training is complete, the fine-tuned model is evaluated on held-out data. Its output is parsed to extract task-specific ABSA elements, which are then compared against ground-truth annotations to compute performance metrics.

For fine-tuning, we employ Quantized Low-Rank Adaption (QLoRA) [60] using the python library *unsloth*.[2] QLoRA is a technique designed to

[2] https://github.com/unslothai/unsloth

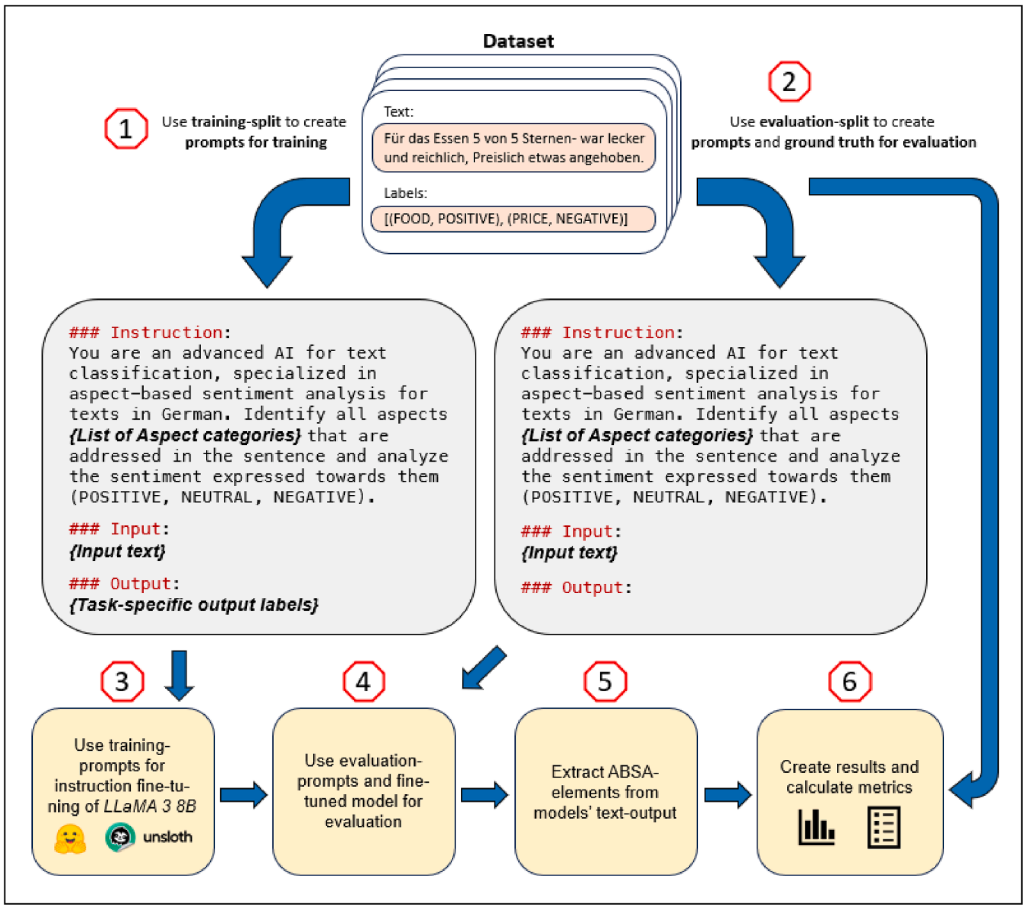


Fig. 1. Conceptual overview of our fine-tuning and evaluation pipeline for ABSA using instruction-based prompts. For the technical execution and evaluation procedure, see Section 4.4.

adapt LLMs efficiently to specific tasks by leveraging quantization and Low-Rank Adaptation (LoRA) [61]. Quantization reduces the model's precision, which significantly reduces the computational resources required, while LoRA introduces task-specific adaptation by adding low-rank matrices to the model's weights. This combination allows efficient and effective fine-tuning even with limited computational resources. For the evaluation of the fine-tuned models, quantization was no longer applied, instead, the LoRA adapters are merged with the original models during inference using the library *vLLM* [62]. We base our fine-tuning efforts on *LLaMA 3 8B* [63], which is motivated by the fact that the LLaMA model family serves as the foundation for a significant portion of public open source models, thereby covering the base architecture of the majority of available models. Fine-tuning and model evaluation are conducted on a workstation equipped with a Nvidia RTX A5000 GPU with 24 GB of VRAM.

3.2. Prompting strategy

Considering the influence of a prompt's content on the output produced by an LLM [27,28] and the recognition of prompt engineering as a critical technique for adapting LLMs to downstream tasks [64], we investigate three different prompting styles to determine the optimal formulation for various tasks and training dataset sizes. The general structure of our prompts is based on experiments by Zhang et al. [43]. Our selection is further motivated by findings from Simmering and Huoviala [18], who compared different prompt formulations (ranging from minimal to detailed instructions) for GPT-based models in ABSA. Inspired by their setup, we extend this line of investigation to a broader range of ABSA tasks and datasets, using a fine-tuned open source LLM. The prompting styles are further referenced as *Basic* (short prompt), *Context* (*Basic* prompt with additional task-related context) and *CoT* (*Context* style prompt with *CoT* style output).

An example of the *CoT* prompt for the TASD task on the DE-Rest dataset can be seen in Fig. 2, and similar prompts for the ACD, ACSA, E2E and TASD tasks for both *Basic* and *Context* prompts are depicted in Appendix A.1. An additional example for the EN-Rest dataset is provided in Appendix A.2. The only difference between the task-specific prompts across datasets lies in the specific descriptions and explanations of the respective aspect categories, which are derived from the official annotation guidelines of the respective datasets. Building upon the prompt design principles outlined by Amatriain [65], we structure our prompts in English using specific tags to create distinct sections: task description (`### Instruction`), text input (`### Input`) and expected output of the LLM (`### Output`). Furthermore, all instructions and descriptions of a prompt are adapted depending on the ABSA task. The contents of the prompts are as follows:

Basic An introduction to the task with task-specific information and a list of aspect categories and sentiment polarities which have to be extracted.

Context In addition to the contents from the *Basic* prompt, we provide brief explanations for each aspect and polarity label to improve the LLM's understanding of the aspects' context, similarly as Wang et al. [66] investigated in the context of sentiment phrases. Additionally, we specify the desired output of the model within the prompt.

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
German. Extract all (aspect category, sentiment polarity, aspect phrase) triples of a sentence by identifying
all the aspect categories addressed with their corresponding phrases and analyzing the sentiment expressed
towards each aspect. Return a list of triples, each containing three strings in parentheses. If an aspect is
implied but not explicitly stated, identify its aspect category and sentiment and assign the aspect phrase
"NULL". Return an empty list if no aspects are addressed in the sentence. Return only the list, without any
further comments or text.

* Consider the following aspects: [FOOD, SERVICE, PRICE, AMBIENCE, GENERAL-IMPRESSION].
* Consider the following sentiment polarities: [POSITIVE, NEUTRAL, NEGATIVE].

FOOD refers to the food in general or specific dishes and drinks. SERVICE includes ratings on the service in
general, the attitude of the staff, waiting times or other services such as takeaway. PRICE relates to
opinions on the general pricing level or prices of food, drinks or other restaurant services. AMBIENCE refers
to the atmosphere inside and outside the restaurant, the facilities and the general noise level in the
restaurant. GENERAL-IMPRESSION includes opinions on the restaurant as a whole, without focus on the aspect
categories mentioned.
The labels POSITIVE, NEUTRAL and NEGATIVE describe the positive, neutral or negative sentiment expressed
towards the aspect.

### Input:
Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.

### Output:
Lets do this step by step. We would like to extract all aspect-sentiment-phrase-triples from the following
sentence: "Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.". First, we
identify all aspects addressed in the sentence and their corresponding phrases: The aspect FOOD is referenced
with the phrase "Essen" while the aspect PRICE  is inferred from the context of the sentence and is therefore
referenced without a phrase. We thus assign its phrase the value "NULL". Next, we determine the sentiment
expressed towards these aspects: The aspect FOOD is referenced positively and the aspect PRICE is mentioned
negatively in the text.
The final result thus consists of the following aspect-sentiment-phrase-triples: [(FOOD, POSITIVE, "Essen"),
(PRICE, NEGATIVE, "NULL")]
```

Fig. 2. Example of a *Chain-of-Thought (CoT)* prompt used in the TASD task for the sentence "*For the food 5 out of 5 stars - was delicious and plentiful, slightly higher in price*". While *CoT* and *Context* prompts use the same input, they differ in the structure of the expected output. Additional prompt examples can be found in Appendix A.1.

Table 1
Illustration of the ABSA subtasks used in this study, each with available input and expected output of the respective task. Example sentence is: "*It has great sushi and even better service.*. Abbr.: c = aspect category, p = sentiment polarity, a = aspect term."

Subtask	Input	Output	Example
Aspect Category Detection (ACD)	S	c_1, c_2	[FOOD#QUALITY, SERVICE#GENERAL]
Aspect Category Sentiment Classification (ACSA)	S	(c_1, p_1), (c_2, p_2)	[(FOOD#QUALITY, POSITIVE), (SERVICE#GENERAL, POSITIVE)]
End-To-End ABSA (E2E)	S	(a_1, p_1), (a_2, p_2)	[("sushi", POSITIVE), ("service", POSITIVE)]
Target Aspect Sentiment Detection (TASD)	S	(a_1, c_1, p_1), (a_2, c_2, p_2)	[("sushi", FOOD#QUALITY, POSITIVE), ("service", SERVICE#GENERAL, POSITIVE)]

CoT We use the same instruction text as for the context prompt style, but reformulate the answer into a step-by-step solution to the task (c.f. Wei et al. [59] and Zhou et al. [67]). This method is designed to help the LLM logically break down the task, potentially leading to more accurate and thorough responses [14].[3]

In contrast to the *CoT* prompt, the outputs for the *Basic* and *Context* prompt styles consist only of predefined lists of strings (see Table 1 for example outputs per task). Since both datasets contain implicit aspects without extractable phrases, we instructed the LLM to assign the phrase 'NULL' to these implicit aspects for the E2E and TASD tasks.

Considering that Simmering and Huoviala [18] have shown that the inclusion of few shot examples has no positive effect on the performance of fine-tuned LLMs in the field of ABSA, we have refrained from including them in our prompts.

[3] As the ACD task is sufficiently straightforward to be designed as a single-step task, we have not used the CoT prompt in this case.

4. Experimental setup

4.1. Datasets

To evaluate the effectiveness of LLMs for ABSA, we utilize two different datasets from the restaurant domain: SemEval 2016 (Task 5, restaurant domain; further referenced as EN-Rest) and GERestaurant (further referenced as DE-Rest). The EN-Rest dataset is a benchmark corpus extensively used in ABSA research [68]. It comprises 2295 English sentences (train: 1,708; test: 587) annotated across 12 aspect categories. This dataset has been pivotal in advancing ABSA methodologies, providing a standardized platform for evaluating model performance [13,25,43,57,69–74]. Complementing this, the GERestaurant dataset offers a substantial resource for ABSA in the German language. It contains 3078 sentences (train: 2,154; test: 924) annotated over 5 aspect categories. Utilizing both datasets allows for a comprehensive evaluation of LLMs' capabilities in ABSA across different languages and resource settings. The Rest-16 dataset provides a well-established benchmark for English, while GERestaurant facilitates assessment in a resource-scarce,

non-English context. Both datasets include annotations for aspect categories, sentiment polarities, and aspect terms. They cover both explicit and implicit aspects, allowing for the investigation of a wide range of ABSA subtasks.

4.2. Tasks

ABSA involves several subtasks that focus on different parts of aspect-based sentiments within texts. In this work, we investigate the four most common ABSA tasks feasible with the annotations of our datasets: Aspect Category Detection (ACD), Aspect Category Sentiment Analysis (ACSA), End-To-End ABSA (E2E), and Target Aspect Sentiment Detection (TASD)[4] [31] - see Table 1 for an illustration of exemplary inputs and outputs of each task.

While ACD solely aims to extract all aspect categories of a given text, ACSA also captures the sentiment polarities expressed towards each aspect. Target Aspect Sentiment Detection (TASD) takes ACSA a step further by also detecting the specific phrases or expressions that represent the identified aspects in the text. This involves identifying the aspect categories, analyzing the sentiment polarity for each category, and extracting the exact phrases in the text that correspond to each aspect and are the target of the expressed sentiment. End-To-End ABSA (E2E) represents the middle ground between ACSA and TASD, where sentiment is predicted toward a textual aspect phrase rather than a predefined category. In existing literature, evaluation protocols vary for this task, some include implicit aspects (those not explicitly mentioned in the text), while others focus exclusively on explicit targets. To ensure broad applicability and align with recent generative approaches capable of modeling both types, we explicitly include implicit aspects in our dataset and evaluation. Specifically, models must predict when no aspect phrase is present (i.e., implicit targets) and correctly assign the corresponding sentiment. This setup is used consistently across all E2E evaluations unless otherwise stated.

4.3. Parameters for fine-tuning

As with fine-tuning for a task in the context of transfer learning, there is also a large number of parameters that can influence the result when instruction-tuning LLMs [61,75]. We investigate four of these parameters in detail: LoRA rank, LoRA α, learning rate, and the number of training epochs.

LoRA α and LoRA rank. LoRA rank controls the dimensionality reduction in the model's weight matrices during adaptation, with higher ranks capturing more information but requiring more computational resources. LoRA α regulates the scaling of these low-rank matrices, influencing their impact on the original model parameters [61]. While general recommendations exist for LoRA α and rank values [61,76], these parameters are less explored in the context of ABSA.

Ding et al. [46] found that a rank of 8 yielded the best results for fine-tuning a LLaMA model on ABSA subtasks. Building on this, we use a rank of 8 as a starting point and explore different scaling factors, as higher scaling may enhance performance [76]. This leads to testing combinations of rank $= 8$; $\alpha = 8$ and rank $= 32$; $\alpha = 32$ for standard scaling, and rank $= 8$; $\alpha = 16$ and rank $= 32$; $\alpha = 64$ for increased weighting on the LoRA adapters.

Learning rate and number of training Epochs. The learning rate determines the magnitude of adjustments to the model parameters during training and is closely influenced by the batch size and the number of training epochs, as these factors collectively impact the model's behavior and stability. Based on previous research, we test constant learning rates of 3×10^{-4} and 3×10^{-5}: Wu et al. [44] used 3×10^{-4} for fine-tuning an LLM with LoRA adapters on ABSA tasks in Chinese, while Huang et al. [45] applied 3×10^{-5} for instruction tuning an LLM for ABSA. By exploring these rates, we aim to identify effective learning rates for fine-tuning LoRA adapters on our specific dataset and language contexts. To allow sufficient task adaption even in low-resource scenarios, all models are trained for up to 10 epochs. However, instead of relying on a fixed final epoch, we evaluate task-specific performance after each epoch using the micro-averaged F1 score and select the checkpoint with the best validation performance for all subsequent evaluations on the test set. This ensures that our results reflect the most optimal point during training.

Additional configurations used for LoRA fine-tuning are described in Appendix A.4.

4.4. Evaluation procedure

Our study follows a systematic three-step process to fine-tune the LLM and evaluate its performance on ABSA tasks (see Fig. 3). To maintain comparability across multiple datasets and splits while ensuring reproducibility and robustness, we emphasize a structured evaluation design, reflecting best practices from prior ABSA and NLP research [77–79].

Step 1: hyperparameter optimization. First, we identify the optimal hyperparameters for each prompt style across different dataset sizes and ABSA tasks. Therefore, the training datasets are divided into six stratified splits, with five used for training and one for validation (see Table A.9 in the Appendix for the size of each split per setting). The validation split is used exclusively for model selection during hyperparameter optimization, with task-specific performance measured by the micro-averaged F1 score serving as the primary selection criterion. This approach allows us to evaluate combinations of LoRA parameters (α and rank), learning rates, and training epochs, ensuring effective fine-tuning of the models across varying conditions.

For the hyperparameter optimization, we examine a total of 44 configurations per dataset (4 ABSA tasks $\times$ 4 dataset sizes $\times$ 2/3 prompt styles) and investigate 8 different hyperparameter combinations for each scenario (comprising 4 LoRA settings $\times$ 2 learning rates).

Step 2: cross-validation for prompt selection and performance dependency on dataset size. Using the optimal hyperparameters from Step 1, we perform 5×5 cross-validation on the training subsets, excluding the validation split to avoid data contamination. This process provides a robust average performance across multiple runs. Our objectives are to identify the best prompt formulation for each dataset, size, and ABSA task and to compare the LLMs' performance against baseline approaches in various resource-scarce scenarios. We also analyze how the dataset size available for training, validation and testing (full dataset, 1,000, 500, and 50 examples) influence LLM performance, ensuring consistent evaluation by using the full dataset's test splits across all sizes.

Step 3: comparative evaluation. Finally, we train the LLM on the entire DE-Rest and EN-Rest training splits using the best hyperparameters and prompting strategies from previous steps and evaluate the models on their respective original test splits. This final evaluation allows us to compare our results with other studies on the same datasets, providing a comprehensive assessment of our approaches' performance for ABSA tasks.

Evaluation settings. We use greedy-search with temperature $= 0$ for model evaluation to ensure that the outputs remain deterministic and reproducible. The output of the LLM is controlled with stop words, which prevents an uncontrolled continuation of the output as soon as unwanted parts of the prompt are repeated, such as the instructions or the input sentence. The aspect and sentiment outputs are then checked for validity and extracted using regex expressions to parse the ABSA elements

[4] TASD is also referred to as Aspect Category Sentiment Detection (ACSD). To maintain consistency with recent literature, we use the term TASD.

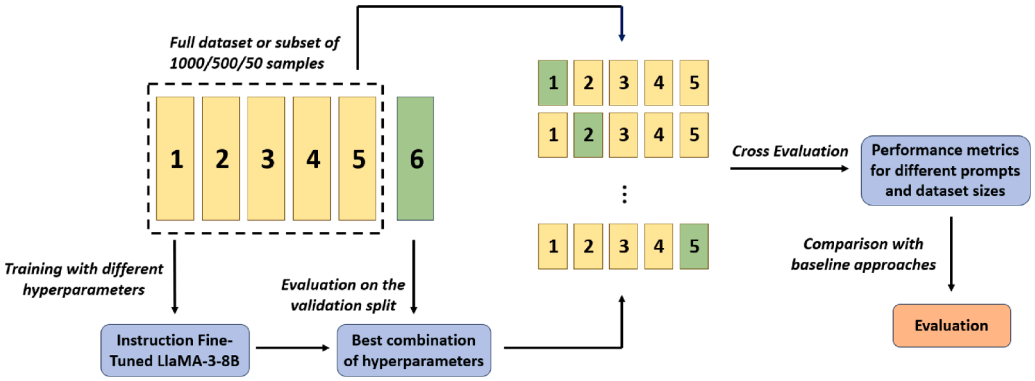


Fig. 3. Exemplary process of steps 1 (hyperparameter optimization for each prompt and dataset size) & 2 (cross-evaluation for prompt selection and baseline comparison) of the study design.

from the tuple-template formatted output (see Table 1) before evaluating model performance.

Limitations of the 50-sample setting. Since we cannot rely on a validation-based selection strategy for the 50-samples setting, we adopt the configuration of each approach as defined and evaluated in related work (see Appendix A.3). Splitting a meaningful validation set from such limited data (a potential validation set would consist of only ≈10 examples) is impractical, and hyperparameter tuning in this context would yield unreliable results.

Importantly, hyperparameter tuning is typically intended to maximize the final performance of individual models. However, for the 50-samples setting, our aim is not to optimize each model's performance, but to assess relative differences between methods under consistent and realistic low-resource conditions. We intentionally fix such factors to ensure comparability across tasks, languages, and domains, which is crucial for drawing robust conclusions about generalization behavior.

This experimental design mirrors real-world low-resource scenarios, such as those reflected in the RAFT benchmark [80], where datasets are similarly small and lack validation splits. Therefore, to ensure fairness and interpretability, we use fixed hyperparameter settings based on recommendations from the literature. This has proven to be more reliable than choosing arbitrary or insufficiently evaluated hyperparameters [27].

Therefore, to ensure comparability across languages, domains, and model types, we adopt values from prior work [18,44,46,61]: LoRA rank and α set to 8, learning rate to 3e-4, and a standard prompting template for ICL and fine-tuning methods (*Basic* template), while training for 10 epochs due to the small dataset size.

4.5. Metrics and empirical evaluation

Similar to previous studies, we rely our evaluation on the micro-averaged F1 score [29]. We provide other metrics such as macro-averaged F1 score, precision, recall and accuracy on GitHub.

We employ multiple statistical tests to assess the significance of observed differences in parameter combinations, hyperparameter settings and dataset sizes ($p_{\text{adj}} \leq 0.05$). For hyperparameter tuning, we use bootstrapping [81] (n = 1,000) to estimate performance metrics and apply the Kruskal-Wallis test [82] with Bonferroni-Holm correction [83] for pairwise comparisons. In evaluating prompting styles and dataset sizes, we test for normality using the Shapiro-Wilk test [84]. For normally distributed data, we use repeated measures ANOVA followed by paired t-tests for pairwise comparisons [85]. When normality is not assumed, the Friedman test is applied, followed by Wilcoxon signed-rank tests [86], with Bonferroni-Holm correction ensuring reliability.

4.6. Baselines

To assess the performance of our approach, we implement several baselines. We utilize various SOTA methods from the restaurant domain to obtain comparative results under our experimental conditions. The baseline methods are selected based on the results they achieved on the respective dataset they are used for, whereby we only considered approaches that are reproducible with provided code and can therefore be evaluated under our specific conditions.

For the ACD and ACSA subtasks, we implement two baseline models:

BERT-CLF A BERT-based multi-label classification approach, as used by Hellwig et al. [30], where one or more classes within a sentence are predicted. Each class represents either an aspect category, such as `FOOD` for ACD, or an aspect category combined with its positive, neutral or negative sentiment, such as `FOOD:POSITIVE` for ACSA.

Hier-GCN An approach based on hierarchical graph convolutional networks with BERT for sentence encoding by Cai et al. [87].

For the E2E task, we use the following baselines, which are able to handle explicit and implicit opinion targets:

InstructABSA A combined approach based on generative models by Scaria et al. [13] which solves term extraction and sentiment classification in a unified way.

TAS-BERT A BERT-based approach for a combined extraction of both aspect terms and aspect categories for sentiment classification [72].

For the TASD task, we implement the following baseline models:

Paraphrase A sentence paraphrase approach proposed by Zhang et al. [70] and adapted for the TASD task in German by Hellwig et al. [30], which utilizes sequence-to-sequence modeling to convert sentences into a predefined template before aspect extraction.

MvP Multi-View-Prompting (MvP) leverages element order prompts to generate sentiment tuples in multiple orders and aggregates results through a voting mechanism to capture interdependencies [25].

For all ABSA subtasks, we additionally include a baseline based on instruction-based in-context learning (ICL) without fine-tuning:

LLaMA Few-Shot A prompting-based approach using ICL without any gradient updates. Building upon prior research that successfully integrated task-relevant contextual information into few-shot prompts [18,20], we re-use our *Context* prompt template

for few-shot experiments and embed 5, 10, or 25 few-shot examples directly into the prompt. The choice of 5 and 10 examples is motivated by prior work [43,55], which demonstrated improved performance on complex ABSA tasks such as TASD with more in-context examples. Moreover, we include a 25-shot variant to further investigate this assumption. This setup enables a direct comparison between our fine-tuned LLM approach and purely prompt-based inference.

Each method is reproduced using its original configuration, reusing the same hyperparameters, model variants, and sizes wherever possible to avoid relying on suboptimal or untuned settings. If a method relies on an underlying pre-trained model, we use the original model. In cases where the pre-trained model is language-specific and not suitable for German, we select an equivalent model in the appropriate language.

For our fine-tuning baselines, given that several of our experimental conditions involve training with reduced dataset sizes, we want to ensure fair comparisons with the baseline models without delving into extensive hyperparameter tuning. Therefore, to potentially improve the baselines methods performance on the reduced datasets, we implement two different configurations: First, we train the models using the default number of steps or epochs for all dataset sizes, thus the settings with artificially reduced dataset sizes are trained over a smaller amount of total samples. Second, we increase the number of training steps or epochs so that the settings with reduced training datasets are trained over the same total number of samples or steps as the full dataset. This approach allows us to assess whether extended training on smaller datasets can compensate for the reduced amount of data and provides a more comprehensive comparison with our fine-tuned LLM approach for ABSA. The superior training approach for the baseline methods is chosen based on its results achieved on the separate validation set.

In the 50-samples setting, the same limitations apply to the baseline approaches as outlined for our fine-tuned LLMs in Section 4.4. We therefore refrain from validation-based hyperparameter tuning and instead adopt recommended configurations from prior literature to ensure fair and consistent comparisons in this extreme low-resource scenario.

5. Results and discussion

For transparency and reproducibility, our GitHub repository[5] provides detailed result tables for all experiments, including per-run statistics and breakdowns by aspect category and sentiment polarity. Additional metrics such as macro F1, precision, recall, and accuracy are also included.

5.1. Hyperparameter optimization

Detailed results and the optimal hyperparameter combinations for each experimental setting are reported in Table A.10 in the Appendix and are additionally available in our digital repository. The hyperparameters identified during this tuning phase are subsequently fixed and used for all following evaluations under comparable conditions (e.g., comparison of prompting styles and final performance evaluation). For the DE-Rest dataset, statistically significant hyperparameter combinations were identified for 27 out of 33 configurations, with the remaining 6 configurations significantly outperforming 6 out of 7 alternative setups. Similarly, for the EN-Rest dataset, statistically significant combinations were found in 28 out of 33 configurations, while the remaining 5 configurations also demonstrated superior performance compared to 6 out of 7 alternatives. As an average across both datasets, all prompt styles and dataset sizes, the combination of a learning rate of 3×10^{-4}, a LoRA rank of 8, and a LoRA α of 8 consistently emerged as the best-performing hyperparameter configuration. Additionally, the analysis reveals that configurations with a stronger weighting of the LoRA adapter, characterized by a higher LoRA α than LoRA rank, perform worse on average compared to those with equal weighting.

5.2. Prompt formulation

The results for the cross evaluation of the various prompt styles in their respective best parameter combination (see Table A.10 in the Appendix for best combinations) are shown in Table 2. *Basic* prompts perform comparably well on both datasets, especially in the ACD and TASD tasks, generally staying slightly ahead or sometimes slightly behind *Context* prompts by a small margin. In two cases, a statistically significant best result is identified for the *Basic* prompt (marked with †). Prompts based on the CoT methodology usually perform worst, with one minor exception on the ACSA task. This weakness is likely related to the fact that CoT formulations elicit longer reasoning-style outputs, which misalign with the label-focused fine-tuning objective.

Impact of task complexity. Regardless of prompt formulation, the obtained F1-micro values decrease with increasing task complexity (the more ABSA elements have to be jointly predicted), where performance peaks at 85.21 (ACD), 80.72 (ACSA), 80.07 (E2E), and 75.65 (TASD) on the EN-Rest dataset, and at 87.88, 84.40, 80.58, and 75.61 respectively on the DE-Rest dataset. These findings are consistent with prior research, which has repeatedly shown that more complex ABSA tasks (such as E2E and TASD) introduce compounding challenges due to the need to jointly model multiple, interdependent subtasks like aspect extraction and sentiment classification [13,43,72,88].

Impact of a prompt's integrated information on performance. Similar to Bai et al. [55], we analyze performance at the level of individual ABSA elements to better understand how prompt formulations affect model behavior. The results show that adding more ABSA elements, such as aspect terms, into a prompt can improve the performance of predicting other ABSA components (e.g., aspect category). Although, this effect varies by task and dataset. For the classification of aspect categories, more informative prompts (e.g., prompts for ACSA or TASD) lead, on average, to higher F1 scores on EN-Rest (+1.75 for using TASD over ACD and +0.43 for using TASD over ACSA) and DE-Rest (+1.10 for using TASD over ACSA). In aspect + polarity classification, the effect of using TASD instead of ACSA prompts is smaller or inconsistent (-0.08 on EN-Rest and +0.22 on DE-Rest).

Our results suggest that while richer prompts can improve performance, especially in aspect classification, the benefits for more complex tasks like polarity classification are smaller and inconsistent. Since the role of additional prompt information is not yet fully understood, future work should investigate under which conditions it meaningfully contributes to model performance.

Influence of prompt styles on hyperparameter tuning. Despite the variability across different datasets and parameter combinations, we observe that for both datasets different prompt styles significantly influence the optimal hyperparameters, underlining the need for prompt and dataset-specific tuning.

Prompt robustness and performance differences. Cross-evaluation with best-performing hyperparameters shows that differences between *Basic* and *Context* prompt styles are minimal, with significant differences occurring in only 2 out of 32 comparisons. This suggests that prompt formulation is less critical for the final performance outcomes of LLM fine-tuning in ABSA. At the same time, we note that prompt styles can still influence the choice of optimal hyperparameters, which underlines the importance of tuning, even if the relative ranking of prompts remains unchanged. Overall, this advocates for short and concise prompts to enhance efficiency and reduce training times. A similar observation was made by Simmering and Huoviala [18], who systematically varied the contextual information included in prompts during LLM fine-tuning

[5] https://github.com/JakobFehle/Fine-Tuning-LLMs-for-ABSA

Table 2
Results for the cross-validation setting of prompt style and dataset size evaluation as a micro F1 score averaged over five splits. Best results are in bold - results are marked with † if they are significantly better than both other prompts. Prompt styles include: *Basic* (minimal instruction), *Context* (additional task descriptions and label explanations), and *CoT* (step-by-step reasoning).

Task	Method	EN-Rest				DE-Rest			
		Full	1,000	500	50	Full	1,000	500	50
ACD	Basic	84.41	**81.46**	**81.15**	73.36	**87.87**	**86.65**	**86.12**†	80.86
	Context	**85.21**	79.74	78.72	73.37	87.83	86.45	83.24	**82.24**
	CoT	–	–	–	–	–	–	–	–
ACSA	Basic	**80.72**	79.32	**77.63**	**67.27**	84.11	**81.24**	80.38	76.61
	Context	80.70	**80.18**	76.70	68.67	**84.40**	79.77	81.15	**76.72**
	CoT	79.14	79.64	75.18	67.04	82.22	80.20	**81.38**	73.33
E2E	Basic	**80.07**	**77.98**	**73.93**	63.21	**80.58**	**78.43**	**73.85**†	**63.34**
	Context	79.95	76.60	73.39	**63.71**	78.86	78.42	71.09	62.81
	CoT	78.01	58.33	60.22	51.97	76.32	74.53	69.51	47.36
TASD	Basic	**75.65**	**73.10**	**71.46**	**54.10**	75.12	**74.03**	70.60	**59.54**
	Context	74.95	72.99	70.40	52.86	**75.61**	73.75	**72.86**	58.11
	CoT	70.17	69.46	62.50	47.39	71.80	68.23	67.80	48.03

Table 3
Mean and standard deviation (std) of performance differences in F1-micro scores between prompts across all hyperparameter combinations. This aggregation controls for tuning variance and highlights the relative impact of prompt design on model performance across tasks. *Basic* and *Context* differ only marginally, while *CoT* consistently underperforms and exhibits higher variance.

Task	Basic → Context		Basic → CoT		Context → CoT	
	Mean	Std	Mean	Std	Mean	Std
ACD	+0.27	1.61	–	–	–	–
ACSA	–0.21	4.96	–2.79	4.77	–2.58	5.88
E2E	–1.36	6.10	–11.00	8.42	–9.63	10.03
TASD	–0.15	3.76	–9.59	6.29	–9.44	6.95

Table 4
Mean and standard deviation (Std) of F1-micro scores within each prompt across all hyperparameter combinations and both datasets. *Basic* is consistently the most stable prompt type, while *Context* and *CoT* exhibit higher variance in complex tasks, with CoT reaching the highest values overall.

Prompt	ACD		ACSA		E2E		TASD	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Basic	82.49	5.15	77.80	8.36	73.65	6.06	69.69	7.98
Context	82.85	4.96	77.71	9.73	72.29	11.91	69.50	8.10
CoT	–	–	75.13	10.47	62.67	12.81	60.12	10.80

and evaluated its effect on ABSA performance. Like our findings, their results indicate that minimal prompts tend to be particularly effective, suggesting that extensive contextualization is not necessarily beneficial in this setting.

During hyperparameter tuning, the relative ranking of prompts remains consistent with the cross-evaluation results. Here, differences were computed pairwise for identical hyperparameter configurations, meaning that Table 3 reports averaged run-level deltas. This provides a more fine-grained view on prompt-hyperparameter interactions. As shown in Table 3, *Basic* and *Context* differ only marginally, with mean deltas close to zero and low variance. In contrast, *CoT* consistently underperforms, with average differences of up to -11 F1-micro points compared to *Basic*, and also exhibits higher variability across hyperparameter settings. This indicates that while prompt-hyperparameter interactions exist, they do not change the overall pattern: *Basic* and *Context* remain the most reliable choices, whereas *CoT* introduces instability.

We attribute these discrepancies primarily to differences in output format, as already discussed above. While *Context* prompts elicit short, label-based answers aligned with the training objective, *CoT* prompts lead to longer, natural-language style outputs that reflect reasoning steps which misalign with the model's fine-tuning objective. This contrasts with prior work reporting benefits of CoT in ICL scenarios [14,59], and suggests that CoT's advantages do not necessarily transfer to full fine-tuning setups where models can internalize task logic without explicit step-by-step reasoning. All results are publicly available in our repository, enabling future work with further analysis and replication.

Our variance analysis further supports these findings by showing that performance within each prompt type remains relatively stable across hyperparameter configurations (see Table 4), particularly when compared to the higher fluctuations of *CoT* prompts. *Basic* and *Context* prompts are comparatively robust, with standard deviations typically in the range of 5 - 8 F1-micro points. In contrast, *CoT* prompts fluctuate more strongly, reaching up to 12.8 F1-micro points for E2E and 10.8 for TASD. We also observe that variance increases across all prompt types as task complexity grows, from relatively low values in ACD to markedly higher values in E2E and TASD, suggesting that more complex subtasks amplify the effect of hyperparameter choices. Taken together, these analyses reinforce the cross-evaluation results: *Basic* and *Context* provide consistently stable performance, whereas *CoT* remains less reliable, particularly in complex tasks.

Manual vs. automatic prompt design. All prompt templates in our study were manually created and adjusted based on the specific ABSA task and dataset. While this ensured precise task adaptation, it may limit reproducibility and scalability to other languages, domains and datasets. Automatic prompt generation methods such as AutoPrompt [89] have already shown promising results in sentiment analysis with smaller language models. More recent approaches leveraging LLMs for prompt generation [90,91] or reinforcement learning-based optimization [92] further highlight the potential of automating prompt design. Exploring such techniques for ABSA could reduce manual effort and improve cross-domain generalizability.

5.3. Performance based on dataset size

The values achieved by the fine-tuned LLM remain relatively stable when the amount of training data is reduced (see Table 5). From the full dataset down to 500 examples, F1 scores typically decrease by only 2-4 points, indicating strong robustness under moderate data scarcity. However, in the extreme low-resource setting of only 50 training samples, more substantial performance drops are observed. The most substantial decrease occurs in the E2E task on EN-Rest, with a decline of 16.4

Table 5
Results for the cross-validation of dataset sizes in comparison with baseline approaches as a F1-micro score averaged over five splits. Best results are in bold - results are marked with † if they are significantly better than all other approaches. Few-shot results (LLaMA Few-Shot) were obtained using 25 in-context examples with the *Context* prompt, while LLaMA-FT-ABSA refers to our fine-tuned model using fixed prompts and optimized hyperparameters. Other approaches are established SOTA baselines from the literature.

Task	Method	EN-Rest				DE-Rest			
		Full	1,000	500	50	Full	1,000	500	50
ACD	BERT-CLF	76.05	68.70	58.32	18.05	**92.29**†	**91.22**†	**90.13**†	19.43
	Hier-GCN	82.31	80.32	75.03	49.80	89.71	87.41	85.52	58.36
	LLaMA Few-Shot (25)	74.64	74.58	74.52	**74.49**	82.01	81.95	81.98	82.01
	LLaMA-FT-ABSA	**85.21**†	**81.46**	**81.15**	73.37	87.87	86.65	86.12	**82.24**
ACSA	BERT-CLF	51.24	42.11	43.20	8.03	83.17	**84.36**	81.69	3.49
	Hier-GCN	72.41	71.08	63.59	35.76	82.49	79.25	73.83	42.36
	LLaMA Few-Shot (25)	71.46	71.19	71.14	**71.56**†	78.56	78.53	78.67	**78.67**
	LLaMA-FT-ABSA	**80.72**	**80.18**†	**77.63**†	68.67	**84.40**	81.24	**81.38**	76.72
E2E	InstructABSA	75.63	74.95	**74.87**	43.97	71.50	69.96	67.94	34.18
	TAS-BERT	70.96	67.59	59.81	31.46	71.05	66.66	60.86	40.41
	LLaMA Few-Shot (25)	60.05	59.95	59.95	60.05	65.77	65.77	65.87	**65.76**
	LLaMA-FT-ABSA	**80.07**†	**77.98**	73.93	**63.71**	**80.58**†	**78.43**†	**73.85**†	63.34
TASD	MvP	70.12	68.14	64.63	43.11	70.06	65.94	63.62	42.09
	Paraphrase	71.84	68.38	63.76	38.57	70.39	66.58	62.81	34.85
	LLaMA Few-Shot (25)	45.45	45.41	45.51	45.62	49.30	49.59	49.91	49.72
	LLaMA-FT-ABSA	**75.65**†	**73.10**	**71.46**	**54.10**†	**75.61**†	**74.03**†	**72.86**†	**59.54**†

points (from 80.07 to 63.71). Similar drops are observed in the TASD task, particularly on DE-Rest, where performance falls by 16.1 points (from 75.61 to 59.54).

These results indicate that LLM fine-tuning is generally robust under moderate data scarcity, though applications in extremely low-resource settings should anticipate more noticeable performance losses.

Comparison with baseline methods in data-scarce conditions. Compared to our fine-tuned LLM, the baseline methods generally perform worse under data-scarce conditions. For example, TAS-BERT's F1 score on the E2E task drops by over 10 points when moving from full data to 500 examples. The classification-based baseline models, BERT-CLF and Hier-GCN, show a considerable drop in performance under class-rich conditions and low data availability, as seen in the ACD and ACSA tasks on EN-Rest, which contains 12 aspect categories, resulting in 12 classes for ACD and 36 classes (12 aspect categories x 3 polarities) for ACSA to predict. Although BERT-CLF outperforms our approach in a few isolated cases (e.g., in ACD and ACSA on DE-Rest), our approach consistently delivers the highest F1 scores across the majority of tasks and dataset sizes on both datasets.

Even under the extreme low-resource condition with only 50 samples, the fine-tuned LLM maintains strong and competitive performance across most tasks. While all methods experience noticeable performance drops when moving from 500- to 50-sample condition, the decline is least pronounced for our fine-tuned LLM. This is particularly evident in the more complex E2E and TASD tasks, where our approach continues to provide acceptable performance levels despite the severe data limitations. Notably, in the TASD task on DE-Rest, all differences between our LLM approach and the baseline methods are statistically significant. Furthermore, the baseline results we report are closely aligned with those from Gou et al. [25] in comparable low-resource ABSA settings, supporting the reliability and external validity of our experimental setup and findings.

Comparison with few-shot performance. Fine-tuning usually improves performance compared to few-shot ICL, particularly for more complex tasks like E2E and TASD, where the model must combine classification and term extraction. Across all moderate and resource-scarce dataset sizes (from full data to 500 examples), the fine-tuned models consistently achieve better performance than few-shot prompting by a clear margin.

Only under extreme low-resource conditions (50 samples) does few-shot prompting show isolated advantages. In specific tasks such as ACD and ACSA, it achieves performance close to or slightly above that of the fine-tuned model. However, this comes at the cost of longer input sequences, higher inference latency, and increased sensitivity to prompt design. These results highlight that while few-shot prompting may be useful when training is entirely infeasible, it does not match the robustness and overall performance of fine-tuning in most scenarios.

As discussed in Section 5.2, we observe that prompt formulation has minimal impact on fine-tuning outcomes. This, in turn, allows for more compact inputs during inference, making fine-tuned models significantly more efficient than few-shot prompting approaches, which require long prompts with many embedded examples. In contrast to ICL-based methods, fine-tuned models also deliver stable and competitive performance across all data sizes, even under limited training conditions. While our results confirm previous findings that few-shot prompting can serve as a useful fallback in training-constrained scenarios [43,69], fine-tuning remains the more cost-efficient and robust choice, particularly for high-throughput or production applications [93]. Moreover, few-shot prompting typically relies on general-purpose instruction-tuned models that may not generalize reliably across domains or languages, an important consideration in extreme low-resource scenarios, where domain shifts and language constraints are common.

Furthermore, recent studies have shown that demonstration retrieval methods, where in-context examples are selected dynamically based on similarity to the input, can further enhance ICL performance, particularly in low-resource and cross-domain settings [94–96]. For instance, Zheng et al. [97] and Wang et al. [98] present first retrieval-based approaches tailored for ABSA that rank or select in-context examples based on generation likelihood or multiple linguistic perspectives. While these studies show promising improvements in few-shot settings, they typically rely on access to representative and sufficiently large sets of examles for selection, an assumption that may not always hold in truly low-resource scenarios.

Summary: LLM advantages in handling limited data. Our results demonstrate that fine-tuned LLMs are well suited to low-resource scenarios, particularly when task complexity increases. In fact, the performance remains relatively stable, even with severely limited data and for tasks like TASD and E2E, highlighting the adaptability of LLMs in extracting ABSA structures under suboptimal conditions. In several cases, the fine-

tuned model achieves F1 scores in the 500-sample setting that match or exceed those of baseline models trained on the full dataset, emphasizing the effectiveness of fine-tuning for resource-poor domains. This suggests that our approach can achieve comparable or even superior results with substantially fewer training examples, thereby reducing the need for extensive and resource-intense annotation efforts, since, given the fine-grained and complex nature of ABSA, such annotations are often time-consuming, costly, and require considerable task- and domain-specific expertise [99,100].

That said, fine-tuned LLMs are not universally superior. For instance, the BERT-based BERT-CLF baseline achieves higher performance than our LLM on the ACD task on DE-Rest, a setting with relatively low task complexity. This suggests that while LLMs excel in more demanding tasks like E2E and TASD, their advantage over simpler classification baselines may diminish in lower-complexity scenarios.

5.4. Ablation study: generalizability across models, languages, and domains

To build upon our previous findings and further investigate the robustness and generalizability of our fine-tuned LLM approach under extreme low-resource conditions (50-samples setting), we conduct an extensive ablation study across multiple dimensions. We expand our evaluation beyond LLaMA 3 8B to include two additional LLMs, namely Mistral 7B v0.3 [101] and Gemma 3 4B [102]. This extension allows us to investigate whether our previously observed advantages are specific to the LLaMA architecture or whether they are more generally applicable to instruction fine-tuned LLMs.

Furthermore, to assess language generalizability, we extend our initial evaluations from German and English in the restaurant review domain to four additional languages covered by the SemEval 2016 datasets: Russian, Dutch, Spanish, and French [29]. This multilingual evaluation ensures that the observed trends are not language-dependent phenomena restricted to specific linguistic properties of German or English.

Additionally, we address domain generalizability by supplementing the established restaurant domain datasets with two German-language domains: user reviews focused on inclusion in public transport (MobASA [103]) and hotel reviews [104]. While these datasets are fewer in number due to limited availability of annotated ABSA datasets, they nonetheless provide important insights into how robustly the fine-tuned LLM approach generalizes beyond well-established domains.

Evaluation procedure and hyperparameters. Consistent with the analysis presented in Section 5.3, we again compare the performance of our fine-tuned LLMs with established SOTA baseline methods (as described in Section 4.6). Similarly to German and English, to ensure optimal comparability and representative baseline performance, we utilize specialized language-specific transformer models for each evaluated language, such as *rubert-base*[6] [105] for BERT-based methods and *ruT5-base*[7] [106] for T5-based approaches on the Russian-langauge dataset (see Appendix A.3 for a full list of models).

The general evaluation procedure remains consistent with the procedure outlined in Section 4.4. Therefore, for each condition (combination of language, domain, model, and method), we trained on five distinct training and test splits, reporting the average F1-micro score across these splits. Statistical significance between our fine-tuned LLM approach and baseline models was computed based on these averages with each langauge and domain treated as an individual sample.

As in our initial experiments (Sections 4.4 and 4.6), we fixed the hyperparameters across all configurations in the ablation study due to the lack of validation data in the 50-sample setting.

[6] https://huggingface.co/DeepPavlov/rubert-base-cased

[7] https://huggingface.co/ai-forever/ruT5-base

General performance across conditions. The results of our extended ablation study (see Table 6) demonstrate that the core trends observed in our initial 50-sample experiments on German and English data generalize well across additional languages and domains. While the main part of this work already highlighted the benefits of fine-tuned LLMs under extreme low-resource conditions, the ablation study confirms that these observations are not limited to the restaurant domain or to specific languages. Instead, similar performance patterns emerge consistently across a broader multilingual and multi-domain evaluation. To support visual interpretation, Table 6 highlights settings (in blue) where fine-tuned LLMs outperform all baselines, including prompting. Conversely, settings where a baseline outperforms all fine-tuned LLMs are highlighted in orange, indicating task-specific strengths of prompting in simpler classification setups.

In particular, across both newly introduced German-language domains, inclusion in public transport and hotel reviews, fine-tuned LLMs often achieve the best performance across all ABSA subtasks. This finding highlights their robustness not only beyond the commonly studied restaurant domain but also across less explored and structurally different domains. Even though absolute performance differs greatly according to the applied language, relative performance and ranking between methods usually remains stable.

Strengths of fine-tuned LLMs in complex tasks. Generally, instruction-fine-tuned LLMs (LLaMA 3 8B, Mistral 7B, Gemma 3 4B) continue to outperform baseline models in the more complex ABSA tasks such as E2E and TASD. For these tasks, at least two of the three LLMs consistently outperform all baseline approaches across nearly all domains and languages. Despite using the conservative Bonferroni-Holm correction in our significance testing, we identify specific LLMs that achieve statistically significant improvements over all baselines for both E2E and TASD (marked with † in Table 6 in the “Model Average”-column). These findings are in line with previous studies, which have shown that fine-tuning LLMs often yields better performance than prompting for structured or multi-step tasks, as it allows the model to internalize task-specific reasoning patterns and output formats [107,108].

Limitations of baselines and prompting trade-offs. By contrast, most existing SOTA baseline methods struggle to produce competitive results in this severely data-scarce scenario. Similar to the findings by Zhang et al. [109], classifier-based approaches such as BERT-CLF, Hier-GCN and TAS-BERT exhibit particularly poor performance in downstream tasks such as ACD, ACSA and E2E due to their reliance on sufficient training examples for each label combination, a condition not always met under low-resource constraints. This stands in stark contrast to the strong performance reported in Section 5.3 for resource-rich environments, especially for simpler classification tasks like ACD and ACSA, where these models were often able to keep up with or even surpass fine-tuned LLMs.

For these simpler tasks, only few-shot prompting with LLaMA yields competitive or even superior results in low-resource scenarios, echoing insights gained from analyzing performance on German and English restaurant reviews in Section 5.3. This also aligns with prior literature suggesting that large general-instruction pre-trained models can perform reasonably well in classification settings via ICL, even when training is not possible [110,111]. In fact, prompting often outperforms fully fine-tuned LLMs on ACD and ACSA, highlighting the task-specific trade-offs between fine-tuning and prompting under extreme data limitations.

In summary, our results offer a conclusive picture: When operating under extreme low-resource constraints, ICL proves most effective in simpler classification tasks (ACD, ACSA), while instruction fine-tuned LLMs excel in more complex setups (E2E, TASD), where the joint application of term extraction and aspect/sentiment classification adds an additional layer of difficulty that few-shot prompting and baseline classifiers struggle to solve.

Table 6
F1-micro scores for all ABSA tasks (ACD, ACSA, E2E, TASD) under extreme low-resource conditions (50 samples). Abbr.: FT = Fine-Tuned, FS (25) = Few-Shot with 25 examples. The table compares instruction-tuned LLMs, prompting, and SOTA baselines across eight domain-language pairs. Best scores per dataset are bolded; blue highlights settings where the best fine-tuned LLM outperforms all baselines; orange highlights settings where a baseline surpasses all fine-tuned models; † indicates significant improvements ($p_{adj} \leq 0.05$).

Aspect Category Detection (ACD)									
Method	Inclusion	Hotel	Restaurant						Model
	de	de	de	en	es	fr	nl	ru	Average
LLaMA 3 8B FT	**84.10**	**80.01**	**82.68**	71.42	68.84	66.88	65.16	57.93	72.13
Gemma 3 4B FT	76.11	75.04	81.55	71.76	69.53	63.36	69.16	61.06	70.95
Mistral 7B FT	83.00	75.70	76.59	66.08	69.58	65.67	67.68	69.27	71.70
LLaMA FS (25)	76.14	79.00	81.98	**74.49**	**73.66**	**68.42**	**73.91**	**72.06**	**74.96**
BERT-CLF	50.49	64.32	19.43	18.05	55.42	43.00	39.02	42.12	41.48
Hier-GCN	66.57	67.47	58.36	49.80	58.04	41.53	37.18	46.32	53.16

Aspect Category Sentiment Analysis (ACSA)									
Method	Inclusion	Hotel	Restaurant						Model
	de	de	de	en	es	fr	nl	ru	Average
LLaMA 3 8B FT	**72.08**	66.12	76.13	66.60	64.94	54.15	62.26	59.52	65.22
Gemma 3 4B FT	65.06	65.11	55.12	62.90	62.40	58.67	61.01	51.36	60.20
Mistral 7B FT	71.56	59.50	71.55	67.79	61.81	57.04	61.75	63.45	64.31
LLaMA FS (25)	67.28	**71.32**	**78.67**	**71.56**	**69.69**	**63.66**	**67.53**	**64.27**	**69.28**†
BERT-CLF	3.77	44.74	3.49	8.03	26.50	0.32	1.73	5.35	11.74
Hier-GCN	48.39	51.14	42.36	35.76	49.45	30.48	28.12	34.69	40.05

End-to-End ABSA (E2E)									
Method	Inclusion	Hotel	Restaurant						Model
	de	de	de	en	es	fr	nl	ru	Average
LLaMA 3 8B FT	**60.95**	–	66.69	65.99	**62.36**	58.53	50.98	50.20	59.39†
Gemma 3 4B FT	56.27	–	**67.20**	64.31	60.68	58.49	52.52	**56.31**	**59.40**
Mistral 7B FT	59.86	–	64.35	**66.10**	51.79	**58.84**	**53.82**	46.55	57.33
LLaMA FS (25)	57.78	–	65.76	60.05	60.04	54.29	49.83	48.95	56.67
InstructABSA	40.23	–	34.18	43.97	31.31	25.28	17.47	14.20	29.52
TAS-BERT	36.16	–	40.41	31.46	43.34	23.68	15.50	31.64	31.74

Target Aspect Sentiment Detection (TASD)									
Method	Inclusion	Hotel	Restaurant						Model
	de	de	de	en	es	fr	nl	ru	Average
LLaMA 3 8B FT	57.81	–	**58.83**	54.59	**53.29**	**48.91**	**48.24**	37.46	51.30†
Gemma 3 4B FT	43.09	–	55.73	49.71	46.42	35.14	39.87	42.83	44.68
Mistral 7B FT	**62.21**	–	57.94	**57.14**	48.56	48.11	44.58	**45.69**	**52.03**†
LLaMA FS (25)	50.07	–	49.72	45.62	46.00	37.40	38.48	40.00	43.90
MvP	52.16	–	42.09	43.11	30.50	29.51	34.37	30.84	37.51
Paraphrase	47.81	–	34.85	38.57	28.24	20.17	27.58	16.33	30.51

5.5. Comparative evaluation

We compare the results of our approach with those of current SOTA methods under identical conditions, including the same datasets, training/test splits, and ABSA tasks. For our approach, we report the performance values for all available prompt styles, as we did not observe statistically significant performance differences between prompt styles during cross-evaluation.

For evaluations on the EN-Rest dataset we use the following approaches: for the ACD task, we consider the graph convolutional network ECAN [112], for the ACSA task the T5-based approach LEGO-ABSA [88]; and for the ACD and ACSA tasks, the BERT-graph network mixture model Hier-GCN-Bert [87].

For the E2E task, to reflect the diversity of evaluation practices in prior work, we evaluate both on the full task (including explicit and implicit targets) and on an explicit-only subset. Specifically, InstructABSA [13] and TAS-BERT [72] are compared on the full E2E task, while GRACE [113] and DTW-GCN [114] serve as baselines for explicit-only extraction. This dual evaluation ensures fairness and compatibility with model-specific capabilities. Notably, many prior works employ modified or filtered versions of the EN-Rest dataset tailored to their specific task scope. To avoid introducing confounding variables from such preprocessing differences, we limit direct comparisons to models evaluated on subsets compatible with our configuration. Results for the explicit-only variant are reported in Table 7 within brackets. For the TASD task, we evaluate against four T5-based text generation approaches: MvP [25], LEGO-ABSA, TAS-BERT [72], and Paraphrase [70]. Additionally, we consider the results from Šmíd et al. [20], which fine-tuned Orca 2 (7B/13B) and LLaMA 2 (7B/13B).

Since the DE-Rest dataset has not yet been extensively studied using SOTA approaches, we use the baseline methods from Hellwig et al. [30] as reference values. In their work, the ACD and ACSA tasks were treated as multi-label text classification problems and addressed using a pre-trained BERT model (BERT-CLF), based on the approach by Fehle et al. [104]. For the E2E task, they employed E2E-ABSA, a BERT-based token classification approach for explicit aspects based on Li et al. [115].

Table 7
F1-micro values achieved on both datasets. Best results are in bold. For the E2E task, values in parentheses are results for explicit opinion target phrases only. For the EN-Rest (Rest-16) dataset, values with "*" are taken from Cui et al. [112] and for the DE-Rest (GERestaurant), values with "*" are taken from Hellwig et al. [30]. Few-shot results were obtained using 25 examples, which proved to be the best-performing configuration for each task during validation.

EN-Rest (Rest-16)				
Method	ACD	ACSA	E2E	TASD
LEGO-ABSA	–	76.20	–	71.80
ECAN	**88.75**	–	–	–
MvP	–	–	–	72.76
Hier-GCN-BERT	86.54*	74.55	–	–
InstructABSA	–	–	74.24	–
TAS-BERT	–	–	72.92 (75.68)	65.89
GRACE	–	–	(76.49)	–
DTW-GCN	–	–	(79.03)	–
Paraphrase	–	–	–	71.97
FT-Orca 2 7B	–	–	–	76.10
FT-Orca 2 13B	–	–	–	**78.82**
FT-LLaMA 2 7B	–	–	–	71.39
FT-LLaMA 2 13B	–	–	–	74.08
LLaMA Few-Shot (25)	71.89	67.89	65.35 (76.04)	45.88
LLaMA-FT-ABSA-Basic	83.33	78.00	80.49 (80.48)	72.50
LLaMA-FT-ABSA-Context	81.09	81.61	**81.77** (77.18)	76.72
LLaMA-FT-ABSA-CoT	–	**82.48**	80.49 **(80.70)**	70.20
DE-Rest (GERestaurant)				
Method	ACD	ACSA	E2E	TASD
BERT-CLF	**91.82***	85.14*	–	–
E2E-ABSA	–	–	**(81.61)***	–
Paraphrase	–	–	–	68.86*
LLaMA Few–Shot (25)	83.76	79.52	63.49 (73.35)	57.10
LLaMA-FT-ABSA-Basic	88.43	**85.45**	75.44 (78.88)	**75.13**
LLaMA-FT-ABSA-Context	87.67	84.70	**77.44** (75.35)	72.97
LLaMA-FT-ABSA-CoT	–	76.24	66.89 (78.22)	73.53

Additionally, for the TASD task, they used an implementation of the Paraphrase approach adapted for German.

In addition to that, for both datasets, we include results from ICL with Few-Shot Prompting with LLaMA-3-8B using the *Context* prompt and 25 few-shots, which achieved the best performance during cross evaluation.

The results for the four ABSA tasks on both datasets are presented in Table 7. For the EN-Rest (Rest-16) dataset, our LLM-based approach shows a noticeable performance gap for the ACD task, with a F1 score trailing the best-performing model (ECAN) by up to 5.4 points. However, in the ACSA task, our approach establishes a new SOTA with a F1 score of 82.48 using the *CoT*-style prompt, outperforming all previous methods. The E2E task further highlights the strengths of our approach, especially in scenarios requiring the detection of both explicit and implicit opinion targets. Using the *Context* prompt, our method surpasses the previous SOTA, InstructABSA, by approximately 8.5 points in F1 score. Similarly, for approaches focusing solely on explicit opinion targets, our approach outperforms the current SOTA DTW-GCN, improving the F1-micro score by 1.7 points to achieve a result of 80.70.

In the TASD task, our LLM-based approach achieves a F1 score of 76.72 using the *Context* prompt, surpassing previous SOTA methods such as MvP or Paraphrase. However, it ranks second to the fine-tuned Orca 2 13B model from Šmíd et al. [20], which has nearly double the parameter size. Notably, our approach, which is based on LLaMA 3 8B, achieves higher F1 scores than other LLM-based models of comparable size, such as Orca 2 7B and LLaMA 2 7B. Given the significant performance boost observed by Šmíd et al. [20] when increasing the parameter size of LLMs, we hypothesize that employing an equally large or larger LLM for our approach could result in similar performance gains, potentially aligning our results with those achieved by Orca 2 13B.

For the DE-Rest (GERestaurant) dataset our approach produces slightly worse F1 scores for the ACD task and the E2E task with focus on explicit opinion targets than the baselines implemented by Hellwig et al. [30]. However, for the ACSA and TASD tasks we can surpass the baselines, achieving a marginal improvement for ACSA and a considerable increase of nearly 6 points for TASD, with F1 scores of 85.45 and 75.13, respectively.

5.6. Resource requirements and efficiency analysis

To assess the resource demands of our approach, we compare training time, inference latency, and memory usage across all subtasks and both datasets, based on the cross-evaluation phase using the full dataset setting. All experiments were carried out on identical hardware (Nvidia RTX A5000 with 24GB VRAM) to ensure fair comparability. For consistency, all methods were evaluated under the same conditions. The results of our LLaMA-based approach are reported using a fixed configuration (learning rate: 3e-4, LoRA rank/α 8, 10 epochs), which, on average, achieved the best results during our evaluations. Table 8 summarizes the results. Runtimes reflect only actual training and inference durations, excluding overheads such as model initialization or preprocessing. As expected, run-timee and memory usage scale roughly proportionally with dataset size.

Training runtime and memory usage. Fine-tuning our LLaMA-based model QLoRA incurs moderate training costs. Depending on task complexity and prompt length, training times range from 22 minutes (ACD on DE-Rest) to 36 minutes (TASD on both datasets), with consistent memory usage of around 7.6 GB. In contrast, traditional baselines like BERT-CLF and Hier-GCN require only seconds to minutes but deliver mixed performance, sometimes superior (e.g., ACD on DE-Rest), yet often weaker, especially under low-resource or class-rich conditions. Other baselines such as TAS-BERT and MvP demand substantially more resources (e.g., TAS-BERT: >3h on DE-Rest, 6.5h on EN-Rest; MvP: ≥1h), with memory usage peaking at 18 GB for Paraphrase. These higher costs are not offset by superior performance, particularly in data-scarce settings. MvP's runtime is further increased by its data augmentation strategy, which inflates the dataset fivefold but fails to deliver with significantly superior performance.

Our results show that the advantages of larger, instruction-tuned LLMs, such as improved generalization and robustness, can be leveraged without suffering from excessive resource requirements, thanks to efficient fine-tuning techniques like QLoRA.

Inference efficiency. Few-shot prompting with 25-shot input leads to noticeably longer inference times due to increased prompt length, e.g., 13.9 s vs. 7.1 s (TASD on DE-Rest) and 12.8 s vs. 6.1 s (TASD on EN-Rest). This aligns with prior findings by Zhou et al. [93], which highlight the impact of input length on inference time. In contrast, our fine-tuned models use short prompts, enabling significantly faster inference while maintaining strong performance, making them well suited for real-time and high-throughput applications, such as enterprise use cases. Traditional baselines like BERT-CLF and Hier-GCN are faster at inference due to their simpler architectures, but lack the robustness and task flexibility of LLMs, especially for more complex situations.

Both fine-tuned and few-shot models were executed using the vLLM framework [62], which supports high-throughput decoding via continuous batching and other inference optimizations. For the remaining baselines (e.g., BERT- or T5-based models), we used the original implementations provided by their authors. While these do not inherently benefit from inference-optimized backends, similar frameworks for encoder-

Table 8
Resource usage across models and tasks (training time, GPU memory, inference time) on DE-Rest and EN-Rest datasets. All values are based on the full-dataset setting used in the cross-evaluation phase. Training times and memory utilization reflect average and peak values across five runs. For LLaMA-FT-ABSA, only training memory is reported; inference memory utilization is managed via model parameters in vLLM. In our experiments, `gpu_memory_utilization` was set to 0.8, corresponding to approximately 20 GB reserved VRAM.

Aspect Category Detection (ACD)						
	DE-Rest			EN-Rest		
Model	Train	Memory	Inference	Train	Memory	Inference
BERT-CLF	33.2 s	6.7 GB	0.5 s	25. 2s	6.9 GB	0.5 s
LLaMA-FT-ABSA	22m 10.3 s	7.6 GB	4.0 s	25 m 42.7 s	7.6 GB	3.7 s
LLaMA Few-Shot	–	–	11.4 s	–	–	3.7 s
Aspect Category Sentiment Analysis (ACSA)						
	DE-Rest			EN-Rest		
Model	Train	Memory	Inference	Train	Memory	Inference
BERT-CLF	33.6 s	6.7 GB	0.5s	25.2 s	6.9 GB	0.5 s
HIER-GCN	7 m 50.8 s	3.2 GB	1.3 s	7 m 56.1 s	3.4 GB	1.2 s
LLaMA-FT-ABSA	25 m 30.1 s	7.6 GB	5.1 s	28 m 23.0 s	7.6 GB	4.6 s
LLaMA Few-Shot	–	–	5.2 s	–	–	10.0 s
End-to-End ABSA (E2E)						
	DE-Rest			EN-Rest		
Model	Train	Memory	Inference	Train	Memory	Inference
TAS-BERT	202 m 31.0 s	5.6 GB	35.5 s	388 m 41.0 s	5.6 GB	66.9 s
InstructABSA	2 m 39.8 s	9.9 GB	6.6 s	2 m 17.0 s	8.7 GB	6.3 s
LLaMA-FT-ABSA	32 m 56.0 s	7.6 GB	5.8 s	25m 11.0 s	7.6 GB	4.1 s
LLaMA Few-Shot	–	–	5.4 s	–	–	9.5 s
Target Aspect Sentiment Detection (TASD)						
	DE-Rest			EN-Rest		
Model	Train	Memory	Inference	Train	Memory	Inference
Paraphrase	20 m 44.7 s	18.4 GB	30.4 s	16 m 24.4 s	18.1 GB	25.1 s
MvP	72 m 21.0 s	13.5 GB	331.7 s	57 m 48.0 s	13.5 GB	281.5s
LLaMA-FT-ABSA	36 m 13.0 s	7.6 GB	7.1 s	36 m 43.5 s	7.6 GB	6.1 s
LLaMA Few-Shot	–	–	13.9 s	–	–	12.8 s

based models, such as ONNX Runtime[8] or TensorRT,[9] could further reduce latency and may improve their applicability in real-time scenarios.

6. Conclusion

This study provides insights into the application of fine-tuned open source LLMs for ABSA across two datasets and languages. We showed that different prompt styles significantly influenced optimal hyperparameters, though minimal variations in performance were observed across prompts, suggesting concise prompts are sufficient for LLM fine-tuning. Fine-tuned LLMs demonstrate superior performance over baseline methods in data-scarce scenarios, particularly in more complex ABSA tasks like E2E and TASD, where they maintain relatively stable performance across varying dataset sizes and often achieve higher F1 scores than baselines trained on full datasets. Evaluations on EN-Rest and DE-Rest datasets revealed competitive performance across the ABSA tasks ACD, ACSA, E2E, and TASD, while achieving a new SOTA F1 score of 82.48 for ACSA and 81.77 for E2E (F1: 80.70 for explicit opinion targets only) on Rest-16 as well as 85.45 for ACSA and 75.13 for TASD on GERestaurant. Our comprehensive evaluation solidifies the relevance of fine-tuned LLMs for ABSA research and applications in both well-resourced and resource-scarce environments, showing that even a small fine-tuning dataset can enable SOTA-like results for ABSA, without requiring more computational resources than many traditional baselines.

These findings are further reinforced by our ablation study, which confirms that the advantages of fine-tuned LLMs generalize across different languages and domains, even under extreme low-resource constraints. In addition, our results show that, under such dataset limitations, prompting-based methods remain a competitive choice for simpler classification tasks (e.g., ACD, ACSA), whereas instruction fine-tuned LLMs consistently outperform alternatives in more complex setups such as E2E and TASD, highlighting a task-specific trade-off between ease of deployment and performance.

Overall, our findings highlight the strong potential of fine-tuned open source LLMs for ABSA and provide a solid basis for further research into more efficient, generalizable, and domain-adaptive approaches.

7. Limitations and future work

There are several limitations that may affect the results of our study. First, we only examined and fine-tuned one open source LLM (LLaMA 3 8B) on one dataset per language, which affects the generalizability of the results about the fine-tuning of open source models in general. We partially address this concern through our ablation study, which evaluates additional model architectures (Mistral 7B, Gemma 3 4B) as well as further domains and languages. These additional experiments confirm many of our core findings and increase confidence in the general robustness of fine-tuned LLMs under low-resource conditions. Similarly, we only examined the smallest model of the LLaMA-3 model family, whereas the possibility exists that our results and findings are not transferable to larger or different models. Additionally, locally hosted models also come with drawbacks which have to be considered, including the need for significant computational resources.

Another limitation concerns the choice of evaluation metrics. We primarily base our discussion on F1-micro, which is the predominant metric in ABSA research due to its sensitivity to frequent classes and widespread use in benchmark studies [29,116]. At the same time, we also report F1-macro scores in our repository, acknowledging that prior work has used this metric to account for class imbalance [117–119]. While F1-micro facilitates comparability with most existing studies, future work could complement our findings with a stronger focus on macro-level evaluation to capture more nuanced performance differences across classes.

Moreover, while our study disentangles prompt design and hyperparameter tuning to some extent, we did not systematically investigate their interactions, which may further influence fine-tuning outcomes. Our variance analysis suggests that prompt-hyperparameter interactions are relatively stable across settings, with *Basic* and *Context* prompts showing low variance and *CoT* prompts introducing more fluctuation. However, a deeper investigation is required to fully understand under which conditions prompt styles interact with hyperparameters in meaningful ways. Since all results and evaluation scripts are openly available in our repository, future work can directly build on our findings to replicate analyses of prompt-hyperparameter dynamics, extend them, or compare with results from similar studies.

Another promising direction lies in mechanistic or explainability analyses, which could provide deeper insights into why fine-tuned LLMs succeed in ABSA tasks and how they internalize task-specific knowledge. In addition, our evaluation is currently limited to established benchmark datasets; extending analyses to more diverse domains and noisy, real-world data would offer a stronger assessment of robustness. Finally, although we provide a basic discussion of computational requirements, a more systematic efficiency analysis is desirable. Future work should compare different training and inference frameworks, unify evaluation setups across methods, and more thoroughly investigate trade-offs between performance, cost, and accessibility.

Beyond these limitations, several avenues for future research arise directly from our findings. Prompt engineering, a rapidly advancing field,

[8] https://github.com/microsoft/onnxruntime
[9] https://github.com/NVIDIA/TensorRT

could be further explored for ABSA, particularly with techniques beyond few-shot ICL [64]. Automatic prompt generation methods using LLMs may help reduce manual effort and improve generalizability across tasks, domains, and languages [89,91]. Furthermore, recent retrieval-based methods for selecting in-context examples have shown promising results for ABSA and may offer complementary benefits in few-shot setups [97,98]. While we used base models for fine-tuning, future work could explore the benefits of instruction-tuned variants, particularly in extreme low-resource scenarios where instruction-following behavior or pre-training might offer an advantage. Furthermore, evaluating the performance of the fine-tuning model with as few as 5, 10, or 25 examples could provide deeper insights into the break-even point between ICL and fine-tuning. Lastly, exploring techniques for domain [120] or language transferability [121] of fine-tuned LLMs could further enhance their practical applicability for unexplored domains.

CRediT authorship contribution statement

Jakob Fehle: Conceptualization, Methodology, Project administration, Software, Writing – original draft, Writing – review & editing; **Udo Kruschwitz:** Methodology, Supervision, Writing – review & editing; **Nils Constantin Hellwig:** Writing – review & editing; **Christian Wolff:** Supervision, Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A.

A.1. Prompt examples for ACD, ACSA, E2E, and TASD for the DE-Rest dataset

See Fig. A.4, Fig. A.5, Fig. A.6, Fig. A.7, Fig. A.8, Fig. A.9, Fig. A.10, and Fig. A.11.

A.2. Prompt examples for TASD for the EN-Rest dataset

See Fig. A.12.

A.3. Model configurations for baseline approaches

For the LLaMA Few-Shot baseline, we use the instruction fine-tuned *LLaMA-3-8B-instruct* model. Prompting is applied to all four ABSA tasks (ACD, ACSA, E2E, and TASD) and evaluated using the *Context* prompt. Additionally, we explore different few-shot settings with 5, 10, and 25 examples. For each cross-validation split, a random subset of examples is selected once and consistently reused across all prompt variants and dataset sizes within that split. For each configuration, the optimal few-shot setting is determined based on the validation split. To ensure reproducibility, we use a temperature of 0 during inference.

For our fine-tuned baseline approaches, we adhere to the configurations specified in the respective papers for BERT-CLF [30], Hier-GCN [87], InstructABSA [24], TAS-BERT [72], MvP [25], and Paraphrase [30], ensuring consistency with previous work. Since these configurations were primarily defined for the English language, we replace the underlying pre-trained models with German-language versions where applicable for the DE-Rest (GERestaurant) dataset. The resulting specifications for each individual method are as follows:

For the ACD and ACSA tasks, BERT-CLF and Hier-GCN are implemented based on pre-trained models suitable for the language of the dataset: *bert-base*[10] and *bert-large*[11] for the English dataset and *gbert-base*[12] and *gbert-large*[13] for the German dataset. For the BERT-CLF approach on the DE-Rest dataset, we use a learning rate of 2×10^{-5}, a batch size of 16, and train the model for 3 epochs in the basic configuration. As it was shown that the hyperparameters for a multi-label classification are sensitive to a higher number of classes [104], we change the learning rate for the EN-Rest (Rest-16) dataset, which has a much larger number of classes than DE-Rest, and experiment with learning rates between 1×10^{-5} and 9×10^{-5}. The final learning rate is determined based on the validation split. For the Hier-GCN approach, we use a learning rate of 5×10^{-5}, a batch size of 8, and trained the model for 20 epochs.

For the E2E task, we use InstructABSA with the pre-trained *tk-instruct-base-def-pos*[14] model for the English dataset and *T5-base*[15] for the German dataset. *Tk-instruct-base-def-pos* is an instruction-tuned version of *T5-base*, which gets further adapted to the E2E task through InstructABSA. However, since *tk-instruct-base-def-pos* is specifically fine-tuned for English, we evaluated its non-fine-tuned base model, *T5-base*, on our DE-Rest validation split and observed better performance. As a result, *T5-base* was used for all subsequent evaluations with DE-Rest. Additionally, we employ TAS-BERT as a second baseline model, using *bert-base-uncased* for English and *gbert-base* for German, training both with a learning rate of 2×10^{-5}, a batch size of 24, over 30 epochs.

For the TASD task, both approaches, MvP and Paraphrase, are based on the multilingual model *T5*. For MvP, we use *T5-base* with a learning rate of 1×10^{-4}, a batch size of 16, and train the model for 20 epochs. Following the hyperparameter configurations provided by Gou et al. [25] for different dataset sizes, we adjust our settings for the subsets: for the 1,000-sample condition, we use a batch size of 8 and train for 30 epochs, for the 500-sample condition, we maintain the same batch size but increase the training to 50 epochs, while for the 50-sample condition we increase the training to 100 epochs. Furthermore, given the importance of capitalization in the German language, we avoid lowercasing text when working with the DE-Rest dataset. Following Hellwig et al. [30] for the Paraphrase approach, we use *T5-large*[16] and set the learning rate to 3×10^{-4}, use a batch size of 16, and train the model for 20 epochs. For Few-Shot Prompting, we set the context window to 8192 tokens and, consistent with our fine-tuning setup, use greedy decoding with a temperature of 0 to ensure deterministic and reproducible outputs. For each data split, the same few-shot examples are used consistently across all dataset size settings. The optimal combination of prompt style and few-shot configuration is selected based on validation results.

For the multilingual extension of our ablation study, we selected strong, language-specific transformer models for the Spanish, Dutch, Russian, and French datasets. For Spanish, we used `bert-base-spanish`[17] [122] and `T5-base-spanish` [123].[18] For Dutch, we employed `bert-base-dutch`[19] [124] and `T5-base-dutch`.[20] For Russian, we used `rubert-base`[21] [105] and `ruT5-base` [106].[22] For French, we relied on `bert-base-french`[23] and the `T5-base`[24] model, since French is one of its core pre-trained languages. These models serve as competitive language-specific baselines for BERT-based and T5-based baselines, respectively.

[10] https://huggingface.co/google-bert/bert-base-uncased
[11] https://huggingface.co/google-bert/bert-large-uncased
[12] https://huggingface.co/deepset/gbert-base
[13] https://huggingface.co/deepset/gbert-large
[14] https://huggingface.co/allenai/tk-instruct-base-def-pos
[15] https://huggingface.co/google-t5/t5-base
[16] https://huggingface.co/google-t5/t5-large
[17] https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased
[18] https://huggingface.co/vgaraujov/t5-base-spanish
[19] https://huggingface.co/GroNLP/bert-base-dutch-cased
[20] https://huggingface.co/yhavinga/t5-base-dutch
[21] https://huggingface.co/DeepPavlov/rubert-base-cased
[22] https://huggingface.co/ai-forever/ruT5-base
[23] https://huggingface.co/dbmdz/bert-base-french-europeana-cased
[24] https://huggingface.co/t5-base

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
German. Identify all aspects (FOOD, SERVICE, PRICE, AMBIENCE, GENERAL-IMPRESSION) that are addressed in the
sentence.

### Input:
Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.

### Output:
[FOOD, PRICE]
```

Fig. A.4. Prompt example for the *Basic* prompt of the ACD task for the DE-Rest (GERestaurant) dataset.

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
German. Extract all aspects that are addressed in the sentence and return them as a list. Return an empty list
if no aspects are addressed in the sentence. Return only the list, without any further comments or text.

* Consider the following aspects: [FOOD, SERVICE, PRICE, AMBIENCE, GENERAL-IMPRESSION].

FOOD refers to the food in general or specific dishes and drinks. SERVICE includes ratings on the service in
general, the attitude of the staff, waiting times or other services such as takeaway. PRICE relates to
opinions on the general pricing level or prices of food, drinks or other restaurant services. AMBIENCE refers
to the atmosphere inside and outside the restaurant, the facilities and the general noise level in the
restaurant. GENERAL-IMPRESSION includes opinions on the restaurant as a whole, without focus on the aspect
categories mentioned.

### Input:
Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.

### Output:
[FOOD, PRICE]
```

Fig. A.5. Prompt example for the *Context* prompt of the ACD task for the DE-Rest (GERestaurant) dataset.

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
German. Identify all aspects (FOOD, SERVICE, PRICE, AMBIENCE, GENERAL-IMPRESSION) that are addressed in the
sentence and analyze the sentiment expressed towards them (POSITIVE, NEUTRAL, NEGATIVE).

### Input:
Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.

### Output:
[(FOOD, POSITIVE), (PRICE, NEGATIVE)]
```

Fig. A.6. Prompt example for the *Basic* prompt of the ACSA task for the DE-Rest (GERestaurant) dataset.

A.4. Additional parameters used for LoRA fine-tuning

In addition to the previously discussed parameters, our LoRA fine-tuning configurations include integrating FlashAttention-2 [125] to accelerate attention computations. We use a LoRA dropout rate of 0.05 and a batch size of 8 due to computational constraints. For QLoRA, we utilize 4-bit NormalFloat (NF4) with double quantization and the bfloat16 floating-point format. We employ the AdamW optimizer [126] to fine-tune all attention and multi-layer perceptron (MLP) layers using LoRA. Furthermore, we utilize Neftune [127] which has shown to positively influence instruction-based fine-tuning by adding noise to the embedding vectors during training.

A.5. Dataset sizes for each step of the evaluation process

See Table A.9

A.6. Best-performing hyperparameter combinations identified during hyperparameter tuning

See Table A.10

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
German. Extract all (aspect category, sentiment polarity) tuples within a sentence by determining all aspects
and analyzing the sentiment expressed towards them. Return a list of tuples containing two strings in
parentheses. Return an empty list if no aspects are addressed in the sentence. Return only the list, without
further comments or text.

* Consider the following aspects: [FOOD, SERVICE, PRICE, AMBIENCE, GENERAL-IMPRESSION].
* Consider the following sentiment polarities: [POSITIVE, NEUTRAL, NEGATIVE].

FOOD refers to the food in general or specific dishes and drinks. SERVICE includes ratings on the service in
general, the attitude of the staff, waiting times or other services such as takeaway. PRICE relates to opinions
on the general pricing level or prices of food, drinks or other restaurant services. AMBIENCE refers to the
atmosphere inside and outside the restaurant, the facilities and the general noise level in the restaurant.
GENERAL-IMPRESSION includes opinions on the restaurant as a whole, without focus on the aspect categories
mentioned.
The labels POSITIVE, NEUTRAL and NEGATIVE describe the positive, neutral or negative sentiment expressed towards
the aspect.

### Input:
Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.

### Output:
[(FOOD, POSITIVE), (PRICE, NEGATIVE)]
```

Fig. A.7. Prompt example for the *Context* prompt of the ACSA task for the DE-Rest (GERestaurant) dataset.

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
German. Extract all (opinion target phrase, sentiment polarity) tuples of a sentence by identifying all
sentiments [POSITIVE, NEUTRAL, NEGATIVE] expressed in the text and determining the target phrase towards which
the sentiment expression is directed. Return a list of tuples, each containing two strings in parentheses. If an
opinion target is implied but not explicitly stated, identify its sentiment polarity and assign the opinion
target phrase "NULL".

### Input:
Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.

### Output:
[("Essen", POSITIVE), ("NULL", NEGATIVE)]
```

Fig. A.8. Prompt example for the *Basic* prompt of the E2E task for the DE-Rest (GERestaurant) dataset.

Table A.9
Dataset sizes for all steps of the evaluation process.

	Subset	HT Train-Size	HT Val-Size	CV Train-Size per run	CV Test Size per run	Original Train-Size	Original Test-Size
DE-Rest	Full	1795	359	1436	359	2154	924
	1000	833	167	667	359		
	500	416	84	333	359		
	50	–	–	40	359		
EN-Rest	Full	1423	285	1138	285	1708	587
	1000	833	167	667	285		
	500	416	84	333	285		
	50	–	–	40	285		

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
German. Extract all (opinion target phrase, sentiment polarity) tuples of a sentence by identifying all
sentiments expressed in the text and determining the target phrase towards which the sentiment expression is
directed. Return a list of tuples, each containing two strings in parentheses. If an opinion target is implied
but not explicitly stated, identify its sentiment polarity and assign the opinion target phrase "NULL". Return
an empty list if no sentiments are expressed in the sentence. Return only the list, without any further comments
or text.

* Consider the following sentiment polarities: [POSITIVE, NEUTRAL, NEGATIVE].

The labels POSITIVE, NEUTRAL and NEGATIVE describe the positive, neutral or negative sentiment expressed towards
the aspect.

### Input:
Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.

### Output:
[("Essen", POSITIVE), ("NULL", NEGATIVE)]
```

Fig. A.9. Prompt example for the *Context* prompt of the E2E task for the DE-Rest (GERestaurant) dataset.

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
German. Extract all (aspect category, sentiment polarity, aspect phrase) triples of a sentence by identifying
all the addressed aspect categories (FOOD, SERVICE, PRICE, AMBIENCE, GENERAL-IMPRESSION) and their
corresponding phrases and analyzing the sentiment (POSITIVE, NEUTRAL, NEGATIVE) expressed towards each aspect.
Return a list of triples, each containing three strings in parentheses. If an aspect is implied but not
explicitly stated, identify its aspect category and sentiment and assign the aspect phrase "NULL".

### Input:
Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.

### Output:
[(FOOD, POSITIVE, "Essen"), (PRICE, NEGATIVE, "NULL")]
```

Fig. A.10. Prompt example for the *Basic* prompt of the TASD task for the DE-Rest (GERestaurant) dataset.

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
German. Extract all (aspect category, sentiment polarity, aspect phrase) triples of a sentence by identifying
all the aspect categories addressed with their corresponding phrases and analyzing the sentiment expressed
towards each aspect. Return a list of triples, each containing three strings in parentheses. If an aspect is
implied but not explicitly stated, identify its aspect category and sentiment and assign the aspect phrase
"NULL". Return an empty list if no aspects are addressed in the sentence. Return only the list, without any
further comments or text.

* Consider the following aspects: [FOOD, SERVICE, PRICE, AMBIENCE, GENERAL-IMPRESSION].
* Consider the following sentiment polarities: [POSITIVE, NEUTRAL, NEGATIVE].

FOOD refers to the food in general or specific dishes and drinks. SERVICE includes ratings on the service in
general, the attitude of the staff, waiting times or other services such as takeaway. PRICE relates to opinions
on the general pricing level or prices of food, drinks or other restaurant services. AMBIENCE refers to the
atmosphere inside and outside the restaurant, the facilities and the general noise level in the restaurant.
GENERAL-IMPRESSION includes opinions on the restaurant as a whole, without focus on the aspect categories
mentioned.
The labels POSITIVE, NEUTRAL and NEGATIVE describe the positive, neutral or negative sentiment expressed towards
the aspect.

### Input:
Für das Essen 5 von 5 Sternen- war lecker und reichlich, Preislich etwas angehoben.

### Output:
[(FOOD, POSITIVE, "Essen"), (PRICE, NEGATIVE, "NULL")]
```

Fig. A.11. Prompt example for the *Context* prompt of the TASD task for the DE-Rest (GERestaurant) dataset.

```
### Instruction:
You are an advanced AI for text classification, specialized in aspect-based sentiment analysis for texts in
English. Extract all (aspect category, sentiment polarity, aspect phrase) triples of a sentence by identifying
all the aspect categories addressed with their corresponding phrases and analyzing the sentiment expressed
towards each aspect. Return a list of triples, each containing three strings in parentheses. If an aspect is
implied but not explicitly stated, identify its aspect category and its sentiment and assign the aspect phrase
"NULL". Return an empty list if no aspects are addressed in the sentence. Return only the list, without any
further comments or text.

* Consider the following aspects: [AMBIENCE#GENERAL, DRINKS#PRICES, DRINKS#QUALITY, DRINKS#STYLE_OPTIONS,
FOOD#PRICES, FOOD#QUALITY, FOOD#STYLE_OPTIONS, LOCATION#GENERAL, SERVICE#GENERAL, RESTAURANT#GENERAL,
RESTAURANT#PRICES, RESTAURANT#MISCELLANEOUS].
* Consider the following sentiment polarities: [POSITIVE, NEUTRAL, NEGATIVE].

AMBIENCE#GENERAL refers to the atmosphere inside and outside the restaurant, the facilities and the general
noise level in the restaurant. DRINKS#PRICES refers to the general pricing level of the drinks, DRINKS#QUALITY
refers to the quality of the drinks and DRINKS#STYLE_OPTIONS refers to the selection of drinks and the variety
of the drinks menu. FOOD#PRICES refers to the general pricing level of the food, FOOD#QUALITY refers to the
quality of the food and FOOD#STYLE_OPTIONS refers to the selection of food and the variety of the food menu.
LOCATION#GENERAL refers the the location of the restaurant. SERVICE#GENERAL includes ratings on the service in
general, the attitude of the staff, waiting times or other services such as takeaway. RESTAURANT#GENERAL refers
to general opinions about the restaurant, RESTAURANT#PRICES refers to the general pricing level of a restaurant
visit and RESTAURANT#MISCELLANEOUS includes miscellaneous opinions about the restaurant, without focus on the
aspect categories already mentioned.
The labels POSITIVE, NEUTRAL and NEGATIVE describe the positive, neutral or negative sentiment expressed towards
the aspect.

### Input:
I have eaten at Saul, many times, the food is always consistently, outrageously good.

### Output:
[(FOOD#QUALITY, POSITIVE, "food")]
```

Fig. A.12. Prompt example for the *Context* prompt of the TASD task for the EN-Rest (Rest-16) dataset.

Table A.10
Best-performing hyperparameter settings based on hyperparameter search, which are subsequently used for cross-evaluation. For the 50-examples setting, no hyperparameter search was performed, instead we relied on values from the literature as described in Section 4.4. Abbr.: LR = learning rate; r = LoRA rank; α = LoRA α #E = number of training epochs.

EN-Rest

Setting	Prompt	ACD				ACSA				E2E				TASD			
		LR	r	α	#E	LR	r	α	#E	LR	r	α	#E	LR	r	α	#E
Full	Basic	$3\epsilon^{-4}$	8	16	8	$3\epsilon^{-5}$	32	64	7	$3\epsilon^{-5}$	32	64	8	$3\epsilon^{-4}$	8	8	9
	Context	$3\epsilon^{-4}$	8	8	6	$3\epsilon^{-5}$	32	32	7	$3\epsilon^{-5}$	32	64	8	$3\epsilon^{-4}$	8	8	7
	CoT	–	–	–	–	$3\epsilon^{-5}$	32	64	5	$3\epsilon^{-4}$	8	8	8	$3\epsilon^{-4}$	8	8	7
1000	Basic	$3\epsilon^{-4}$	32	32	8	$3\epsilon^{-5}$	32	64	9	$3\epsilon^{-5}$	32	64	8	$3\epsilon^{-4}$	8	16	8
	Context	$3\epsilon^{-4}$	32	32	9	$3\epsilon^{-4}$	8	8	7	$3\epsilon^{-4}$	8	16	7	$3\epsilon^{-4}$	8	8	6
	CoT	–	–	–	–	$3\epsilon^{-4}$	8	8	10	$3\epsilon^{-4}$	8	16	9	$3\epsilon^{-4}$	8	16	8
500	Basic	$3\epsilon^{-4}$	8	16	10	$3\epsilon^{-5}$	32	64	5	$3\epsilon^{-4}$	8	16	2	$3\epsilon^{-4}$	8	8	7
	Context	$3\epsilon^{-4}$	8	16	6	$3\epsilon^{-4}$	8	8	6	$3\epsilon^{-4}$	8	16	2	$3\epsilon^{-4}$	32	32	4
	CoT	–	–	–	–	$3\epsilon^{-4}$	8	8	7	$3\epsilon^{-4}$	32	64	10	$3\epsilon^{-4}$	8	16	6
50	Basic	$3\epsilon^{-4}$	8	8	10	$3\epsilon^{-4}$	8	8	10	$3\epsilon^{-4}$	8	8	10	$3\epsilon^{-4}$	8	8	10

DE-Rest

Setting	Prompt	ACD				ACSA				E2E				TASD			
		LR	r	α	#E	LR	r	α	#E	LR	r	α	#E	LR	r	α	#E
Full	Basic	$3\epsilon^{-5}$	32	64	10	$3\epsilon^{-5}$	8	16	8	$3\epsilon^{-5}$	32	64	10	$3\epsilon^{-4}$	8	8	4
	Context	$3\epsilon^{-4}$	8	16	10	$3\epsilon^{-5}$	32	32	9	$3\epsilon^{-4}$	8	8	5	$3\epsilon^{-5}$	32	32	9
	CoT	–	–	–	–	$3\epsilon^{-4}$	8	16	7	$3\epsilon^{-4}$	8	16	7	$3\epsilon^{-4}$	8	8	7
1000	Basic	$3\epsilon^{-5}$	32	64	9	$3\epsilon^{-4}$	8	8	7	$3\epsilon^{-4}$	8	8	9	$3\epsilon^{-4}$	8	8	5
	Context	$3\epsilon^{-4}$	8	8	4	$3\epsilon^{-4}$	8	8	6	$3\epsilon^{-4}$	8	8	9	$3\epsilon^{-5}$	32	64	8
	CoT	–	–	–	–	$3\epsilon^{-5}$	32	32	9	$3\epsilon^{-4}$	8	16	9	$3\epsilon^{-5}$	32	64	7
500	Basic	$3\epsilon^{-5}$	32	32	7	$3\epsilon^{-5}$	8	8	8	$3\epsilon^{-4}$	32	64	4	$3\epsilon^{-4}$	32	32	3
	Context	$3\epsilon^{-5}$	32	32	2	$3\epsilon^{-4}$	8	8	5	$3\epsilon^{-4}$	32	64	7	$3\epsilon^{-4}$	8	8	8
	CoT	–	–	–	–	$3\epsilon^{-4}$	8	8	10	$3\epsilon^{-4}$	8	8	6	$3\epsilon^{-4}$	32	32	7
50	Basic	$3\epsilon^{-4}$	8	8	10	$3\epsilon^{-4}$	8	8	10	$3\epsilon^{-4}$	8	8	10	$3\epsilon^{-4}$	8	8	10

References

[1] M. Wankhade, A.C.S. Rao, C. Kulkarni, A survey on sentiment analysis methods, applications, and challenges, Artif. Intell. Rev. 55 (7) (2022) 5731–5780.

[2] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, Cambridge University Press, Cambridge, England, 2020.

[3] H.H. Do, P.W.C. Prasad, A. Maag, A. Alsadoon, Deep learning for aspect-based sentiment analysis: a comparative review, Expert Syst. Appl. 118 (2019) 272–299.

[4] Y.-C. Chang, C.-H. Ku, D.-D.L. Nguyen, Predicting aspect-based sentiment using deep learning and information visualization: the impact of COVID-19 on the airline industry, Inf. Manag. 59 (2) (2022) 103587.

[5] H. Li, B.X.B. Yu, G. Li, H. Gao, Restaurant survival prediction using customer-generated content: an aspect-based sentiment analysis of online reviews, Tour. Manag. 96 (104707) (2023) 104707.

[6] M. Rodríguez-Ibánez, A. Casánez-Ventura, F. Castejón-Mateos, P.-M. Cuenca-Jiménez, A review on sentiment analysis from social media platforms, Expert Syst. Appl. 223 (119862) (2023) 119862.

[7] S.U.S. Chebolu, F. Dernoncourt, N. Lipka, T. Solorio, A review of datasets for aspect-based sentiment analysis, in: Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Stroudsburg, PA, USA, 2023.

[8] G. Li, H. Wang, Y. Ding, K. Zhou, X. Yan, Data augmentation for aspect-based sentiment analysis, Int. J. Mach. Learn. Cybern. 14 (1) (2023) 125–133.

[9] M.A. Hedderich, L. Lange, H. Adel, J. Strötgen, D. Klakow, A survey on recent approaches for natural language processing in low-resource scenarios, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Stroudsburg, PA, USA, 2021, pp. 2545–2568.

[10] OpenAI, ChatGPT, 2023, (2023). Available at: https://chatgpt.com/.

[11] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, D. Yang, Is ChatGPT a general-purpose natural language processing task solver?, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, abs/2302.06476, Association for Computational Linguistics, Stroudsburg, PA, USA, 2023.

[12] J. Kocoń, I. Cichecki, O. Kaszyca, M. Kochanek, D. Szydło, J. Baran, J. Bielaniewicz, M. Gruza, A. Janz, K. Kanclerz, A. Kocoń, B. Koptyra, W. Mieleszczenko-Kowszewicz, P. Miłkowski, M. Oleksy, M. Piasecki, L. Radlinski, K. Wojtasik, S. Woźniak, P. Kazienko, ChatGPT: jack of all trades, master of none, Int. J. Inf. Fusion 99 (101861) (2023) 101861.

[13] K. Scaria, H. Gupta, S. Goyal, S. Sawant, S. Mishra, C. Baral, InstructABSA: instruction learning for aspect based sentiment analysis, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), Association for Computational Linguistics, Stroudsburg, PA, USA, 2024.

[14] Q. Zhong, L. Ding, J. Liu, B. Du, D. Tao, Can ChatGPT understand too? a comparative study on ChatGPT and fine-tuned BERT (2023). arXiv:2302.10198

[15] OpenAI, OpenAI GPT-4 API, 2024, (2024). Available at: https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4.

[16] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, LLaMA: open and efficient foundation language models, 2023, (2023). arXiv:2302.13971

[17] A.Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D.S. Chaplot, D. de las Casas, E.B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L.R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T.L. Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, W.E. Sayed, Mixtral of experts, 2024, (2024). arXiv:2401.04088

[18] P.F. Simmering, P. Huoviala, Large language models for aspect-based sentiment analysis, 2023, (2023). arXiv:2310.18025

[19] X. Zhang, N. Talukdar, S. Vemulapalli, S. Ahn, J. Wang, H. Meng, S.M.B. Murtaza, D. Leshchiner, A.A. Dave, D.F. Joseph, M. Witteveen-Lane, D. Chesla, J. Zhou, B. Chen, Comparison of prompt engineering and fine-tuning strategies in large language models in the classification of clinical notes, AMIA Summits Transl. Sci. Proc. 2024 (2024) 478.

[20] J. Šmíd, P. Priban, P. Kral, LLaMa-based models for aspect-based sentiment analysis, in: O. De Clercq, V. Barriere, J. Barnes, R. Klinger, J. Sedoc, S. Tafreshi (Eds.), Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, Association for Computational Linguistics, Stroudsburg, PA, USA, 2024, pp. 63–70.

[21] C. Irugalbandara, A. Mahendra, R. Daynauth, T.K. Arachchige, J. Dantanarayana, K. Flautner, L. Tang, Y. Kang, J. Mars, Scaling down to scale up: a cost-benefit analysis of replacing openAI's LLM with open source SLMs in production, in: 2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), IEEE, 2024, pp. 280–291.

[22] H.W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S.S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E.H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q.V. Le, J. Wei, Scaling instruction-finetuned language models, J. Mach. Learn. Res. 25 (70) (2024) 1–53.

[23] Y. Wang, H. Ivison, P. Dasigi, J. Hessel, T. Khot, K.R. Chandu, D. Wadden, K. MacMillan, N.A. Smith, I. Beltagy, H. Hajishirzi, How far can camels go? exploring the state of instruction tuning on open resources, Adv. Neural Inf. Process. Syst. 36 (2023) 74764–74786.

[24] S. Varia, S. Wang, K. Halder, R. Vacareanu, M. Ballesteros, Y. Benajiba, N. Anna John, R. Anubhai, S. Muresan, D. Roth, Instruction tuning for few-shot aspect-Based sentiment analysis, in: J. Barnes, O. De Clercq, R. Klinger (Eds.), Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 19–27.

[25] Z. Gou, Q. Guo, Y. Yang, MvP: multi-view prompting improves aspect sentiment tuple prediction, Annual Meeting of the Association for Computational Linguistics (2023) 4380–4397.

[26] H. Xu, Y. Zhang, Q. Wang, R. Xu, DS^2-ABSA: dual-stream data synthesis with label refinement for few-shot aspect-based sentiment analysis, arXiv [cs.CL] (2024).

[27] E. Perez, D. Kiela, K. Cho, True few-shot learning with language models, Adv. Neural Inf. Process. Syst. 34 (2021) 11054–11070.

[28] Y. Lu, M. Bartolo, A. Moore, S. Riedel, P. Stenetorp, Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 8086–8098.

[29] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S.M. Jiménez-Zafra, G. Eryiğit, Semeval-2016 task 5 : aspect based sentiment analysis, in: ProWorkshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, 2016, pp. 19–30.

[30] N.C. Hellwig, J. Fehle, M. Bink, C. Wolff, GERestaurant: a german dataset of annotated restaurant reviews for aspect-based sentiment analysis, Proceedings of the 20th Conference on Natural Language Processing (KONVENS 2024), Association for Computational Linguistics, 2024, p. 123-133.

[31] W. Zhang, X. Li, Y. Deng, L. Bing, W. Lam, A survey on aspect-based sentiment analysis: tasks, methods, and challenges, IEEE Trans. Knowl. Data Eng. 35 (11) (2023) 11019–11038.

[32] A.I. Anthropic, The claude 3 model family: opus, sonnet, haiku, Claude-3 Model Card 1 (2024).

[33] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A.M. Dai, A. Hauth, K. Millican, et al., Gemini: a family of highly capable multimodal models, 2024, (2024) arXiv:2312.11805

[34] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, L. Zhao, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, B. Ge, Summary of chatGPT-related research and perspective towards the future of large language models, Meta-Radiol. 1 (2) (2023) 100017.

[35] A. Bahrini, M. Khamoshifar, H. Abbasimehr, R.J. Riggs, M. Esmaeili, R.M. Majdabadkohne, M. Pasehvar, ChatGPT: applications, opportunities, and threats, in: 2023 Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2023, pp. 274–279.

[36] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, Y. Tang, A brief overview of chatGPT: the history, status quo and potential future development, IEEE/CAA J. Autom. Sin. 10 (5) (2023) 1122–1136.

[37] S. Kukreja, T. Kumar, A. Purohit, A. Dasgupta, D. Guha, A literature survey on open source large language models, in: Proceedings of the 2024 7th International Conference on Computers in Management and Business, ACM, New York, NY, USA, 2024.

[38] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A survey of large language models, 2024, (2024). arXiv:2303.18223

[39] S. Ateia, U. Kruschwitz, Can open-source LLMs compete with commercial models? exploring the few-shot performance of current GPT models in biomedical tasks, in: CEUR Workshop Proceedings, 3740, 2024.

[40] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, L. Yan, J. Shen, T. Liu, J. Liu, D. Metzler, X. Wang, M. Bendersky, Large language models are effective text rankers with pairwise ranking prompting, in: K. Duh, H. Gomez, S. Bethard (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, Stroudsburg, PA, USA, 2024, pp. 1504–1518.

[41] X. Ma, L. Wang, N. Yang, F. Wei, J. Lin, Fine-tuning LLaMA for multi-stage text retrieval, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 1, ACM, New York, NY, USA, 2024, pp. 2421–2425.

[42] M.M. Amin, E. Cambria, B.W. Schuller, Will affective computing emerge from foundation models and general AI? a first evaluation on chatGPT, IEEE Intell. Syst. 38 (2) (2023) 15–23.

[43] W. Zhang, Y. Deng, B. Liu, S. Pan, L. Bing, Sentiment analysis in the era of large language models: a reality check, in: Findings of the Association for Computational Linguistics: NAACL 2024, 2024, pp. 3881–3906.

[44] Z. Wu, Y. Wu, X. Feng, J. Zou, F. Yin, Improve chinese aspect sentiment quadruplet prediction via instruction learning based on large generate models, Comput. Mater. Contin. 0 (0) (2024) 1–10.

[45] J. Huang, Y. Cui, J. Liu, M. Liu, Supervised and few-shot learning for aspect-based sentiment analysis of instruction prompt, Electronics 13 (10) (2024) 1924.

[46] X. Ding, J. Zhou, L. Dou, Q. Chen, Y. Wu, C. Chen, L. He, Boosting large language models with continual learning for aspect-based sentiment analysis, Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, 2024, p. 4367-4377.

[47] M. Ahmed, Q. Chen, Y. Wang, Y. Nafa, Z. Li, T. Duan, DNN-Driven gradual machine learning for aspect-term sentiment analysis, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Stroudsburg, PA, USA, 2021.

[48] M. Ahmed, W. Bo, P. Shengfeng, S. Jianlin, A. Luo, L. Yunfeng, BERT-ASC: auxiliary-sentence construction for implicit aspect learning in sentiment analysis, Expert Syst. Appl. 258 (125195) (2024) 125195.

[49] A. Magueresse, V. Carles, E. Heetderks, Low-resource languages: a review of past work and future challenges, 2020, (2020). arXiv:2006.07264

[50] Y. Aliyu, A. Sarlan, K.U. Danyaro, A.S.B. Rahman, M. Abdullahi, Sentiment analysis in low-resource settings: a comprehensive review of approaches, languages, and data sources, IEEE Access 12 (2024) 66883–66909.

[51] A. Khattak, M.Z. Asghar, A. Saeed, I.A. Hameed, S. Asif Hassan, S. Ahmad, A survey on sentiment analysis in Urdu: a resource-poor language, Egypt. Inform. J. 22 (1) (2021) 53–74.

[52] M.S. Akhtar, A. Ekbal, P. Bhattacharyya, Aspect based sentiment analysis in hindi: resource creation and evaluation, in: N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp. 2703–2709.

[53] S. Rani, M.W. Anwar, Resource creation and evaluation of aspect based sentiment analysis in Urdu, in: B. Shmueli, Y.J. Huang (Eds.), Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop, Association for Computational Linguistics, Stroudsburg, PA, USA, 2020, pp. 79–84.

[54] Y.R. Regatte, R.R.R. Gangula, R. Mamidi, Dataset creation and evaluation of aspect based sentiment analysis in telugu, a low resource language, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 5017–5024.

[55] Y. Bai, Z. Han, Y. Zhao, H. Gao, Z. Zhang, X. Wang, M. Hu, Is compound aspect-based sentiment analysis addressed by LLMs?, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Stroudsburg, PA, USA, 2024, pp. 7836–7861.

[56] C. Wu, B. Ma, Z. Zhang, N. Deng, Y. He, Y. Xue, Evaluating zero-shot multilingual aspect-based sentiment analysis with large language models (2024). arXiv:2412.12564

[57] M. Hu, Y. Wu, H. Gao, Y. Bai, S. Zhao, Improving aspect sentiment quad prediction via template-order data augmentation, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 2022, pp. 7889–7900.

[58] N.C. Hellwig, J. Fehle, C. Wolff, Exploring large language models for the generation of synthetic training samples for aspect-based sentiment analysis in low resource settings, Expert Syst. Appl. 261 (125514) (2025) 125514.

[59] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Proceedings of the 35th International Conference on Neural Information Processing Systems, 35, Curran Associates, Inc., 2022, pp. 24824–24837.

[60] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRa: efficient finetuning of quantized LLMs, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, 36, Curran Associates Inc., 2023, pp. 10088–10115.

[61] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: low-rank adaptation of large language models, 2022. https://openreview.net/pdf?id=nZeVKeeFYf9.

[62] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C.H. Yu, J. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: Proceedings of the 29th Symposium on Operating Systems Principles, SOSP '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 611–626.

[63] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., The llama 3 herd of models (2024). arXiv:2407.21783

[64] P. Sahoo, A.K. Singh, S. Saha, V. Jain, S. Mondal, A. Chadha, A systematic survey of prompt engineering in large language models: techniques and applications, 2024, (2024). arXiv:2402.07927

[65] X. Amatriain, Prompt design and engineering: introduction and advanced methods (2024). arXiv:2401.14423

[66] Q. Wang, K. Ding, B. Liang, M. Yang, R. Xu, Reducing spurious correlations in aspect-based sentiment analysis with explanation from large language models, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 2930–2941.

[67] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q.V. Le, E.H. Chi, Least-to-most prompting enables complex reasoning in large language models, in: The Eleventh International Conference on Learning Representations, 2023.

[68] Y.C. Hua, P. Denny, K. Taskova, J. Wicker, A systematic review of aspect-based sentiment analysis (ABSA): domains, methods, and trends, arXiv [cs.CL] (2023).

[69] C. Wu, B. Ma, Z. Zhang, N. Deng, Y. He, Y. Xue, Evaluating zero-shot multilingual aspect-based sentiment analysis with large language models, arXiv [cs.CL] (2024).

[70] W. Zhang, Y. Deng, X. Li, Y. Yuan, L. Bing, W. Lam, Aspect sentiment quad prediction as paraphrase generation, in: M.-F. Moens, X. Huang, L. Specia, S.W.-T. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 9209–9219.

[71] X. Xu, J.-D. Zhang, R. Xiao, L. Xiong, The limits of chatGPT in extracting aspect-category-opinion-sentiment quadruples: a comparative analysis, arXiv [cs.CL] (2023).

[72] H. Wan, Y. Yang, J. Du, Y. Liu, K. Qi, J.Z. Pan, Target-aspect-sentiment joint detection for aspect-based sentiment analysis, AAAI Conf. Artif. Intell. 34 (05) (2020) 9122–9129.

[73] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, H. Wu, Unified structure generation for universal information extraction, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Stroudsburg, PA, USA, 2022, pp. 5755–5772.

[74] W. Zhang, X. Li, Y. Deng, L. Bing, W. Lam, Towards generative aspect-Based sentiment analysis, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Association for Computational Linguistics, Online, 2021, pp. 504–510.

[75] D. Biderman, J.G. Ortiz, J. Portes, M. Paul, P. Greengard, C. Jennings, D. King, S. Havens, V. Chiley, J. Frankle, C. Blakeney, J.P. Cunningham, LoRA learns less and forgets less, arXiv [cs.LG] (2024).

[76] D. Kalajdzievski, A rank stabilization scaling factor for fine-tuning with LoRA (2023). arXiv:2312.03732

[77] R. Mukherjee, S. Shetty, S. Chattopadhyay, S. Maji, S. Datta, P. Goyal, Reproducibility, replicability and beyond: assessing production readiness of aspect based sentiment analysis in the wild, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2021, pp. 92–106.

[78] A. Belz, S. Agarwal, A. Shimorina, E. Reiter, A systematic review of reproducibility research in natural language processing, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, Stroudsburg, PA, USA, 2021, pp. 381–393.

[79] H. Yang, C. Zhang, K. Li, PyABSA: a modularized framework for reproducible aspect-based sentiment analysis, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, ACM, New York, NY, USA, 2023.

[80] N. Alex, E. Lifland, L. Tunstall, A. Thakur, P. Maham, C.J. Riedel, E. Hine, C. Ashurst, P. Sedille, A. Carlier, M. Noetel, A. Stuhlmüller, RAFT: a real-world few-shot text classification benchmark, 2022.

[81] B. Efron, Bootstrap methods: another look at the jackknife, in: S. Kotz, N.L. Johnson (Eds.), Breakthroughs in Statistics: Methodology and Distribution, Springer New York, New York, NY, 1992, pp. 569–593.

[82] W.H. Kruskal, W.A. Wallis, Use of ranks in one-criterion variance analysis, J. Am. Statist. Assoc. 47 (260) (1952) 583–621.

[83] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (2) (1979) 65–70.

[84] S.S. Shapiro, M.B. Wilk, An analysis of variance test for normality (complete samples), Biometrika 52 (3–4) (1965) 591–611.

[85] A. Field, J. Miles, Z. Field, Discovering Statistics Using R, SAGE, 2012.

[86] F. Wilcoxon, Individual comparisons by ranking methods, in: S. Kotz, N.L. Johnson (Eds.), Breakthroughs in Statistics: Methodology and Distribution, Springer New York, New York, NY, 1992, pp. 196–202.

[87] H. Cai, Y. Tu, X. Zhou, J. Yu, R. Xia, Aspect-category based sentiment analysis with hierarchical graph convolutional network, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 833–843.

[88] T. Gao, J. Fang, H. Liu, Z. Liu, C. Liu, P. Liu, Y. Bao, W. Yan, LEGO-ABSA: a prompt-based task assemblable unified generative framework for multi-task aspect-based sentiment analysis, in: N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T.K. Lee, E. Santus, F. Bond, S.-H. Na (Eds.), Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 7002–7012.

[89] T. Shin, Y. Razeghi, R.L. Logan, IV, E. Wallace, S. Singh, Autoprompt: eliciting knowledge from language models with automatically generated prompts, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Stroudsburg, PA, USA, 2020, pp. 4222–4235.

[90] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Stroudsburg, PA, USA, 2021, pp. 3816–3830.

[91] Y. Zhou, A.I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, J. Ba, Large language models are human-level prompt engineers, in: The Eleventh International Conference on Learning Representations, 2023.

[92] M. Deng, J. Wang, C.-P. Hsieh, Y. Wang, H. Guo, T. Shu, M. Song, E. Xing, Z. Hu, RLPrompt: optimizing discrete text prompts with reinforcement learning, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, 2022, pp. 3369–3391.

[93] Z. Zhou, X. Ning, K. Hong, T. Fu, J. Xu, S. Li, Y. Lou, L. Wang, Z. Yuan, X. Li, S. Yan, G. Dai, X.-P. Zhang, Y. Dong, Y. Wang, A survey on efficient inference for large language models, arXiv [cs.CL] (2024).
[94] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, W. Chen, What makes good in-context examples for GPT-3?, in: E. Agirre, M. Apidianaki, I. Vulić (Eds.), Proceedings of Deep Learning inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, Association for Computational Linguistics, Dublin, Ireland and Online, 2022, pp. 100–114.
[95] O. Rubin, J. Herzig, J. Berant, Learning to retrieve prompts for in-context learning, in: M. Carpuat, M.-C. de Marneffe, I.V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Stroudsburg, PA, USA, 2022, pp. 2655–2671.
[96] K. Peng, L. Ding, Y. Yuan, X. Liu, M. Zhang, Y. Ouyang, D. Tao, Revisiting demonstration selection strategies in-context learning, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Stroudsburg, PA, USA, 2024, pp. 9090–9101.
[97] G. Zheng, J. Wang, L.-C. Yu, X. Zhang, Instruction tuning with retrieval-based examples ranking for aspect-based sentiment analysis, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics ACL 2024, Association for Computational Linguistics, Stroudsburg, PA, USA, 2024, pp. 4777–4788.
[98] Q. Wang, H. Xu, K. Ding, B. Liang, R. Xu, In-context example retrieval from multi-perspectives for few-shot aspect-based sentiment analysis, LREC (2024) 8975–8985.
[99] M. El-Haj, U. Kruschwitz, C. Fox, Creating language resources for under-resourced languages: methodologies, and experiments with Arabic, Lang. Resour. Eval. 49 (2015) 549–580.
[100] M. Poesio, J. Chamberlain, U. Kruschwitz, L. Robaldo, L. Ducceschi, Phrase detectives: utilizing collective intelligence for internet-scale language resource creation, ACM Trans. Interac. Intell.Syst. (TiiS) 3 (1) (2013) 1–44.
[101] A.Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D.S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L.R. Lavaud, M.-A. Lachaux, P. Stock, T.L. Scao, T. Lavril, T. Wang, T. Lacroix, W.E. Sayed, Mistral 7B, 2023, (2023). arXiv:2310.06825
[102] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, et al., Gemma 3 Technical Report, 2025, (2025). arXiv:2503.19786
[103] A. Gabryszak, P. Thomas, MobASA: corpus for aspect-based sentiment analysis and social inclusion in the mobility domain, in: M. Wan, C.-R. Huang (Eds.), Proceedings of the First Computing Social Responsibility Workshop within the 13th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 35–39.
[104] J. Fehle, L. Münster, T. Schmidt, C. Wolff, Aspect-based sentiment analysis as a multi-label classification task on the domain of german hotel reviews, in: Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023), Association for Computational Lingustics, 2023, pp. 202–218.
[105] Y. Kuratov, M. Arkhipov, Adaptation of deep bidirectional multilingual transformers for Russian language, arXiv [cs.CL] (2019).
[106] D. Zmitrovich, A. Abramov, A. Kalmykov, M. Tikhonova, E. Taktasheva, D. Astafurov, M. Baushenko, A. Snegirev, T. Shavrina, S. Markov, V. Mikhailov, A. Fenogenova, A family of pretrained transformer language models for Russian, LREC (2023) 507–524.
[107] M. Mosbach, T. Pimentel, S. Ravfogel, D. Klakow, Y. Elazar, Few-shot fine-tuning vs. in-context learning: a fair comparison and evaluation, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Stroudsburg, PA, USA, 2023, pp. 12284–12314.
[108] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, C. Raffel, Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, Neural Inf. Process. Syst. abs/2205.05638 (2022) 1950–1965.
[109] T. Zhang, F. Wu, A. Katiyar, K.Q. Weinberger, Y. Artzi, Revisiting few-sample BERT fine-tuning, Int. Conf. Learn Represent. abs/2006.05987 (2020).
[110] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, Y. Wu, B. Neyshabur, G. Gur-Ari, V. Misra, Solving quantitative reasoning problems with language models, Neural Inf. Process. Syst. abs/2206.14858 (2022) 3843–3857.
[111] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, Others, Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.
[112] J. Cui, F. Fukumoto, X. Wang, Y. Suzuki, J. Li, N. Tomuro, W. Kong, Enhanced coherence-aware network with hierarchical disentanglement for aspect-category sentiment analysis, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, 2024, p. 5843-5855.
[113] H. Luo, L. Ji, T. Li, D. Jiang, N. Duan, GRACE: gradient harmonized and cascaded labeling for aspect-based sentiment analysis, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Stroudsburg, PA, USA, 2020.
[114] Y. Mu, S. Shi, Dependency-type weighted graph convolutional network on end-to-end aspect-based sentiment analysis, in: IFIP Advances in Information and Communication Technology, Springer Nature Switzerland, Cham, 2024, pp. 46–57.
[115] X. Li, L. Bing, W. Zhang, W. Lam, Exploiting BERT for end-to-end aspect-based sentiment analysis, in: Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), Association for Computational Linguistics, Stroudsburg, PA, USA, 2019.
[116] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, SemEval-2014 task 4: aspect based sentiment analysis, in: P. Nakov, T. Zesch (Eds.), Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics, Dublin, Ireland, 2014, pp. 27–35.
[117] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent attention network on memory for aspect sentiment analysis, in: M. Palmer, R. Hwa, S. Riedel (Eds.), Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2017, pp. 452–461.
[118] C. Zhang, Q. Li, D. Song, Aspect-based sentiment classification with aspect-specific graph convolutional networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, 2019, pp. 4568–4578.
[119] H. Yan, J. Dai, T. Ji, X. Qiu, Z. Zhang, A unified generative framework for aspect-based sentiment analysis, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 2416–2429.
[120] S. Yuan, M. Li, Y. Du, Y. Xie, Cross-domain aspect-based sentiment classification with hybrid prompt, Expert Syst. Appl. 255 (124680) (2024) 124680.
[121] P. Přibáň, J. Šmíd, J. Steinberger, A. Mištera, A comparative study of cross-lingual sentiment analysis, Expert Syst. Appl. 247 (123247) (2024) 123247.
[122] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained BERT model and evaluation data, arXiv [cs.CL] (2023).
[123] V. Araujo, M. Truşcă, R. Tufino, M. Moens, Sequence-to-sequence Spanish pre-trained language models, LREC abs/2309.11259 (2023) 14729–14743.
[124] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, M. Nissim, BERTje: a dutch BERT model, arXiv [cs.CL] (2019).
[125] T. Dao, Flashattention-2: faster attention with better parallelism and work partitioning, 2024.
[126] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. https://openreview.net/pdf?id=mZn2Xyh9Ec.
[127] N. Jain, P.-Y. Chiang, Y. Wen, J. Kirchenbauer, H.-M. Chu, G. Somepalli, B.R. Bartoldson, B. Kailkhura, A. Schwarzschild, A. Saha, M. Goldblum, J. Geiping, T. Goldstein, NEFTune: noisy embeddings improve instruction finetuning, 2024. https://openreview.net/pdf?id=0bMmZ3fkCk.