Can LLMs Predict Citation Intent? An Experimental Analysis of In-context Learning and Fine-tuning on Open LLMs

Paris Koloveas[1,2][0000−0003−2376−089X],
Serafeim Chatzopoulos[1][0000−0003−1714−5225],
Thanasis Vergoulis[1][0000−0003−0555−4128], and
Christos Tryfonopoulos[2][0000−0003−0640−9088]

[1] IMSI, Athena RC, Athens, Greece
{pkoloveas, schatz, vergoulis}@athenarc.gr
[2] University of the Peloponnese, Tripolis, Greece
{pkoloveas, trifon}@uop.gr

Abstract. This work investigates the ability of open Large Language Models (LLMs) to predict citation intent through in-context learning and fine-tuning. Unlike traditional approaches relying on domain-specific pre-trained models like SciBERT, we demonstrate that general-purpose LLMs can be adapted to this task with minimal task-specific data. We evaluate twelve model variations across five prominent open LLM families using zero-, one-, few-, and many-shot prompting. Our experimental study identifies the top-performing model and prompting parameters through extensive in-context learning experiments. We then demonstrate the significant impact of task-specific adaptation by fine-tuning this model, achieving a relative F1-score improvement of 8% on the SciCite dataset and 4.3% on the ACL-ARC dataset compared to the instruction-tuned baseline. These findings provide valuable insights for model selection and prompt engineering. Additionally, we make our end-to-end evaluation framework and models openly available for future use.

1 Introduction

Citations are references of research articles to external sources of information included to support claims, provide context, criticize, or acknowledge prior work. Although their primary function is to inform and redirect the reader, citations are also frequently used for other purposes, such as serving as proxies of scientific impact in various types of analysis [18]. In such cases, understanding the exact intent of a citation is crucial, as not all types of citations should be considered. For instance, while measuring the impact of an article, citations that criticize the work should generally not be considered as contributing to its impact.

Predicting citation intent based on its context (i.e., the sentences in the manuscript that accompany the citation) and other related information, has become an important classification problem to support the aforementioned use cases. Existing approaches have traditionally relied on linguistic features [17], machine learning methods [32], and, recently, on domain-specific pre-trained language models (PLMs), such as SciBERT [3], that require large scientific datasets such as [21] and task-specific architectures[3]. In contrast, this work is the first to explore the potential of open, general-purpose Large Language Models (LLMs) – which, despite potentially encountering scientific text during their broad pre-training, are not specifically optimized for the scientific domain like SciBERT – to accurately identify citation intent, evaluating their effectiveness through in-context learning and fine-tuning on minimal task-specific data.

Large Language Models are advanced natural language processing systems trained on extensive text corpora to perform a wide range of language tasks. Unlike traditional pre-trained language models, LLMs are general-purpose models capable of adapting to new tasks with minimal additional training. Their ability to process and generate coherent text across diverse contexts makes them particularly suitable for tasks like citation intent classification, where understanding nuanced language patterns is essential.

In-context learning is a paradigm that allows LLMs to learn tasks given only a few examples in the form of demonstration [4,8]. In-context learning is particularly suited to citation intent classification, as it allows the models to leverage their contextual understanding of language to make accurate predictions.

In our study, we conduct an extensive experimental analysis of 12 general-purpose Instruction-tuned LLMs from 5 model families, on two widely used datasets for this task. In this context, we analyze the impact of multiple parameters on model performance and identify the optimal configurations and best-performing models for our problem. These experiments address the following research questions:

- *RQ1:* How well can pre-trained LLMs perform on citation intent classification without task-specific training?
- *RQ2:* What are the differences in performance between open LLMs of varying parameter counts?
- *RQ3:* How do different prompting-related parameters affect model performance?

Furthermore, we take the analysis a step further by selecting the top-performing cases from the aforementioned experiments for Supervised Fine-Tuning (SFT) on our datasets, allowing us to evaluate their optimal performance. This essentially addresses an additional research question:

- *RQ4:* How much does supervised fine-tuning with task-specific training affect the performance of the instruction-tuned models?

[3] In this paper, when referring to traditional PLMs in this context, we primarily mean models based on the encoder-only Transformer architecture, such as BERT and its derivatives (e.g., SciBERT), which were commonly fine-tuned for specific classification tasks. This distinguishes them from the general-purpose Large Language Models (LLMs) evaluated in this work, which typically utilize decoder-only or encoder-decoder Transformer architectures and are often leveraged for their generative and in-context learning capabilities.

Our experiments aim to contribute to the broader research community by guiding prompt engineering and model selection strategies for similar tasks. To further assist researchers, we openly[4] provide our complete testing suite and evaluation results, allowing seamless integration of Hugging Face models and easy configuration adjustments. We also publish the weights and checkpoints of the fine-tuned models[5] presented in Section 4.

2 Methodology

In this section, we elaborate on the evaluation methodology followed in our experimental study. We first provide an overview of the used models (Section 2.1), datasets (Section 2.2), and configuration parameters (Section 2.3). Finally, in Section 2.4, we present the technical specifications of the system we used for the experimentation.

2.1 Models

For our experiments, we selected the Instruction-tuned versions of five prominent open-weight model families: *LLaMA* [11], *Mistral* [15], *Phi* [1], *Gemma* [10], and *Qwen* [2,26]. A key challenge was performing the experiments on commodity hardware, both due to computational limitations and to demonstrate that citation intent classification can be executed efficiently with limited resources. This influenced the selection of models, which range in size from small (1B parameters) to medium-sized (27B parameters).

Since in-context learning inherently increases the number of tokens to each prompt (particularly in the many-shot scenario), we opted for a lower cutoff of $8,192$ tokens in the context length. This ensures that all selected models could process the longest prompts in our experiments without truncation.

To reduce the memory footprint and computational requirements of our evaluation, we utilize the 8-bit (Q8) quantized versions of the models. This approach significantly reduces memory usage without compromising performance. *Quantization* involves converting model parameters from higher numerical precisions (e.g. 16-bit floating-point) to formats of lower numerical precision (e.g. 8-bit integers). This process enables more efficient computation and reduced memory usage while largely preserving model performance and expressive power [16].

The model variations used for our evaluation were the following:

- Llama 3 & 3.1 (8B), Llama 3.2 (1B, 3B)
- Mistral Nemo (12B)
- Phi 3 Medium (14B), Phi 3.5 Mini (3.8B)
- Gemma 2 (2B, 9B, 27B)
- Qwen 2 (7B), Qwen 2.5 (14B)

[4] https://github.com/athenarc/CitationIntentOpenLLM
[5] https://huggingface.co/collections/sknow-lab/citationintentllm-67b72f1d5ca6113f960dba04

2.2 Datasets

For our experiments, we used two datasets:

- *SciCite* [5] consists of 11,021 citation strings annotated with three classes: `Background Information`, `Method`, and `Results Comparison`.
- *ACL-ARC* [17] contains 1,941 citation strings split into six categories: `Background`, `Motivation`, `Uses`, `Extends`, `Compares or Contrasts`, and `Future`.

These two datasets are widely used in citation intent classification research. The SciCite dataset is larger and more diverse, with broad coverage across multiple scientific fields, namely, Computer Science, Medicine, Neuroscience, and Biochemistry, while the ACL-ARC dataset offers a more granular classification scheme, focused on Computational Linguistics. Accross our evaluation, we used the original splits of the datasets, which include training, validation, and test sets (75% - 8% - 17% for SciCite, and 87% - 7% - 6% for ACL-ARC).

2.3 Configuration Parameters

In this section, we describe the configuration parameters of the examined models, including in-context learning methods, system prompts, query templates, example methods, and temperature settings.

In-context Learning Methods (ILM). To evaluate model performance, we applied four prompting methods[6] by following the In-Context Learning paradigm [4]: *Zero-shot* (no examples), *One-shot* (a single example per class), *Few-shot* (5 examples per class), and *Many-shot* (10 examples per class).

These prompting methods were selected to evaluate whether model performance improves as the number of examples increases, and to identify any saturation point where additional examples yield diminishing or negative returns. This aligns with prior literature on in-context learning methods, where these specific configurations have been extensively studied [8]. Incorporating examples directly into the prompts allows models to better understand the task and the expected output format, which is especially important for citation intent classification, where the citation context is key to determining the correct label.

System Prompts (SP). System prompts (SP) are foundational to our experimental design, establishing the context and expected behavior for the models in the citation intent classification task. They provide task-specific instructions, class definitions, and output guidelines. We developed and evaluated three distinct system prompt variations, shown in Figure 1.

- **SP1:** This initial prompt offered a straightforward, intuitive instruction for the classification task.

[6] Throughout the paper, we use "In-context Learning Methods" and "Prompting Methods" interchangibly.

System Prompt 1 [SP1]

```
You are an expert researcher tasked with classifying the
intent of a citation in a scientific publication.

The {{num_class}} possible classes are the following:
{{classes}}.

The definitions of the classes are:

{{class_definitions}}

For each given sentence, you must analyse only the citation
with the @@CITATION@@ tag. You must assign only one
class to each citation. Only return the class name, with no
elaboration.
```

System Prompt 2 [SP2]

```
# CONTEXT #

You are an expert researcher tasked with classifying the
intent of a citation in a scientific publication.

########

# OBJECTIVE #
You will be given a sentence containing a citation, you must
output the appropriate class as an answer.

########

# CLASS DEFINITIONS #
The {{num_class}} possible classes are the following:
{{classes}}.

The definitions of the classes are:
{{class_definitions}}

########

# RESPONSE RULES #

You must strictly adhere to the following rules for your
response:
- For each sentence, you must analyse only the citation with
the @@CITATION@@ tag.
- You must assign only one class to each citation.
- Only respond with the class name, with no explanation or
elaboration.
- Only answer with one or two words.
- Always be very brief.
```

System Prompt 3 [SP3]

```
# CONTEXT #

You are an expert researcher tasked with classifying the
intent of a citation in a scientific publication.

########

# OBJECTIVE #
You will be given a sentence containing a citation. You must
classify the intent of the citation by assigning it to one
of {{num_class}} predefined classes.

########

# CLASS DEFINITIONS #
The {{num_class}} possible classes are the following:
{{classes}}.

The definitions of the classes are:
{{class_definitions}}

########

# RESPONSE RULES #
- Analyze only the citation marked with the @@CITATION@@
tag.
- Assign exactly one class to each citation.
- Respond only with the exact name of one of the following
classes: {{classess}}.
- Do not provide any explanation or elaboration.
```

Fig. 1. Our System Prompts.

- **SP2:** This prompt introduced more structure, drawing inspiration from the CO-STAR framework [31]. We adapted relevant CO-STAR components by definint the task *Context*, the *Objective* of the model, and detailed *Response Rules*. Given that our task requires a single, specific class label output, the stylistic components of CO-STAR (*Style*, *Tone*, and *Audience*) were not incorporated.
- **SP3:** Building upon SP2, this prompt further refined wording of the *Objective* and *Response Rules*. It also explicitly reiterated the expected labels in the *Response Rules* section, aiming to improve task clarity and consistency in the model's outputs.

Query Templates (QT). Query templates (QT) define the specific format used to present citation sentences to the model, whether as part of an in-context learning example or as the actual query requiring classification. We evaluated two template structures:

- **Simple Query:** Initially, we used a basic format where the citation sentence was followed by "Class:". For examples, the correct class was appended; for queries, this was left for the model to predict.
- **Multiple-choice Query:** We observed that some models struggled to adhere to the strict class-label-only output expected with the Simple Query, resulting in inconsistent performance. To mitigate this, we introduced a more

structured template. Here, after the citation sentence, the model is explicitly prompted to choose the most appropriate citation intent from a presented list of all possible class labels.

Although the Multiple-choice template increases the token count per instance, it yielded significant performance improvements (detailed in Section 3.2) by providing clearer guidance to the models.

Example Presentation Formats (EPF). In our experiments, we explored two formats for presenting in-context examples to the models: *Inline* and *Conversational*.

- **Inline:** Example citation sentences and their corresponding classes (structured according to the defined query templates) are embedded directly and sequentially within the main prompt, immediately preceding the actual query sentence that the model needs to classify.
- **Conversational:** Examples are structured to simulate a dialogue. Each example consists of a "user" turn providing a citation sentence and an "assistant" turn providing the correct class. This series of example turns is provided before the final "user" query, which presents the new citation sentence for the model to classify. This format is also commonly referred to as a conversational or turn-taking format.

Temperature (T). Temperature is a hyperparameter that controls the randomness or creativity of a language model's outputs by adjusting the probability distribution of possible next tokens [16]. Lower temperatures (i.e., close to 0) correspond to greedy decoding, where the model deterministically selects the most probable token at each step, while higher temperatures (closer to 1) introduce greater variability by allowing the model to sample from a broader range of options.

For classification tasks, we aim for the model to output the most probable class label. Therefore, we use a temperature of 0 as the baseline, ensuring fully deterministic predictions. To explore how controlled randomness affects classification performance, we also evaluate higher temperatures, such as 0.2, 0.5, and 1.0. A temperature of 0.2 introduces a small degree of randomness but still heavily favors the most probable answer, while 0.5 strikes a balance between randomness and determinism. At 1.0, randomness is maximized, allowing us to assess whether excessive stochasticity degrades performance.

2.4 Technical Specifications

We conducted our experiments on an M1 Max Mac Studio with 64GB of memory, chosen to demonstrate the feasibility of running inference for Citation Intent Classification on commodity hardware.

For model hosting, we used LM Studio[7] which offers an intuitive interface for testing and interacting with models hosted on HuggingFace or locally. It

[7] https://lmstudio.ai/

Table 1. Highest Performance by Model.

SciCite			ACL-ARC		
Rank	**Model**	**F1-Score**	**Rank**	**Model**	**F1-Score**
1	Qwen 2.5 – 14B	78.33	1	Qwen 2.5 – 14B	63.68
2	Gemma 2 – 27B	77.86	2	Gemma 2 – 27B	58.95
3	Mistral Nemo – 12B	77.39	3	Gemma 2 – 9B	57.19
4	Gemma 2 – 9B	75.12	4	Qwen 2 – 7B	51.26
5	Phi 3 Medium – 14B	74.67	5	LLaMA 3.1 – 8B	48.45
6	LLaMA 3 – 8B	74.39	6	Mistral Nemo – 12B	48.11
7	Qwen 2 – 7B	72.89	7	Phi 3.5 Mini – 3.8B	43.74
8	LLaMA 3.1 – 8B	72.46	8	Phi 3 Medium - 14B	43.46
9	Gemma 2 - 2B	68.79	9	Gemma 2 - 2B	40.96
10	Phi 3.5 Mini – 3.8B	68.25	10	LLaMA 3.2 - 3B	40.07
11	LLaMA 3.2 - 3B	67.99	11	LLaMA 3 - 8B	38.06
12	LLaMA 3.2 - 1B	45.44	12	LLaMA 3.2 - 1B	24.60

also supports a local server mode compatible with the OpenAI API, allowing interaction through an API accessible in multiple programming languages. A command-line interface (CLI) tool[8] is also available for managing the server without using the UI.

3 Experimental Results and Analysis

This section presents the results of our experiments, focusing on model performance across various configurations and parameter settings. An initial overview of the peak F1-score achieved by each of the twelve models on both the SciCite and ACL-ARC datasets is summarized in Table 1. The subsections that follow provide a detailed analysis of top-performing model configurations, evaluate overall model rankings, and explore the influence of specific parameters.

3.1 Model Performance over Configurations

Our first evaluation focuses on identifying the top-performing models across the configurations described in Section 2. We employed two complementary approaches: (i) the Best-Performer Evaluation identifies the single best model, and (ii) the Ranked Evaluation considers the relative performance of all models across configurations.

Best-Performer Evaluation. To identify the most effective models across configurations, we initially conducted a Best-Performer Evaluation. In particular, for each configuration, we examined a metrics table containing precision, recall, F1-score, and accuracy for all models. The model with the highest F1-score[9] in each configuration was selected as the best-performing model – in case of a tie, we used Accuracy as the deciding factor.

[8] https://github.com/lmstudio-ai/lms
[9] All F1-scores reported in this paper are macro-averaged F1-scores (macro-F1).

Table 2. Model Ranking based on Best-Performing Count.

Dataset	Rank	Model	Overall	Zero-Shot	One-Shot	Few-Shot	Many-Shot
SciCite	1	Qwen 2.5 – 14B	125	24	42	28	31
	2	Mistral Nemo – 12B	25	0	0	14	11
	4	Gemma 2 – 27B	10	0	0	4	6
	4	Gemma 2 – 9B	8	0	6	2	0
ACL-ARC	1	Qwen 2.5 – 14B	129	19	22	40	28
	2	Gemma 2 – 27B	28	0	21	7	0
	3	Gemma 2 – 9B	11	5	5	1	0

Table 3. RRF-based Model Ranking.

SciCite			ACL-ARC		
Rank	**Model**	**RankScore**	**Rank**	**Model**	**RankScore**
1	Qwen 2.5 – 14B	144.163	1	Qwen 2.5 – 14B	146.496
2	Mistral Nemo – 12B	68.479	2	Gemma 2 – 27B	62.973
3	Gemma 2 – 27B	63.369	3	Gemma 2 – 9B	55.318
4	Gemma 2 – 9B	59.760	4	Qwen 2 – 7B	41.496
5	Qwen 2 – 7B	38.617	5	LLaMA 3.1 – 8B	40.844
6	LLaMA 3.1 – 8B	27.624	6	Phi 3.5 Mini – 3.8B	30.470
7	LLaMA 3 – 8B	26.159	7	Mistral Nemo – 12B	30.285
8	Phi 3.5 Mini – 3.8B	22.088	8	LLaMA 3.2 - 3B	23.984
9	Phi 3 Medium – 14B	19.370	9	Phi 3 Medium - 14B	21.269
10	Gemma 2 - 2B	19.253	10	LLaMA 3.2 - 1B	16.796
11	LLaMA 3.2 - 3B	18.286	11	Gemma 2 - 2B	16.534
12	LLaMA 3.2 - 1B	14.136	12	LLaMA 3 - 8B	16.359

The aggregated results of our evaluation are presented in Table 2. It is evident that Qwen 2.5 14B was the most dominant model on both datasets, significantly outperforming others across all prompting methods.

Ranked Evaluation. While the Best-Performer Evaluation provided insights into the most dominant models across configurations, it lacked the ability to capture nuances among high-performing models. For instance, a model that consistently ranked second in multiple configurations would not be reflected in that approach. To address this limitation, we adopted a ranking methodology inspired by Reciprocal Rank Fusion (RRF) [6]. This approach allowed us to evaluate all 12 models considering their relative performance across different configurations.

For each experimental configuration $c \in C$, models $M = \{m_1, m_2, \ldots, m_{12}\}$ were ranked based on their F1-scores – in case of a tie, Accuracy was used to determine the rank. Each model was then assigned a score based on its rank, where the score $S(m_k, c)$ for the k-th ranked model in configuration c was defined as $S(m_k, c) = \frac{1}{k}$. This score is inversely proportional to the rank, meaning that higher-ranked models (lower-k) receive higher scores, while lower-ranked models (higher-k) receive lower scores. To calculate the overall ranking score for each model m, we aggregated its scores across all configurations $c \in C$:

$$RankingScore(m) = \sum_{c \in C} S(m_k, c) \tag{1}$$

Table 3 presents the results of this ranked evaluation. The ranked evaluation aligns with the Best-Performer Evaluation in identifying Qwen 2.5 14B as the

Table 4. Parameter Performance Analysis of Top 5% Configurations.

Parameter	Setting	SciCite		ACL-ARC	
		Count	Percent	Count	Percent
In-context Learning Method	Few-shot	47	46.53%	55	61.11%
	Many-shot	38	35.64%	21	23.33%
	One-shot	14	13.86%	14	15.55%
	Zero-shot	4	3.96%	-	-
Temperature	0.0	31	30.69%	23	25.55%
	0.2	30	29.70%	22	24.44%
	0.5	20	19.80%	22	24.44%
	1.0	20	19.80%	23	25.55%
System Prompt	SP3	45	44.55%	38	42.22%
	SP2	28	27.72%	24	26.66%
	SP1	28	27.72%	28	31.11%
Query Template	Multiple-Choice	80	79.21%	58	64.44%
	Simple	21	20.79%	32	35.55%
Example Presentation	Conversational	53	54.64%	59	65.55%
	Inline	44	45.36%	31	34.44%

most dominant model. This view also allows us to gain insights into the relative performance of other the models, highlighting distinctions among high performers, as well as the performance of lower-ranked models.

3.2 Parameter Performance Analysis

In this section, we examine how different parameter configurations drive model performance. Identifying the most impactful parameters was challenging due to the extensive search space (168 configurations per dataset, totaling $3,841$ experiments for all models). To address this, we conducted a Quantile-Based Analysis focusing on the top 5% of configurations based on F1-scores. Examining parameter distributions within this high-performing subset (Table 4) reveals trends for optimal configurations across both datasets.

For the *In-context Learning Method*, Few-shot prompting was most prevalent in top SciCite configurations (46.53%), with Many-shot also prominent (35.64%). This Few-shot preference was stronger on ACL-ARC (61.11%), where Many-shot (23.33%) and One-shot (15.55%) were less frequent, and Zero-shot was absent from the top 5%. These results suggest moderate example counts (Few-shot) are broadly effective, though Many-shot also performs well on SciCite.

The optimal *Temperature* settings showed divergence between the datasets. SciCite favored lower values, with T=0.0 (30.69%) and T=0.2 (29.70%) being most frequent, indicating a preference for deterministic outputs, while higher temperatures (T=0.5, T=1.0, both 19.80%) were less common. In contrast, ACL-ARC displayed a more balanced distribution, where T=0.0 and T=1.0 (both 25.55%) were slightly ahead of T=0.2 and T=0.5 (both 24.44%), suggesting that varied levels of randomness can yield top results.

A shared preference for *System Prompt* SP3 emerged across both datasets, found in 44.55% of top SciCite and 42.22% of top ACL-ARC configurations. Secondary preferences, however, varied. SciCite showed equal representation for

Table 5. Chi-Square Test of Independence results for parameter settings and F1-Scores. *(Significance levels: ***p < 0.001, **p < 0.01, *p < 0.05)*

Parameter	SciCite		ACL-ARC	
	χ^2	p-value	χ^2	p-value
In-context Learning Method	29.857	1.48e-06 ***	49.025	1.29e-10 ***
Temperature	4.617	0.202	0.047	0.9973
System Prompt	6.024	0.0491 *	3.647	0.1615
Query Template	35.063	3.19e-09 ***	7.305	0.0068 **
Example Presentation	0.699	0.4031	8.604	0.0033 **

SP1 and SP2 (both 27.72%), whereas ACL-ARC favored SP1 (31.11%) over SP2 (26.66%). This indicates an advantage for the structured SP3, though simpler prompts also achieved high performance.

The *Query Template* parameter revealed a decisive trend, as the Multiple-Choice template was overwhelmingly favored on both SciCite (79.21%) and ACL-ARC (64.44%) when compared to the Simple template (SciCite: 20.79%; ACL-ARC: 35.55%). This highlights the significant effectiveness of explicit multiple-choice options.

Finally, considering the *Example Presentation Format*, the Conversational style was more frequent in top configurations for SciCite (54.64%) and especially ACL-ARC (65.55%), over the Inline format (SciCite: 45.36%; ACL-ARC: 34.44%). This suggests structuring examples conversationally generally yields better results.

These findings offer valuable insights into the parameter settings that consistently enhance performance across the two datasets.

Chi-Square Statistical Test (χ^2). To complement our parameter performance analysis, we also conducted a *Chi-Square Test of Independence* [12] to evaluate the relationship between parameter settings and F1-scores. This approach allowed us to determine whether specific parameter settings were significantly associated with performance outcomes and validate the results of our primary analysis. The results of the test (summarized in Table 5) largely reinforces the findings from our quantile-based analysis regarding the influence of different parameter settings.

The *In-context Learning Method* showed a highly significant association with F1-scores for both SciCite ($p < 0.001$) and ACL-ARC ($p < 0.001$). This statistically underscores the importance of the number of examples provided, aligning with our earlier observation that Few-shot and Many-shot configurations were prevalent among the top performers.

Similarly, the *Query Template* was found to be highly significant for SciCite ($p < 0.001$) and also significant for ACL-ARC ($p < 0.01$). This supports the strong dominance of the Multiple-Choice template in the top 5% configurations, highlighting its critical role in guiding model output effectively.

Other parameters presented more nuanced relationships. The *System Prompt* was statistically significant for SciCite ($p < 0.05$), corroborating the advantage seen for SP3 in its quantile analysis. However, for ACL-ARC, while SP3 was also frequent in top configurations, the choice of system prompt did not emerge as a statistically significant factor overall.

Conversely, the *Example Presentation Format* was not statistically significant for SciCite, which aligns with the relatively balanced distribution (Conversational at 54.64%, Inline at 45.36%) observed in its top configurations. For ACL-ARC, however, this parameter was significant ($p < 0.01$), providing statistical backing for the clearer preference (65.55%) for the Conversational method seen in its quantile analysis.

Finally, the *Temperature* parameter did not show a statistically significant relationship with F1-scores on either dataset. This finding is consistent with the quantile analysis where, for SciCite, multiple temperature settings were present in top configurations (though lower ones were more frequent), and for ACL-ARC, the distribution was particularly balanced, suggesting that temperature, within the tested ranges, was not as decisive a factor as other parameters.

4 Fine-tuning

In this section, we investigate the impact of fine-tuning in the performance of the instruction-tuned Qwen 2.5 14B model on citation intent classification.

4.1 Training Configuration

For this experiment, we used the SciCite and ACL-ARC datasets, which were converted into the Alpaca format [30]. This format includes a system prompt, an instruction, the citing sentence, and the true label. The use of Supervised Fine-Tuning (SFT) was motivated by its ability to adapt pre-trained models to specific tasks using minimal labeled data, making it an effective approach for citation intent classification. To perform the fine-tuning, we used the original training set of each dataset (8,243 examples for SciCite and 1,688 for ACL-ARC). For testing, we used the original test sets as seen in the previous experiments.

To fine-tune the Qwen 2.5 14B Instruct model, we used LLaMA-Factory [33] on an AWS g6e.12xlarge EC2 instance equipped with 4 NVIDIA L40S GPUs, providing a combined GPU memory of 192GB. The training process was conducted using fp16 mixed precision to optimize memory usage and speed. To prevent memory issues during fine-tuning, we employed DeepSpeed ZeRO Stage 3 Offload [27], which enabled efficient memory management by offloading optimizer states and gradients to the CPU. The training parameters included a learning rate of 5e-5, a batch size of 16, and 10 epochs. The model was optimized using AdamW, with a cutoff length of 512 tokens for input sequences.

To optimize the fine-tuning process, we employed Low-Rank Adaptation (LoRA), a widely used parameter-efficient fine-tuning (PEFT) method [14,13]. LoRA enables efficient adaptation of large language models by freezing the pre-trained model weights and introducing trainable low-rank matrices into the Transformer layers, significantly reducing the number of trainable parameters

Table 6. Training Parameters for Fine-Tuning Qwen 2.5 14B.

Parameter	Value	Explanation
Learning Rate	5e-5	Controls step size for updating model weights during training.
Epochs	10	Number of complete passes through the entire training dataset.
Batch Size	16	Number of training examples processed in one iteration.
Cutoff Length	512	Maximum sequence length (in tokens) for model inputs.
Optimizer	AdamW	Algorithm used to adjust model weights to minimize loss.
Compute Type	fp16	Numerical precision used for computations.
Warmup Steps	500	Number of initial steps where learning rate gradually increases.
DeepSpeed Offload	Enabled	Moves optimizer states/grads to RAM to save GPU memory.
DeepSpeed Stage	3	Level of DeepSpeed ZeRO optimization.
LoRA Rank	8	Dimension of the trainable low-rank matrices added by LoRA.
LoRA Alpha	16	Scaling factor for the LoRA updates relative to the rank.
LoRA Dropout	0.1	Dropout probability applied to LoRA layers for regularization.

while maintaining performance. For this experiment, we configured LoRA with a rank of 8, alpha of 16, and a dropout of 0.1, which allowed us to fine-tune the model effectively without exceeding memory constraints. The outlined training parameters are also summarized in Table 6.

4.2 Results

After fine-tuning, we evaluated the new models using the same prompting methods as in Section 2.3 (i.e., zero-shot, one-shot, few-shot, and many-shot) to draw comparisons with the instruction-tuned Qwen 2.5 14B baseline. While our initial extensive experiments with in-context learning focused on 8-bit quantized models (Q8), for this fine-tuning evaluation, we also assessed the 16-bit floating-point (FP16) versions, to examine the impact of numerical precision on performance.

The overall results, presented in Table 7, clearly show that supervised fine-tuning significantly improved performance on both the SciCite and ACL-ARC datasets. On SciCite, the fine-tuned FP16 model achieved a peak F1-score of 86.84. Compared to the instruction-tuned FP16 baseline (F1-score of 80.41), this represents an 8.0% relative improvement. On ACL-ARC, the fine-tuned Q8 model obtained the highest F1-score of 68.48. This constitutes a relative improvement of nearly 4.3% over the instruction-tuned FP16 baseline (F1-score of 65.64). Overall, the highest F1-score achieved after fine-tuning was 86.84 on SciCite and 68.48 on ACL-ARC.

Examining the impact of numerical precision, the fine-tuned FP16 and Q8 models showed minor performance differences. FP16 models generally outperformed Q8 on SciCite, while the fine-tuned Q8 model slightly outperformed its FP16 counterpart on ACL-ARC. This suggests that while FP16 precision can offer a slight advantage, Q8 remains highly competitive and may offer robustness, particularly for tasks with more granular classification schemes like ACL-ARC's 6-class setup.

Table 8 highlights the impact of fine-tuning across the different prompting methods. On SciCite, fine-tuning led to consistent F1-score improvements across all prompting scenarios. These gains were particularly substantial in zero-shot and one-shot settings; for instance, the fine-tuned Q8 model improved upon the

Table 7. F1-Score Performance of Qwen 2.5 – 14B Instruct and Fine-tuned variants.

Model	SciCite	ACL-ARC
Qwen 2.5 – 14B Instruct Q8	78.33	63.68
Qwen 2.5 – 14B Instruct FP16	80.41	65.64
Qwen 2.5 – 14B Fine-tuned Q8	86.47	**68.48**
Qwen 2.5 – 14B Fine-tuned FP16	**86.84**	67.73

Table 8. F1-Score Performance of Qwen 2.5 – 14B Instruct and Fine-tuned variants, divided by prompting method.

Dataset	Model	Zero-Shot	One-Shot	Few-Shot	Many-Shot
SciCite	Instruct Q8	75.74	76.32	78.33	78.23
	Instruct FP16	75.38	77.22	80.41	78.94
	Fine-tuned Q8	**84.84**	**85.46**	86.47	85.62
	Fine-tuned FP16	84.49	85.38	**86.84**	**85.79**
ACL-ARC	Instruct Q8	52.29	60.02	62.86	63.68
	Instruct FP16	52.47	61.73	65.64	64.39
	Fine-tuned Q8	59.92	**68.48**	**67.58**	67.19
	Fine-tuned FP16	**60.29**	67.73	66.94	**67.62**

instruction-tuned FP16 baseline by over 12.5% in zero-shot (84.84 vs 75.38) and by nearly 10.7% in one-shot (85.46 vs 77.22). In few-shot and many-shot for SciCite, performance with the fine-tuned FP16 model approached saturation, achieving the highest F1-scores of 86.84 and 85.79, respectively.

Similar trends were observed on the ACL-ARC dataset. The fine-tuned models generally outperformed their instruction-tuned counterparts across prompting methods, with the fine-tuned Q8 model achieving the highest F1-scores in one-shot (68.48) and few-shot (67.58) setups. In contrast, the fine-tuned FP16 model yielded stronger results in zero-shot (60.29) and many-shot (67.62).

These results demonstrate that supervised fine-tuning not only boosts overall performance but also particularly enhances generalization in low-context (zero-shot and one-shot) scenarios, while still effectively leveraging the additional context provided in few-shot and many-shot settings.

4.3 Discussion

The primary goal of this work was to investigate the capability of large language models to perform citation intent classification, rather than to compete directly with state-of-the-art methods. Nonetheless, it is worth noting that the fine-tuned model achieved performance on SciCite within less than a 3% margin of the best-reported results in the literature, surpassing most of the existing approaches (see Table 9 - Section 5). This demonstrates that LLMs can approach the performance of specialized models, even without task-specific architectures or optimization.

We consider a key advantage of LLMs to be their ease of use and deployment. Tools such as LM Studio and Ollama[10] allow models like those outlined in this paper to be deployed locally with zero technical expertise, making them accessible to users without a computer science background. In addition to their accessibility, LLMs offer significant adaptability. Unlike traditional methods, which

[10] https://ollama.com/

often require complex pretraining or domain-specific tuning, LLMs can be fine-tuned for a wide range of scientometric tasks, such as citation recommendation, paper summarization, or trend analysis, without the need for bespoke architectures. Furthermore, LLMs can scale effectively to new scientific domains or datasets, requiring only small amounts of task-specific data to adapt to under-explored scientific fields; our fine-tuned models required only the several thousand citing sentences provided by our task-specific datasets for adaptation. This contrasts sharply with models like SciBERT, whose effectiveness stems from deliberate pre-training exclusively on a large corpus of scientific papers (millions of articles), optimizing it for scientific language understanding. While the general-purpose LLMs used in our study were likely exposed to scientific text within their vast, diverse pre-training data drawn from the web, this exposure is *incidental* rather than *targeted*. They were not specifically pre-trained or architected with the primary goal of processing scientific literature, unlike SciBERT. Our results demonstrate that even without such specialized scientific pre-training, general-purpose LLMs can achieve competitive performance through efficient fine-tuning on minimal task-specific data.

The promising results observed in this study suggest that the performance of LLMs in citation intent classification can be further improved with techniques such as chain-of-thought prompting and reasoning-focused models. These methods could enhance the models' ability to better distinguish subtle differences in intents, refining their predictions and improving overall classification accuracy.

5 Related Work

The theoretical beginnings of citation analysis can be traced back to foundational works such as [9] identification of reasons for citation [23] studies on citation function. Early annotation schemes, such as those by [29], were later adapted by [32] for supervised machine learning approaches to citation classification.

[17] introduced the ACL-ARC dataset, which contains nearly $2,000$ citations from papers in the NLP field, annotated with a classification scheme of six classes. [25] extended this classification scheme by refining the comparison class to capture similarities, differences, and disagreement. Around the same time, [5] proposed a multitask model incorporating structural information from scientific papers. They also introduced the SciCite dataset, which is significantly larger and spans multiple scientific domains, with three intent classes.

[3] introduced SciBERT, a BERT-based [7] encoder language model pre-trained specifically on scientific text, which has since become the backbone of many citation intent classification methods. SciBERT has been widely adopted due to its ability to generalize across scientific domains. For example, [22] introduced ImpactCite, an XLNet-based method for citation impact analysis, which was later used by [24] to achieve state-of-the-art results on the SciCite dataset. Paolini et al. demonstrated the effectiveness of ensemble classifiers combining fine-tuned SciBERT and XLNet models.

Table 9. Reported F1-Scores of notable works from the literature, sorted chronologically.

Method	SciCite	ACL-ARC
Feature-rich Random Forest [17]	–	53.00
Structural Scaffolds [5]	84.00	67.90
SciBERT [3]	85.22	–
ImpactCite [22]	88.93	–
CitePrompt [20]	86.33	68.39
EnsIntWS [24]	89.46	–
EnsIntWoS [24]	88.48	–
MTL Finetuning (Search) [28]	85.25	64.56
MTL Finetuning (TRL) [28]	85.35	75.57

Recent research has continued to explore PLM-based methods for citation intent classification. [20] used a prompt-based learning approach on SciBERT to identify citation intent, while [28] achieved state-of-the-art performance on the ACL-ARC dataset by proposing a multi-task learning framework that jointly fine-tunes SciBERT on a dataset of primary interest together with multiple auxiliary datasets to take advantage of additional supervision signals. [19] explored various prompting and tuning strategies on SciBERT, including fixed and dynamic context prompts, and found that parameter updating with prompts improved performance. They also briefly experiment on LLMs by evaluating the zero-shot performance of GPT-3.5, which performed well on their recently introduced ACT2 dataset but poorly on the ACL-ARC dataset. However, GPT-3.5 was not evaluated on SciCite. In contrast, our work is the first to focus entirely on evaluating and fine-tuning numerous open-weight, general-purpose LLMs without using models pre-trained explicitly and exclusively for the scientific domain.

Table 9 summarizes the F1-scores of notable works, highlighting the progression of methods and datasets. SciBERT-based methods dominate, while our work is the first to examine LLMs on this task without any reliance on SciBERT.

6 Conclusions

This study investigated open Large Language Models (LLMs) for citation intent classification, demonstrating their viability with in-context learning, particularly when guided by optimized prompting strategies. We found that supervised fine-tuning with minimal data significantly boosts performance; notably, our fine-tuned Qwen 2.5 14B model achieved relative F1-score improvements of approximately 8% on SciCite and 4.3% on ACL-ARC over strong instruction-tuned baselines, reaching performance levels competitive with specialized systems. The detailed insights from our prompting parameter experiments, combined with our openly available evaluation framework and models, aim to facilitate further research and application of LLMs in scientometrics.

Acknowledgments. This work has received funding from the EU's Horizon Europe framework programme as part of the SciLake (GA: 101058573) and GraspOS (GA: 101095129) projects. Part of this work utilized Amazon's cloud computing services, which were made available via GRNET under the OCRE Cloud framework.

References

1. Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., Awan, A.A., ..., N.B., Zhou, X.: Phi-3 technical report: A highly capable language model locally on your phone (2024), https://arxiv.org/abs/2404.14219
2. Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Zhu, X.D..T.: Qwen technical report (2023), https://arxiv.org/abs/2309.16609
3. Beltagy, I., Lo, K., Cohan, A.: Scibert: A pretrained language model for scientific text. arXiv preprint arXiv:1903.10676 (2019)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Amodei, P.D..D.: Language models are few-shot learners (2020), https://arxiv.org/abs/2005.14165
5. Cohan, A., Ammar, W., van Zuylen, M., Cady, F.: Structural scaffolds for citation intent classification in scientific publications. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 3586–3596. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1361, https://www.aclweb.org/anthology/N19-1361
6. Cormack, G.V., Clarke, C.L., Buettcher, S.: Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. pp. 758–759 (2009)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019), https://arxiv.org/abs/1810.04805
8. Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Liu, T., Chang, B., Sun, X., Li, L., Sui, Z.: A survey on in-context learning (2024), https://arxiv.org/abs/2301.00234
9. Garfield, E., et al.: Can citation indexing be automated. In: Statistical association methods for mechanized documentation, symposium proceedings. vol. 269, pp. 189–192. Washington (1965)
10. Gemma, Riviere, M., Pathak, S., Sessa, P.G., Hardin, C., ..., S.B., Andreev, A.: Gemma 2: Improving open language models at a practical size (2024), https://arxiv.org/abs/2408.00118
11. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., ..., A.A.D., Ma, Z.: The llama 3 herd of models (2024), https://arxiv.org/abs/2407.21783
12. Greenwood, P.E., Nikulin, M.S.: A guide to chi-squared testing. John Wiley & Sons (1996)
13. Han, Z., Gao, C., Liu, J., Zhang, J., Zhang, S.Q.: Parameter-efficient fine-tuning for large models: A comprehensive survey (2024), https://arxiv.org/abs/2403.14608
14. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models (2021), https://arxiv.org/abs/2106.09685
15. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., de las Casas ..., D., Lacroix, T., Sayed, W.E.: Mistral 7b (2023), https://arxiv.org/abs/2310.06825
16. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models. 3rd edn. (2025), https://web.stanford.edu/~jurafsky/slp3/, online manuscript released January 12, 2025

17. Jurgens, D., Kumar, S., Hoover, R., McFarland, D., Jurafsky, D.: Measuring the evolution of a scientific field through citation frames. Transactions of the Association for Computational Linguistics **6**, 391–406 (2018)
18. Kanellos, I., Vergoulis, T., Sacharidis, D., Dalamagas, T., Vassiliou, Y.: Impact-based ranking of scientific publications: A survey and experimental evaluation. IEEE Trans. Knowl. Data Eng. **33**(4), 1567–1584 (2021). https://doi.org/10.1109/TKDE.2019.2941206, https://doi.org/10.1109/TKDE.2019.2941206
19. Kunnath, S.N., Pride, D., Knoth, P.: Prompting strategies for citation classification. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. p. 1127–1137. CIKM '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3583780.3615018, https://doi.org/10.1145/3583780.3615018
20. Lahiri, A., Sanyal, D.K., Mukherjee, I.: Citeprompt: Using prompts to identify citation intent in scientific papers. In: 2023 ACM/IEEE Joint Conference on Digital Libraries (JCDL). pp. 51–55 (2023). https://doi.org/10.1109/JCDL57899.2023.00017
21. Lo, K., Wang, L.L., Neumann, M., Kinney, R., Weld, D.: S2ORC: The semantic scholar open research corpus. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4969–4983. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.447, https://aclanthology.org/2020.acl-main.447/
22. Mercier, D., Rizvi., S.T.R., Rajashekar., V., Dengel., A., Ahmed., S.: Impactcite: An xlnet-based solution enabling qualitative citation impact analysis utilizing sentiment and intent. In: Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART. pp. 159–168. INSTICC, SciTePress (2021). https://doi.org/10.5220/0010235201590168
23. Moravcsik, M.J., Murugesan, P.: Some results on the function and quality of citations. Social Studies of Science **5**(1), 86–92 (1975). https://doi.org/10.1177/030631277500500106, https://doi.org/10.1177/030631277500500106
24. Paolini, L., Vahdati, S., Di Iorio, A., Wardenga, R., Heibi, I., Peroni, S.: "why do you cite?" an investigation on citation intents and decision-making classification processes (2024). https://doi.org/10.5281/ZENODO.11841798, https://zenodo.org/doi/10.5281/zenodo.11841798
25. Pride, D., Knoth, P.: An authoritative approach to citation classification. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020. p. 337–340. JCDL '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3383583.3398617, https://doi.org/10.1145/3383583.3398617
26. Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., ..., B.Z., Qiu, Z.: Qwen2.5 technical report (2025), https://arxiv.org/abs/2412.15115
27. Ren, J., Rajbhandari, S., Aminabadi, R.Y., Ruwase, O., Yang, S., Zhang, M., Li, D., He, Y.: Zero-offload: Democratizing billion-scale model training (2021), https://arxiv.org/abs/2101.06840
28. Shui, Z., Karypis, P., Karls, D.S., Wen, M., Manchanda, S., Tadmor, E.B., Karypis, G.: Fine-tuning language models on multiple datasets for citation intention classification (2024), https://arxiv.org/abs/2410.13332
29. Spiegel-Rosing, I.: Science studies: Bibliometric and content analysis. Social Studies of Science **7**(1), 97–113 (1977)
30. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca (2023)

31. Teo, S.: How i won singapore's gpt-4 prompt engineering competition. https://towardsdatascience.com/how-i-won-singapores-gpt-4-prompt-engineer\ing-competition-34c195a93d41 (2023)
32. Teufel, S., Siddharthan, A., Tidhar, D.: Automatic classification of citation function. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. p. 103–110. EMNLP '06, Association for Computational Linguistics, USA (2006)
33. Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., Ma, Y.: Llamafactory: Unified efficient fine-tuning of 100+ language models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations). Association for Computational Linguistics, Bangkok, Thailand (2024), http://arxiv.org/abs/2403.13372