

# DACIP-RC: Domain Adaptive Continual Instruction Pre-Training via Reading Comprehension on Business Conversations



Elena Khasanova\*, Harsh Saini\*, Md Tahmid Rahman Laskar\*, Xue-Yong Fu\*,  
Cheng Chen, Shashi Bhushan TN

Dialpad Inc.

{xue-yong, elena.khasanova, tahmid.rahman, hsaini, cchen, sbhushan}@dialpad.com

## Abstract

The rapid advancements in Large Language Models (LLMs) have enabled their adoption in real-world industrial scenarios for various natural language processing tasks. However, the high inference cost of large-scale LLMs makes their deployment impractical, necessitating the use of smaller models. Despite their efficiency, smaller LLMs lack robust zero-shot instruction-following capabilities across diverse domains, limiting their adaptability to dynamic user requirements. Traditional fine-tuning approaches exacerbate this issue by inducing catastrophic forgetting, reducing the model’s generalization ability for unseen tasks. In this paper, we propose Domain Adaptive Continual Instruction Pre-Training via Reading Comprehension (DACIP-RC), a continual pre-training technique that enhances smaller LLMs’ domain adaptability for business conversational tasks. Unlike conventional pre-training approaches that rely on next-token prediction, DACIP-RC generates diverse task instructions and responses via reading comprehension on conversation transcripts, enabling better instruction generalization. Our empirical evaluations demonstrate that DACIP-RC significantly improves zero-shot generalization across a wide range of business conversational tasks, including meeting summarization, action item generation, and call purpose identification. To the best of our knowledge, this is the first work to apply instruction pre-training on business conversational data, providing insights into how industries can leverage proprietary datasets for domain adaptation.

## 1 Introduction

Recently, there has been a huge increase in the industrial adoption of Large Language Models (LLMs) to build real-world NLP features (Laskar et al., 2023b; Fu et al., 2024; Urlana et al., 2024). This is particularly due to the in-context learning

capability of LLMs, which allows them to follow user-specified instructions to solve diverse tasks without requiring any fine-tuning on task-specific data (Ouyang et al., 2022; Zhao et al., 2023). This is a major improvement over early transformer-based language models (Lewis et al., 2020; Zhang et al., 2020; Raffel et al., 2020; Vaswani et al., 2017; Devlin et al., 2019) that did not have such instruction-following capabilities (Zhang et al., 2023). More importantly, this capability has revolutionized NLP, with the release of numerous LLMs having instruction-following capabilities achieving impressive zero-shot performance across a wide range of tasks (Laskar et al., 2023a; Qin et al., 2023; Bang et al., 2023).

Despite the rapid progress in LLM development, the practical use of LLMs in real-world situations is hindered by their significant inference costs (Fu et al., 2024). These costs increase with the size of the model, even though larger models offer superior instruction-following and zero-shot generalization abilities (Wan et al., 2024; Wang et al., 2024). Since the smaller models do not possess the same level of zero-shot instruction-following capabilities as their larger counterparts, it is important to adapt the smaller LLMs to the specific domain in which they will be deployed.

One straightforward way in this regard could be to fine-tune (Zhang et al., 2023) smaller LLMs in task-specific datasets. However, fine-tuning LLMs to solve only certain tasks would limit their capability if there is a change in user requirements in the real world. For instance, suppose a model is just fine-tuned to generate a concise summary of a given conversation, but in the real world, the user may require the LLM to generate a bullet point summary. Fine-tuned smaller LLMs are also more prone to catastrophic forgetting and so they often lose their zero-shot instruction following capability after fine-tuning (Luo et al., 2023; Huang et al., 2024). Therefore, it is important to adapt smaller

\*Equal Contributions. Sorted by the First Name.

LLMs to the target domain in a way such that they can properly follow diverse user instructions

In this paper, we study how to effectively adapt smaller LLMs ( $\leq 8B$  parameters) on real-world business conversational tasks such that they can achieve zero-shot generalization to perform diverse tasks (e.g. meeting summarization and action item generation, identifying the purpose of calls, etc.) depending on diverse instructions (e.g., long vs short vs bullet point summary generation, responding to multi-query prompts (Laskar et al., 2024), etc.). While continual learning via next token prediction (NTP) (Wu et al., 2024b) objective has demonstrated improved performance in the general domain, applying the NTP objective on noisy conversational data may not be useful to teach LLMs about conversation related tasks since such data lacks task-related information (e.g., in a business meeting, participants may not ask each other to summarize the meeting.). Moreover, Cheng et al. (2024a) also demonstrate that continual pre-training via NTP objective lacks generalization in general NLP tasks.

Inspired by the success of techniques like continual pre-training via reading comprehension-based instruction pre-training (Cheng et al., 2024b) for domain adaptation in general NLP tasks, in this paper, we propose a continual pre-training technique using diverse instructions by applying reading comprehension on conversation transcripts. Our proposed technique, **Domain Adaptive Continual Instruction Pre-Training via Reading Comprehension (DACIP-RC)** successfully adapts LLMs to diverse business conversational tasks, achieving impressive zero-shot generalizability. Our major contributions in this paper are summarized below:

- While prior work mostly studied domain adaptation for NLP tasks on non-conversational data, this paper is the first to propose an effective continual pre-training approach via reading comprehension-based instruction pre-training for better domain adaptation in business conversational tasks.
- We describe the construction of our pre-training dataset in extensive detail to enable others to replicate our approach. Therefore, findings from our extensive experiments will help industries working with conversational data to get insights on how to effectively lever-

age their proprietary datasets for zero-shot generalizability across conversational tasks.

## 2 Related Work

LLMs are initially trained on massive, publicly available internet datasets using the self-supervised NTP objective (Radford, 2018; Radford et al., 2019; Brown et al., 2020), where the model learns to predict the next word in a sequence. However, the data used for this initial training can be vastly different from the specialized, proprietary data used in real-world industry applications. A notable example of this discrepancy is highlighted by Wu et al. (2023), who demonstrate that general-purpose LLMs often struggle to match their performance on public benchmarks when applied to specific real-world tasks since public benchmarks are more aligned with the data they were originally trained.

To bridge this performance gap, a technique known as continual pre-training has emerged (Wu et al., 2024b). This involves taking an existing LLM and further pre-training it on large amounts of a company’s own unlabeled internal data. This process, which also leverages self-supervised learning, helps to adapt the model to the specific nuances and vocabulary of a particular domain (Gururangan et al., 2020), as shown by recent studies (Labrak et al., 2024; Wu et al., 2024a)

Nevertheless, this self-supervised pre-training approach based on the NTP objective is not without its drawbacks. Recent research indicates that while NTP-based continual pre-training on a specific domain can enhance performance on tasks within that domain, it may lead to a decrease in the model’s ability to generalize across diverse NLP tasks (Cheng et al., 2024a,b). To address this, Cheng et al. (2024a) demonstrated that instruction pre-training by generating synthetic task instructions from unlabeled datasets via applying techniques like reading comprehension (Cheng et al., 2024b) could improve domain adaptation and generalization in general NLP tasks. However, prior research on instruction pre-training of LLMs is mostly limited to the general domain, with no prior research studying the effectiveness of domain adaptation via instruction pre-training on conversations.

In this paper, we aim to address the gap in the prior research. Our focus is to investigate the effectiveness of continual instruction pre-training for domain adaptation across conversational tasks. We leverage large amounts of unlabeled business con-

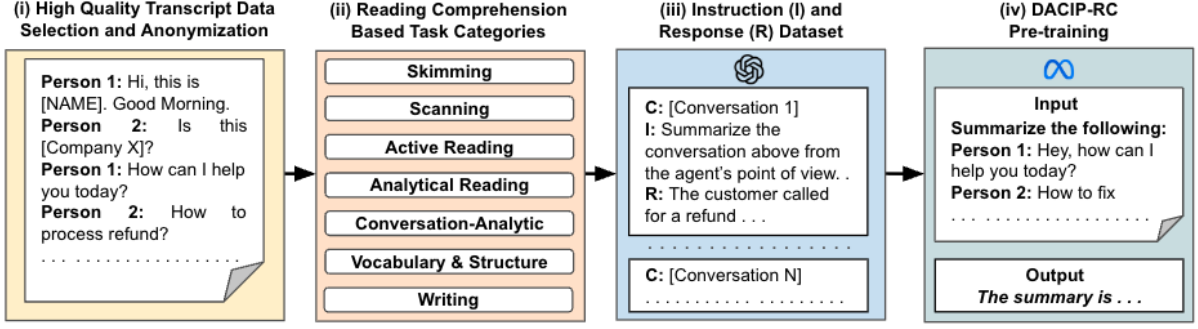


Figure 1: An overview of our proposed DACIP-RC approach: (i) at first high high-quality transcripts are collected and anonymized, (ii) we demonstrate the reading comprehension tasks that were used to construct our (iii) instruction pre-training dataset by generating instructions and responses using a closed-source LLM (e.g., GPT-4o-Mini (OpenAI, 2023)) to (iv) pre-train a smaller open-source LLM (e.g., LLaMA-3.1-8B (Dubey et al., 2024)).

versational data and apply reading comprehension-based techniques for question-and-answer generation from transcripts for instruction pre-training.

### 3 Methodology

In this section, we present our DACIP-RC methodology to construct the instruction pre-training dataset using the reading comprehension technique. First, we demonstrate how we sample the data, followed by discussing how we construct instructions for pre-training inspired by the reading comprehension approach. Finally, we show how we generate and post-process the data to construct our instruction pre-training dataset. Figure 1 summarizes our proposed methodology.

#### 3.1 Data Selection

Our dataset consists of a large number of English-language transcripts from actual business conversations. These conversations, covering a range of topics, industries, and years, were transcribed using our proprietary, in-house automatic speech recognition (ASR) system. To ensure quality and diversity, we only used transcripts that were at least 120 seconds long, had high ASR confidence scores, and involved two or more speakers to reduce the likelihood of genres that are lacking diversity (e.g., voicemails or dropped calls).

We further utilize the approach discussed in Xie et al. (2023) and employ data selection based on token type entropy scores. We then post-process the transcripts by removing personally identifiable information and diversifying the formatting. We follow the approach discussed in Zhang et al. (2024) for data anonymization and use a combination of masking tokens (e.g. <COMPANY\_NAME\_1> instead of the real name) and noising tokens with cus-

tom, contextually relevant replacements (e.g. using a different person name) to allow the model to learn the properties of sensitive tokens without exposing them to the model. We diversify speaker tags (e.g. client vs sales representative) and introduce various transcript formats that include timestamps, different spacing configurations between the turns, etc. to ensure model robustness across different types of transcripts.

#### 3.2 Pre-training Data Construction

##### 3.2.1 Concept

Inspired by the work on reading comprehension approaches to domain adaptation (Cheng et al., 2024b; Jiang et al., 2024), as well as the theory and pedagogy behind reading comprehension, we designed a set of reading comprehension tasks aligned with various reading skills and activities commonly utilized in language and reading education (Brown, 2007; Wright, 2001; Carrell and Carson, 1997; Hill, 2016). These tasks are structured around three primary objectives:

(i) Enhancing the model’s ability to discern the underlying structure of transcripts and accurately retrieve factual and relevant information in downstream tasks.

(ii) Increasing exposure to domain-specific knowledge, particularly within industry-specific business conversational contexts.

(iii) Bridging the gap between general instruction tuning and task-specific fine-tuning.

To accomplish these goals, we developed tasks across the following categories that correspond to four major types of reading:

**(i) Skimming:** Tasks related to big-picture understanding of the conversation.

Examples: *What is the main topic of this conversa-*

tion? What are the stated goals of the speakers in this dialogue?

**(ii) Scanning:** Tasks aimed at extracting specific details from the text.

Examples: *When did Alfildr indicate that the email confirmation would be sent? The meeting is scheduled at \_\_\_ PM next Wednesday.*

**(iii) Active Reading:** Tasks related to actively engaging with the text such as taking notes, summarizing, asking questions, or retrieving information. Example: *Identify topics discussed in this call and write a two-sentence summary of each topic. Then, rewrite this conversation as a marketing email.*

**(iv) Analytical Reading:** Tasks related to discussing underlying assumptions, potential biases, evidence, perspectives of the speakers in the conversation, and various types of divergent questions typically without the right answer.

Example: *Why did the prospect reject the sales agent's proposal? Discuss possible reasons and potential solutions. Explain your rationale based on the provided transcript.*

We also added three more categories that overlap with the reading techniques listed above but have a more specialized focus:

**(v) Conversation-Analytic tasks:** Tasks related to underlying conversational structure and how it is realized in the transcript, e.g. questions addressing turn-taking, organization of utterances, utterance intent, handling misunderstandings.

Example: *Identify questions under discussion (QUD). Analyze the relevance of Speaker 1's inquiries. How did they resolve the QUD?*

**(vi) Vocabulary and Structure:** Tasks related to terminology, structure, and composition of the transcript.

Example: *How is the term 'open balance' defined in financial terms? Reorder the following conversation sections to reflect the correct sequence: A. Computer Issues Discussion. B. Final Arrangements. C. Personal Updates and Plans.*

**(vii) Writing:** Tasks related to text generation tailored to specific industries and genres of business writing to enhance the representation of in-domain vocabulary that typically do not rely on the transcript in the instruction.

Example: *Create a dialogue between a client and a project manager in the Translation Services industry that addresses these topics: Billing Preferences, Current Context, Implementation Timeline.*

This approach combines the benefits of in-

domain knowledge and instruction-following capabilities. Instead of learning a narrowly defined task, the model acquires intermediate competencies for completing diverse downstream tasks with enhanced transcript understanding.

### 3.2.2 Generation Method

**Generation:** For each category described in the previous section, we curated a set of 41 meta prompts that we used to prompt an LLM (GPT-4o-Mini (OpenAI, 2023)) to generate a series of tasks with corresponding answers using a given transcript. The prompts generally follow this template: {transcript}{meta-instruction}{output requirements}{output example}. See an example in Appendix A.1.

For each category, we generate tasks of different varieties, where applicable, which include but are not limited to different types of questions (e.g. yes-no, multiple choice, open-ended), gap-filling and reordering tasks, information extraction, summarization, etc., which may require generating new text, returning a snippet from a transcript verbatim, or a combination of both. We also vary the number of tasks generated per transcript and the output length and format requirements (e.g. a letter, a paragraph, several words, bulleted list).

We developed the meta prompts via extensive prompt-engineering. To minimize API costs and improve generation quality, these meta prompts are designed to generate multiple questions/task instructions per transcript and their corresponding responses in JSON format to ensure easier parsing as seen in Appendix A.1.

**Parsing:** Custom parsers are implemented for different types of output to reliably extract the task instructions and the answers separately. We remove responses that failed to parse due to invalid JSON outputs. After this step, we keep 10% of the generated data unparsed and use original meta prompts with their outputs in JSON format to improve the model's JSON format following capability.

**Postprocessing:** Out of the parsed task-response pairs, we construct prompts and outputs that we use for instruction pre-training. 70% of the pairs are used for single-task prompts, and 30% are used for multitask prompts, with up to 10 instructions for the same transcript. In multitask prompts, the tasks can fall into the same type (e.g. all are multiple choice "scanning" questions) or different categories, and may require outputs in the same or different formats. We also vary whether the in-



| Model                          | Text Classification (F1-Score) |              |       | Text Generation (ROUGE-2) |              |                   |              | Overall Avg. |       |
|--------------------------------|--------------------------------|--------------|-------|---------------------------|--------------|-------------------|--------------|--------------|-------|
|                                | Call Outcome                   | PoC Category | Avg.  | PoC Explanation           | Action Items | Support Call Sum. | Meeting Sum. | Avg.         |       |
| LLaMA-3.1-8B-Instruct          | 13.85                          | 26.46        | 20.16 | 13.71                     | 16.02        | 23.22             | 15.27        | 17.06        | 18.09 |
| LLaMA-3.1-8B-DACIP-RC          | 42.36                          | 54.09        | 48.23 | 15.34                     | 16.40        | 20.46             | 17.05        | 17.31        | 27.62 |
| LLaMA-3.1-8B-Instruct-DACIP-RC | 46.23                          | 53.31        | 49.77 | 14.08                     | 23.89        | 19.46             | 17.68        | 18.78        | 29.11 |

Table 1: Performance on in-domain conversational tasks. Here, ‘Sum.’ denotes ‘Summarization’ and ‘Avg.’ denotes ‘Average’.

structions appear before or after the transcript. See Appendix A.2 for an example of a pretraining task.

The resulting dataset consists of over 26M instances, with an average prompt length (including transcript) of 1448.46 tokens and a response length of 107.09 tokens. This adds up to roughly 25B tokens estimated using *tiktoken*<sup>1</sup> tokenizer.

## 4 Experiments

### 4.1 Training Settings

The pre-training job was conducted using LLaMA-3.1-8B (Dubey et al., 2024) (both base and instruct models) on 8 nodes, each with 8 x NVIDIA H100 80 GB GPUs attached. Deepspeed (Aminabadi et al., 2022) was used in conjunction with PyTorch to orchestrate the training, with relevant training parameters are as follows: datatype: bf16, learning rate:  $2e - 6$ , number of epochs: 1.

### 4.2 Evaluation Benchmarks

**Internal Benchmarks:** Our internal benchmarks consist of the following four tasks (containing 1.5k instances in total):

(i) **Action Items:** The task involves the generation of a list of actionable tasks extracted from a transcript of a conversation. Each task is a short description of an activity that should occur after the conversation has ended.

(ii) **Purpose of Call:** This task considers two cases: (i) classifying the main purpose of a conversation to one of several pre-defined categories (Khasanova et al., 2022) and (ii) free-form generation of the call purpose alongside an explanation.

(iii) **Call Outcome:** The task aims to classify the main outcome of a conversation to one of several pre-defined categories.

(iv) **Summarization:** The task is to generate a concise conversation summary according to specific requirements such as summary length (long, medium, or short) or format (e.g. bullet points). We consider two real-world cases for summarization: (i) summarizing customer-agent support calls, and (ii) summarizing business meetings.

See Appendix A.3 for the prompt for each task.

**External Benchmarks:** For external benchmarking, we use the QMSUM dataset (Zhong et al., 2021), which focuses on query-based meeting summarization and includes 281 test instances.

## 4.3 Experimental Results

### 4.3.1 Performance on Internal Benchmarks

We evaluate our proposed DACIP-RC based LLaMA-3.1-8B models against the baseline LLaMA-3.1-8B-Instruct model on both text classification and text generation tasks in zero-shot. Table 1 summarizes the results across different tasks.

We observe that applying DACIP-RC leads to significant performance improvements on all classification tasks, with the average F-1 score more than doubling for both LLaMA-3.1-8B-DACIP-RC and LLaMA-3.1-8B-Instruct-DACIP-RC in comparison to the baseline. On text generation tasks, the DACIP-RC models also outperform the baseline on ROUGE-2 (Lin, 2004) metric, with the exception of *Support Call Summarization* with a slight performance drop. Particularly noteworthy is the performance gain on *Action Items*, where LLaMA-3.1-8B-Instruct-DACIP-RC achieves a ROUGE-2 increase from 16.02 (baseline) to 23.89.

Based on the autometrics and upon closer inspection of the outputs, we conclude the following:

(i) The advantages of DACIP-RC method are particularly evident on the tasks that rely on a detailed understanding of a transcript, such as *Purpose of Call (PoC)*, *Action Items*, *Call Outcome*;

(ii) On summarization tasks, DACIP-RC brings more significant improvements on *Meeting Summarization*, with longer and more complex multiparty conversations, than on shorter two-party dialogues in the *Support Call Summarization*;

(iii) DACIP-RC models favor shorter and more concise summaries than the baseline model, which can explain the drop in performance observed on *Support Call Summarization*.

Overall, the DACIP-RC method consistently demonstrates its effectiveness in improving the performance across diverse in-domain conversational

<sup>1</sup><https://github.com/openai/tiktoken>

| Model                          | B-S   | R-1   | R-2  | R-L   |
|--------------------------------|-------|-------|------|-------|
| LLaMA-3.1-8B-Instruct          | 36.72 | 11.16 | 3.12 | 7.73  |
| LLaMA-3.1-8B-DACIP-RC          | 49.00 | 18.09 | 4.28 | 11.93 |
| LLaMA-3.1-8B-Instruct-DACIP-RC | 59.74 | 27.57 | 6.32 | 17.95 |

Table 2: Performance comparison on external benchmarks: the QMSUM (Zhong et al., 2021) dataset. Here, ‘B-S’ denotes ‘BERTScore’ (Zhang et al., 2019) and ‘R’ denotes ‘ROUGE’.

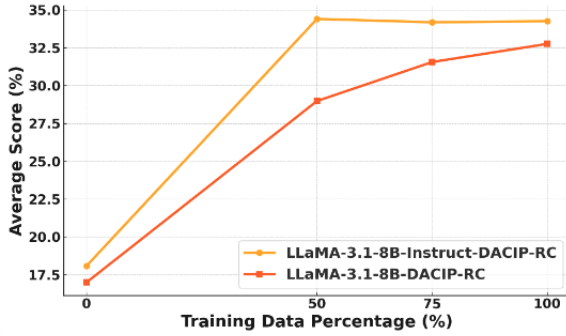


Figure 2: Ablation Tests on Training Data Size

tasks, with the **LLaMA-3.1-8B-Instruct-DACIP-RC** model achieving the best overall performance (having an average score of 29.11).

#### 4.3.2 Performance on External Benchmarks

To evaluate the generalization ability of our models beyond in-domain tasks, we assessed their performance on the QMSUM dataset, a public benchmark for query-focused meeting summarization. Based on the results presented in Table 2, we find that the baseline **LLaMA-3.1-8B-Instruct** achieves the lowest scores across all metrics, with a ROUGE-1 of 11.16, ROUGE-2 of 3.12, and ROUGE-L of 7.73, along with a BERTScore of 36.72.

This demonstrates that the **DACIP-RC** method helps both base and instruct **LLaMA-3.1-8B** models to achieve a huge gain in performance across all metrics, with the **LLaMA-3.1-8B-Instruct-DACIP-RC** model achieving the best result.

#### 4.3.3 Ablation Studies

Our ablation study on training data size (Figure 2) shows that the base LLaMA-3.1 model’s performance consistently improves with more data. The instruct model, however, is more data-efficient; it gets a significant initial boost, but performance gains become negligible as the dataset grows.

#### 4.3.4 Qualitative Evaluation

We also conduct an LLM-judge-based reference-free evaluation using the *Gemini-2.5-Pro* model as the judge on the text generation tasks across 500 test samples by considering factual correctness,

| Model                          | Pointwise (Likert 1-5) | Pairwise Preference |
|--------------------------------|------------------------|---------------------|
| LLaMA-3.1-8B-Instruct          | 2.08                   | 13.7                |
| LLaMA-3.1-8B-Instruct-DACIP-RC | 4.07                   | 85.2                |

Table 3: Reference-free evaluation using *Gemini-2.5-Pro* across 500 test samples.

|                                | PubMedQA | MediQA-QS |
|--------------------------------|----------|-----------|
| LLaMA-3.1-8B-Instruct          | 55.0     | 7.76      |
| LLaMA-3.1-8B-Instruct-DACIP-RC | 57.8     | 9.32      |

Table 4: Out-of-Domain Generalization Results on PubMedQA and MediQA-QS datasets, in terms of Accuracy and ROUGE-2, respectively.

adherence to instructions, format following, and conciseness. The results are as shown in Table 3.

We find that in terms of *pointwise* evaluation (Likert-scale, 1-5), the baseline **LLaMA-3.1-8B-Instruct** model got an average score of 2.08, while our **LLaMA-3.1-8B-Instruct-DACIP-RC** model achieved an average score of 4.07. Moreover, in terms of pairwise evaluation, the **LLaMA-3.1-8B-Instruct-DACIP-RC** model was preferred 85.2% of the time. More interestingly, the DACIP-RC was preferred in 93.7% of the cases in the support call summarization task, in which it performed slightly poorer than the baseline in reference-wise evaluation<sup>2</sup> in terms of ROUGE-2. This evaluation further demonstrates the effectiveness of DACIP-RC.

#### 4.3.5 Out-of-Domain Generalization

While the focus of our work is to demonstrate how we adapt an LLM in a real-world industrial setting for business conversational tasks, we further evaluated our DACIP-RC model in the biomedical domain to investigate its generalizability and the presence of any catastrophic forgetting. We select one classification task, the PubMedQA (Jin et al., 2019) dataset for question answering, and one summarization task, the MediQA-QS (Abacha et al., 2021) dataset for healthcare question summarization. These two datasets are commonly used benchmarks in the biomedical domain for the evaluation of LLMs (Jahan et al., 2024). We compare the performance of DACIP-RC-tuned **LLaMA-3.1-8B-Instruct** with the original zero-shot **LLaMA-3.1-8B-Instruct** evaluated by Jahan et al. (2025). Based on the results given in Table 4, we find that our proposed DACIP-RC approach led to a performance gain even in the out-of-domain biomedical

<sup>2</sup>Differences in performance between reference-free and reference-wise metrics in the evaluation of LLMs are quite common in the literature (Liu et al., 2023).

benchmarks on related tasks (e.g., question answering and summarization). To further demonstrate that typical fine-tuning of LLMs only on a certain task could lead to catastrophic forgetting, we adopt the **Mistral-7B-Instruct** model (Jiang et al., 2023) trained only on the QMSUM-Multi-Query (Zhong et al., 2021; Laskar et al., 2024) dataset for Query-Focused Meeting Summarization from Laskar et al. (2024) and evaluate on our internal text classification datasets (Call Outcome and Purpose of Call Category Classification). We find that while the zero-shot **Mistral-7B-Instruct** achieves an average F1 score of 35.5, the Mistral model fine-tuned only on QMSUM performs very poorly in our classification datasets (average F1 score of 0.0). This demonstrates that typical task-specific fine-tuning can lead to catastrophic forgetting.

#### 4.3.6 Performance Comparison with the Standard NTP Objective

In our proposed DACIP-RC approach, we only predict the response for the given conversation. To investigate the effectiveness of this approach, we pre-train the **LLaMA-3.1-8B** model using the standard NTP objective on our DACIP-RC dataset, consisting of 25M conversations alongside the corresponding DACIP-RC instruction-response pairs. We denote this model as **LLaMA-3.1-8B-Internal-NTP**. Unlike our DACIP-RC recipe (which only predicts the response), this baseline predicts the next token over the entire sequence, which includes the conversation, task instruction, and response. This is done because NTP pre-training only on conversations can destroy the model’s zero-shot instruction-following abilities (Fu et al., 2025). We find that our **LLaMA-3.1-8B-DACIP-RC** model outperforms the **LLaMA-3.1-8B-Internal-NTP** model by a large margin, with the average result across our internal datasets being 29.1 for DACIP-RC, while only 18.5 for the NTP baseline.

#### 4.3.7 Incorporating Structured Output Format for Real World Inference

For real-world inference, generating outputs in a structured format is crucial (Laskar et al., 2024). With structured output such as JSON, developers can parse the results reliably to retrieve the required information to power downstream tasks. Therefore, we further performed experiments to investigate whether our DACIP-RC model is compatible with structured output generation techniques. In this regard, we compare the plain decoding method

| Decoding Method           | Summarization | Action items | QA   |
|---------------------------|---------------|--------------|------|
| Plain decoding            | 0.1658        | 0.1672       | 0.46 |
| JSON constrained decoding | 0.1711        | 0.2674       | 0.7  |

Table 5: Performance comparison (ROUGE-2 for summarization and action items, F1-Score for QA) between causal language model decoding and JSON constrained decoding.

and structured output generation by providing the inference engine (SGLang (Zheng et al., 2024)) a JSON schema to guide the generation of output tokens. We choose three tasks that require a JSON formatted output and provide their Pydantic<sup>3</sup> based decoding hint as follows:

(i) **Combined summarization and action items:** This task prompts the model to generate a JSON object with two keys, “summary” and “action\_items”. The JSON decoding hint is: `RootModel[dict[str, Union[str, list[str]]]]`

(ii) **Question Answering (QA):** This task prompts the model to answer a list of questions about the transcript, each of which is represented by a key-value pair in the output JSON object. The JSON decoding hint is: `RootModel[dict[str, list[str]]]`

For this experiment, we only provide the JSON schema with minimal hints. From Table 5, we can see that even with a minimal hint of JSON-constrained decoding method, the model performance is greatly improved for all tasks.

## 5 Conclusion

This paper introduced DACIP-RC, a continual instruction pre-training method that enhances smaller LLMs’ adaptability to business conversational tasks. By leveraging our proposed reading comprehension-based instruction generation from transcripts approach, DACIP-RC significantly improves zero-shot generalization in both text classification and generation tasks. Moreover, the proposed DACIP-RC framework relies on a one-time manual creation of 41 meta-prompts that automates the generation of over 25 million training instances. Therefore, DACIP-RC offers a scalable approach to improve the performance of LLMs in real-world applications. In the future, we will explore the creation of new benchmarks to evaluate the DACIP-RC approach.

<sup>3</sup><https://docs.pydantic.dev/latest/>

## Limitations

The proposed reading comprehension-based instruction pre-training dataset construction approach is only evaluated on conversational data in the business communication domain. Thus, the findings from our experiments may not be applicable to other domains. Moreover, the analysis of the impact of various subsets of reading comprehension data on the downstream task performance is left out of the scope of this study. In the future, we plan to do ablation studies with various subsets of our datasets in search of a more optimal data mix for domain adaptation. Even though the proprietary dataset used in this paper cannot be released, our decision is also in full compliance with the EMNLP Industry Track policy. In addition, to ensure our research is transparent and replicable, we provided a detailed description of our DACIP-RC methodology throughout the paper so that other researchers can replicate our approach on their own conversational datasets.

## Ethics Statement

We adhered to the licensing terms of the various tool providers we used (e.g., Meta, HuggingFace). Since this study involves the use of proprietary data containing sensitive information, we applied a rigorous anonymization process to safeguard sensitive details to ensure the privacy and security of the internal data. In line with established privacy best practices (Narayanan and Shmatikov, 2007), we have decided not to release these datasets publicly to fully mitigate any risk of sensitive information leakage.

## References

- Asma Ben Abacha, Yassine M'rabet, Yuhao Zhang, Chaitanya Shivade, Curtis Langlotz, and Dina Demner-Fushman. 2021. Overview of the mediqua 2021 shared task on summarization in the medical domain. In *Proceedings of the 20th workshop on biomedical language processing*, pages 74–85.
- Reza Yazdani Aminabadi, Samyam Rajbhandari, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Olatunji Ruwase, Shaden Smith, Minjia Zhang, Jeff Rasley, et al. 2022. DeepSpeed-inference: enabling efficient inference of transformer models at unprecedented scale. In *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–15. IEEE.
- Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. [A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity](#).
- H.D. Brown. 2007. *Teaching by Principles: An Interactive Approach to Language Pedagogy*. Pearson Education.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Patricia L. Carrell and Joan G. Carson. 1997. [Extensive and intensive reading in an eap setting](#). *English for Specific Purposes*, 16(1):47–60. English for Academic Purposes.
- Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024a. [Instruction pre-training: Language models are supervised multitask learners](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550, Miami, Florida, USA. Association for Computational Linguistics.
- Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024b. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Xue-Yong Fu, Elena Khasanova, Md Tahmid Rahman Laskar, Harsh Saini, and Shashi Bhushan TN. 2025. Dacp: Domain-adaptive continual pre-training of large language models for phone conversation summarization. *arXiv preprint arXiv:2510.05858*.
- Xue-Yong Fu, Md Tahmid Rahman Laskar, Elena Khasanova, Cheng Chen, and Shashi Tn. 2024. [Tiny titans: Can smaller large language models punch above their weight in the real world for meeting summarization?](#) In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 387–394, Mexico City, Mexico. Association for Computational Linguistics.



- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.
- Joan Buchanan Hill. 2016. [Questioning techniques: A study of instructional practice](#). *Peabody Journal of Education*, 91(5):660–671.
- Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1416–1428.
- Israt Jahan, Md Tahmid Rahman Laskar, Chun Peng, and Jimmy Huang. 2025. Evaluating the effectiveness of cost-efficient large language models in benchmark biomedical tasks. *arXiv preprint arXiv:2507.14045*.
- Israt Jahan, Md Tahmid Rahman Laskar, Chun Peng, and Jimmy Xiangji Huang. 2024. A comprehensive evaluation of large language models on benchmark biomedical text processing tasks. *Computers in biology and medicine*, 171:108189.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. 2024. [Improving domain adaptation through extended-text reading comprehension](#).
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. 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)*, pages 2567–2577.
- Elena Khasanova, Pooja Hiranandani, Shayna Gardiner, Cheng Chen, Simon Corston-Oliver, and Xue-Yong Fu. 2022. [Developing a production system for Purpose of Call detection in business phone conversations](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 259–267, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.
- Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.
- Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Huang. 2023a. [A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 431–469, Toronto, Canada. Association for Computational Linguistics.
- Md Tahmid Rahman Laskar, Xue-Yong Fu, Cheng Chen, and Shashi Bhushan TN. 2023b. [Building real-world meeting summarization systems using large language models: A practical perspective](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 343–352, Singapore. Association for Computational Linguistics.
- Md Tahmid Rahman Laskar, Elena Khasanova, Xue-Yong Fu, Cheng Chen, and Shashi Bhushan Tn. 2024. [Query-OPT: Optimizing inference of large language models via multi-query instructions in meeting summarization](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1140–1151, Miami, Florida, US. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, et al. 2023. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.
- Arvind Narayanan and Vitaly Shmatikov. 2007. [How to break anonymity of the netflix prize dataset](#).
- OpenAI. 2023. [Gpt-4 technical report](#).

- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.
- Alec Radford. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Ashok Urala, Charaka Vinayak Kumar, Ajeet Kumar Singh, Bala Mallikarjunarao Garlapati, Srini-vasa Rao Chalamala, and Rahul Mishra. 2024. Lms with industrial lens: Deciphering the challenges and prospects—a survey. *arXiv preprint arXiv:2402.14558*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, et al. 2024. Efficient large language models: A survey. *Trans. Mach. Learn. Res.*
- Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhao Mo, Qiuha Lu, Wanjin Wang, Rui Li, Junjie Xu, Xianfeng Tang, et al. 2024. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. *arXiv e-prints*, pages arXiv–2411.
- Larry Wright. 2001. *Critical Thinking: An Introduction to Analytical Reading and Reasoning*. Oup Usa, Oxford, England and New York, NY, USA.
- Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. 2024a. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, page ocae045.
- Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhajan Kam-badur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.
- Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024b. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*.
- Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2023. Efficient continual pre-training for building domain specific large language models. *arXiv preprint arXiv:2311.08545*.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.
- Nathan Zhang, Anne Paling, Tania Habib Preston Thomas, Mahsa Azizi, Shayna Gardiner, Kevin Humphreys, and Frederic Mailhot. 2024. Data anonymization for privacy-preserving large language model fine-tuning on call transcripts. In *Workshop on Computational Approaches to Language Data Pseudonymization (CALD-pseudo)*, page 64.
- Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.
- Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. [Sglang: Efficient execution of structured language model programs](#).
- Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, et al. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921.

## A Appendix

### A.1 Meta Prompt Example

A metaprompt example for a wh- question in the scanning category.

```

=== Conversation Starts ===
{conversation transcript}
=== Conversation Ends ===

```

Based on the conversation above, generate a series of 5 open-ended questions that start with Who, When, Where, What, Which, How, Why. Provide answers to these questions. Questions 1-3 should have an answer as a complete sentence or several sentences. Questions 4-5 should have a short answer of just a few words. Write the results in the JSON format with two keys: question, answer.

Example output:

```

[
  "question": "When are Person 1 and Person 2 planning to meet?",
  "answer": "Wednesday at 2 PM",
  "question": "What time zone is Person 1 in?",
  "answer": "Person 1 is in the Pacific time."
]

```

Be creative. Do not include the example output in your response.

## A.2 Pre-training task example

An example of a pretraining task after parsing and postprocessing:

```

=== Transcript Start ===
{transcript}
=== Transcript End ===

```

Provide responses for the following questions: 1. Select all that apply to answer this question: What are some features discussed in the conversation? A. Installing a fan. B. Buying a TV. C. Building a recess for the fireplace. D. Adding a sound bar.

2. Which of the following is incorrect regarding the plan for the fireplace? A. There will be a TV above it. B. It will be part of a renovation. C. The fireplace is meant for outdoor use. D. The fireplace is meant to be recessed into the wall.

3. Select the correct option: What will be included in the follow-up email Person 1 is sending? A. Installation manual. B. Quote for the fireplace. C. Details about the fan. D. All of the above.

## A.3 Prompt Format for Downstream Tasks

### A.3.1 Summarization

Generate a summary of the {Length Type} following conversation {Format} without assessing its quality.

Note: length (long, medium, short) and format (e.g., bullet points) requirements in the prompt.

### A.3.2 Action Items

For the conversation given below, generate a newline-separated list of work, business, or service-related TODO tasks that should be completed after the conversation. Each task is a one-sentence summary of the action to be taken.

Transcript: {}

### A.3.3 Call Outcome

For the conversation below, apply the appropriate category from the list provided below to describe the outcome of the conversation (respond with 'Other' if no category applies): Call back, Unsuccessful contact, Voicemail Success, Payment / Billing, Status update, Scheduled appointment, Cancellation.

Transcript: {}

Note: this is not an exhaustive list of outcomes that are supported.

### A.3.4 Purpose of Call

For the conversation below, identify a single category for the purpose of the conversation chosen from this list: Account Management, Appointment, Billing Questions, Callback, Cancellation, Claim, Complaint.

Transcript: {}

Note: this is not an exhaustive list of supported categories.