Can probing classifiers reveal the learning by contact center large language models?: *No, it doesn't!*

Varun Nathan, Ayush Kumar and **Digvijay Ingle**
{varun.nathan, ayush, digvijay.ingle}@observe.ai
Observe.AI
Bangalore, India

Abstract

Fine-tuning large language models (LLMs) with domain-specific instruction dataset has emerged as an effective method to enhance their domain-specific understanding. Yet, there is limited work that examines the core characteristics acquired during this process. In this study, we benchmark the fundamental characteristics learned by contact-center (CC) domain specific instruction fine-tuned LLMs with out-of-the-box (OOB) LLMs via probing tasks encompassing conversational, channel, and automatic speech recognition (ASR) properties. We explore different LLM architectures (Flan-T5 and Llama) and sizes (3B, 7B, 11B, 13B). Our findings reveal remarkable effectiveness of CC-LLMs on the in-domain downstream tasks, with improvement in response acceptability by over 48% compared to OOB-LLMs. However, we observe that the performance of probing classifiers are relatively similar and does not reflect the performance of in-domain downstream tasks. A similar observation is also noted on SentEval dataset that assess capabilities of models in terms of surface, syntactic, and semantic information through probing tasks. Our study challenges the premise that probing classifiers can reveal the fundamental characteristics learned by large language models and is reflective of the downstream task performance, via a case-study of LLMs tuned for contact center domain.

1 Introduction and Related Works

Large Language models (LLMs) have made significant strides in recent years, with their ability to generate fluent text on variety of inputs (Wei et al., 2022; OpenAI, 2023). The strategy of fine-tuning the general-purpose models with domain-specific data has led to performance improvements in domains with LLMs such as BioGPT (Luo et al., 2022) and Med-PaLM (Singhal et al., 2023) in biomedical research, CodeT5 (Wang et al., 2021), CodeLLaMa in coding (Rozière et al., 2023), and Bloomberg-GPT (Wu et al., 2023) in finance, demonstrating the need and advantage of domain specific fine-tuning of LLMs. However, one domain that has received relatively little attention is the contact center industry. Contact centers play a crucial role in customer service and support for various businesses. They address a broad spectrum of customer queries, from technical issues to billing concerns. Incorporating LLMs into contact center workflows have a potential to transform the sector. However, noisy queries, spontaneous conversational dynamics and domain specific understanding pose significant challenges for LLMs. Adapting to these nuances is crucial for LLMs to enhance their effectiveness in contact center.

Instruction fine-tuning (Longpre et al., 2023) has emerged as one the promising approaches to develop domain-specific LLMs. Assessing effectiveness of LLMs often involves evaluating their performance on specific downstream tasks. However, probing the representations of the models on different probing tasks provide a deeper insight into the fundamental aspects of what language models capture and learn (Conneau et al., 2018). These tasks have been instrumental in understanding the underlying characteristic of language models. Conneau et al. (2018) introduced probing tasks in SentEval to assess sentence embedding representations of language models. Following this, studies like those by Tenney et al. (2019) and Lin et al. (2019) have applied layer-wise probing to BERT, shedding light on its semantic and hierarchical processing capabilities. While the majority of probing studies have concentrated on general LMs, work by Kumar et al. (2021) delved into the representation capabilities of RoBERTa in contact center domain. Building on this foundation, our study seeks to further understand the intricacies of instruction-fine-tuned LLMs in contact centers through specific research questions, aiming to uncover how these LLMs adapt and learn within this specialized context:

Proceedings of the Fifth Workshop on Insights from Negative Results in NLP, pages 92–100
June 20, 2024 ©2024 Association for Computational Linguistics

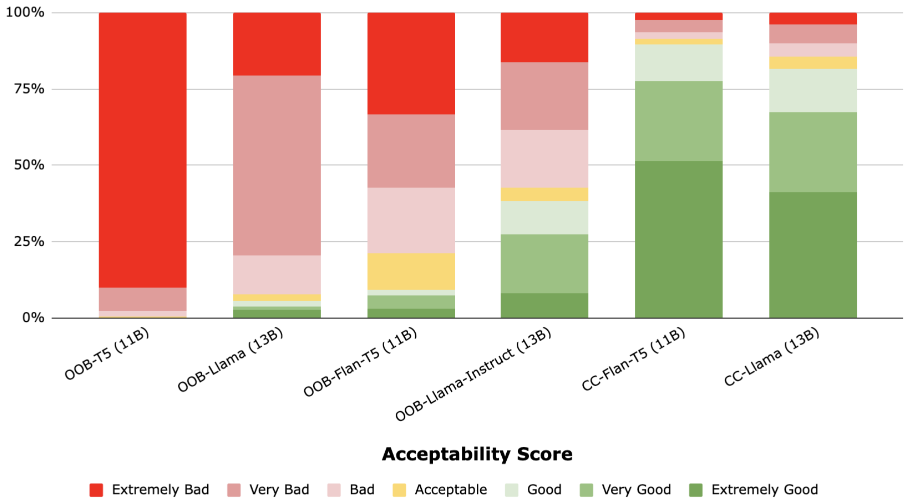

Figure 1: Quality of responses generated by CC LLMs versus OOB LLMs on downstream tasks in contact-center domain using a scale of *Extremely Bad* response to *Extremely Good* response. We note that CC LLMs result in over 48% improvement in response acceptability (>=*Acceptable*) compared to OOB LLMs (Flan-T5, Llama-Instruct).

- **RQ1:** How effective is instruction fine-tuning in enhancing LLMs' performance on downstream tasks within the contact-center domain?
- **RQ2:** What unique properties related to contact-center interactions are acquired by LLMs fine-tuned on CC instruction sets, compared to out-of-the-box models?
- **RQ3:** How does the choice of model architecture and size influence LLMs' performance on probing tasks?
- **RQ4:** Following domain-specific instruction fine-tuning, what general-purpose fundamental properties do LLMs retain?

2 Training Contact-Center LLM

In this work, we train a contact center-specific large language model (CC-LLM) using a proprietary dataset of ASR transcripts[1] from various sectors. Through instruction fine-tuning, we adapt out-of-box (OOB) LLMs to the contact center conversations, characterized by multi-party interactions, disfluencies, and ASR errors. Our training methodology involves generating diverse instructions for a wide array of tasks, such as call summarization, dialog question answering etc., to tailor the model's capabilities for contact center applications. More details is mentioned in Section A.1.

3 Probing tasks

Probing tasks tailored to the contact center domain provide valuable insights into the capabilities and limitations of LMs in this specific area, as demonstrated in a previous study (Kumar et al., 2021). In their work, the authors propose probing tasks to investigate the conversational, channel, and ASR properties of pre-trained LMs. We refer to these probing tasks and utilize the details outlined in the work to construct datasets to investigate the characteristics of contact-center LLMs via the probing tasks. Additionally, we also probe the LMs on a benchmark probing task of SentEval suite (Conneau et al., 2018) that aims to uncover the linguistic knowledge and underlying properties learned by the model. SentEval suite consists of probing tasks across the categories of surface information, syntactic information and semantic information.

4 Implementation Details

We compare two model classes, namely Flan (Longpre et al., 2023) and Llama (Touvron et al., 2023) in the three categories: OOB foundation model, OOB instruction model, and the CC instruction model. Following the previous work by Alain and Bengio, 2017, we utilize one-layer linear MLP classifier to train probing classifiers on the representations extracted from LLMs on the concatenated input of *{task-instruction, dialog/turn transcript}*. More details is outlined in Section A.2.

5 Results and Analysis

5.1 RQ1: Performance on downstream tasks

We perform a qualitative assessment of the responses generated by CC and OOB-LLMs by cate-

[1] We cannot release the dataset due to proprietary reasons.

	OOB Foundation				OOB Instruction-Tuned				Contact Center			
Probing Tasks	**OOB-T5 (3B)**	**OOB-T5 (11B)**	**OOB-Llama (7B)**	**OOB-Llama (13B)**	**OOB-Flan-T5 (3B)**	**OOB-Flan-T5 (11B)**	**OOB-Llama-Instruct (7B)**	**OOB-Llama-Instruct (13B)**	**CC-Flan-T5 (3B)**	**CC-Flan-T5 (11B)**	**CC-Llama-Instruct (7B)**	**CC-Llama-Instruct (11B)**
Disfluency	72.12	71.97	68.30	71.57	71.72	73.03	68.88	69.81	72.24	72.89	69.16	67.83
Pause	80.90	80.70	77.79	81.25	82.09	83.45	80.45	80.25	81.78	83.00	76.85	79.24
Overtalk	86.95	89.55	82.79	81.59	89.45	90.70	83.25	77.70	87.80	88.19	72.55	78.92
Question	77.52	74.49	70.34	74.31	77.59	75.39	73.03	74.33	76.96	80.37	76.22	77.15
Speaker	80.95	81.96	77.54	82.70	82.55	83.39	80.26	80.21	82.11	82.72	78.70	79.94
Resp. Length	67.65	69.35	66.23	69.09	69.20	69.66	66.03	67.29	68.88	68.79	67.27	67.95
Turn Taking	68.30	69.14	65.01	69.33	64.30	67.66	69.62	68.65	66.83	69.59	62.50	63.45
Token Multi	52.45	49.32	40.71	42.64	59.91	63.07	43.02	40.60	59.31	60.73	41.62	42.85
Token Binary	60.50	60.48	50.07	54.93	68.34	73.12	49.84	48.77	70.11	70.07	49.88	50.14
Avg. Score	71.93	71.88	66.53	69.71	73.90	75.50	68.26	67.51	74.00	75.15	66.08	67.50

Table 1: Benchmarking CC and OOB LLMs in terms of Macro F1 evaluated on contact-center probing tasks.

gorizing the responses generated by each of them into one among following seven classes: *Extremely Good*, *Very Good*, *Good*, *Acceptable*, *Bad*, *Very Bad*, and *Extremely Bad*. The annotation process in detail is mentioned in Section A.3. We analyze the responses generated by the LLM groups, and observe significant difference in the distribution of quality of responses (refer Figure 1). Specifically, responses generated by OOB-T5 (11B) (Raffel et al., 2020), OOB-Flan-T5 (11B), OOB-Llama (13B) and OOB-Llama-Instruct (13B) models are consistently skewed towards the lower end of the quality spectrum. A majority of these responses fall within the *Bad* to *Extremely Bad* categories, indicating that without specific fine-tuning, OOB models struggle to generate satisfactory responses for contact center specific instructions. Conversely, responses generated by CC-Flan-T5 (11B) and CC-Llama (13B) models exhibit a notable shift towards higher quality categories. A substantial portion of responses generated by these models lands in the *Acceptable* to *Extremely Good* range, demonstrating their ability to comprehend and generate contextually relevant responses for contact center interactions. Specifically, 91% of responses from CC-Flan-T5 and 87% of responses from CC-Llama has score >=*Acceptable* compared to 22% and 39% from respective OOB instruction models. This improvement in performance can be attributed to the fine-tuning process with contact center data.

5.2 RQ2: Contact-center probing tasks

In order to investigate the conversational properties learnt by CC-LLMs that lead to performance superior to OOB-LLMs, we evaluate these models on the probing tasks in Section 3 and per the methodology described in Section A.2. Although our probing tasks are carefully designed to uncover the latent knowledge within these models, our findings in Table 1 did not conclusively favor either type of LLM. Specifically, we observe a mixed trend where 1 out of 4 CC models, CC-Flan-T5 (3B) have higher average score and 2 out of 4 models, CC-Flan-T5 (11B) and CC-Llama (13B), have marginally lower ($< 0.5\%$) average score compared to their corresponding OOB instruction-tuned counterparts. We also note a similar observation when comparing CC-LLMs with OOB foundation models wherein 3 out of 4 CC-LLMs have comparable or better average score. This intriguing result prompts us to delve deeper into several critical aspects of LLMs and their fine-tuning process prompting us to put forth following opportunities for exploration. **Probing via Hidden Layer Representation:** While this method has been widely employed (Kumar et al., 2021; Fayyaz et al., 2021; Thukral et al., 2021) to unearth linguistic properties by language models, we question whether it is sufficiently nuanced to capture conversational intricacies. It is conceivable that the differences we seek are not embedded in the representations extracted but are instead contingent on the decoding strategy employed during the language generation process. This insight underscores the pivotal role of decoding strategies in converting latent embeddings into coherent sequences of tokens that reflect both the given instruction and input. It prompts us to consider that instructing and fine-tuning a general-purpose model and a domain-specific model may ultimately hinge on decoding proficiency rather than vastly divergent learned representations. We believe that this calls for a deeper investigation into designing right

	OOB Foundation		OOB Instruction Tuned		Contact Center	
Probing Tasks	**OOB-T5 (11B)**	**OOB-Llama (13B)**	**OOB-Flan-T5 (11B)**	**OOB-Llama-Instruct (13B)**	**CC-Flan-T5 (11B)**	**CC-Llama-Instruct (13B)**
Bigram Shift	92.48	85.66	94.19	85.59	92.17	76.79
Coordination Inversion	79.36	68.65	77.59	71.68	76.59	70.30
Object Number	82.70	73.90	89.20	74.20	86.90	76.49
Odd Man Out	73.69	66.09	74.99	66.90	72.99	63.51
Past Present	88.99	84.17	89.19	85.19	89.59	82.98
Sentence Length	100.00	100.00	100.00	100.00	100.00	100.00
Subj Number	86.19	79.49	92.09	79.66	90.29	81.57
Top Constituents	68.85	73.98	74.65	67.44	75.78	58.55
Tree Depth	36.02	28.73	37.24	32.65	38.49	27.92
Average Score	78.70	73.41	81.02	73.70	80.31	70.90

Table 2: Benchmarking CC and OOB LLMs in terms of Macro F1 evaluated on SentEval probing tasks.

probing strategies for recently popular generative language models trained via instruction fine-tuning.

Re-designing probing tasks: The existing set of probing tasks, although comprehensive, may not fully encapsulate the diverse landscape of conversational properties. Conversations are inherently dynamic, context-dependent, and influenced by various factors, including the interplay between participants, the history of the conversation, long-context dependencies and the evolution of topics. However the probing tasks in Kumar et al. (2021) are designed for single utterance inputs. Such scenario may not fully capture these dynamic aspects of conversation. It is plausible that more specific probing tasks tailored to the characteristic of contact center interactions are needed to fully conclude the learnings of the LLMs. These tasks should ideally mirror the challenges posed by real-world downstream applications that help diagnose the contextual properties and the interplay in the conversations.

5.3 RQ3: Model architecture and model size

From our results in Table 1, we note that T5 models consistently outperform Llama models across the three settings, OOB Foundation, OOB Instruction-tuned and Contact Center, highlighting that T5's encoder-decoder architecture has better learnt to comprehend conversational properties compared to Llama's decoder only architecture. Similarly, in downstream task performance (Section 5.1), CC-Flan-T5 (11B), although smaller in size, outperforms CC-Llama (13B). This outcome was surprising, especially considering Flan's smaller size and the Llama model's widespread popularity in the open-source community. It leads to question the impact of model architecture versus size in accurately comprehending the conversational contexts.

5.4 RQ4: General purpose probing tasks

Post fine-tuning on contact center instruction data, CC-Flan-T5 and CC-Llama show a reduced dependency on fundamental linguistic properties as evidenced by the decreased average score on SentEval probing suite. Consistent with prior findings, the Llama models exhibits a lower score compared to Flan models on general purpose probing task as well. Additionally, we note that while performance of CC-Flan-T5 is lower than OOB-Flan-T5 by 0.7%, this drop is 2.8% in Llama. This again suggests distinct learning mechanisms between encoder-decoder and decoder-only architectures, warranting further investigation in the community.

6 Conclusion

Our study contributes to the growing body of research on fine-tuning LLMs with domain-specific instructions. In this work, we demonstrate that CC-LLMs, CC-Flan-T5 and CC-Llama, exhibit superior performance on downstream tasks within the contact center domain. This finding reinforces the effectiveness of fine-tuning LLMs with domain-specific instructions, as expected. However, our comparison between OOB and CC models on the probing task reveals intriguing and unexpected observations. While the performance of CC-LLMs are much superior to the OOB-LLMs on downstream tasks, the performance of probing classifiers across the models shows no substantial differences. This questions the efficacy of traditional probing mechanisms and probing tasks in understanding the LLMs. We also observe that the decoder model (Llama-13B) consistently underperforms compared to the lower sized encoder-decoder model (Flan-11B) in all experiments This prompts more research into the learning dynamics of these architectures.

Limitations

While our study provides valuable insights into training a contact-center specific language model and conducting linear edge probing, it is important to acknowledge certain limitations in our work. Firstly, our exploration of language models is limited to a couple of models belonging to two architectures, one encoder-decoder and one decoder style. We choose these models on the basis of their effectiveness across different tasks as has been surfaced up in the research community, however, the trends we observe may not necessarily hold true for other models within the same class of architecture. Secondly, our work is based on the probing methodology of linear edge probing, which applies a one layer linear MLP on hidden representations. The performance and observations on probing tasks may differ if a different probing setup, such as an attention-based probing, is used. It is crucial to explore alternative probing methods to gain a more comprehensive understanding of the language model's characteristics. Moreover, the set of probing tasks we utilize may not cover the full range of characteristics that a language model can encode. Additional probing tasks can be considered to do a more extensive study of the model's capabilities. Lastly, our research is conducted on a proprietary dataset that cannot be released. This limits the ability of other researchers to directly compare their results or replicate our experiments. Access to the dataset is crucial for future work in this area, and we encourage the development of publicly available datasets for domain-specific language models.

Despite these limitations, our study underscores the importance of domain-specific instruction models and highlights the limited capacity of general-purpose language models to meet domain specific use-cases. Furthermore, we pose thought-provoking questions that can guide further research and contribute to the advancement of the research community's understanding of the properties encoded in generative language models in the new era.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Afra Amini and Massimiliano Ciaramita. 2023. Probing in context: Toward building robust classifiers via probing large language models. *CoRR*, abs/2305.14171.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2126–2136. Association for Computational Linguistics.

Mohsen Fayyaz, Ehsan Aghazadeh, Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. 2021. Not all models localize linguistic knowledge in the same place: A layer-wise probing on bertoids' representations. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2021, Punta Cana, Dominican Republic, November 11, 2021*, pages 375–388. Association for Computational Linguistics.

Ayush Kumar, Mukuntha Narayanan Sundararaman, and Jithendra Vepa. 2021. What BERT based language model learns in spoken transcripts: An empirical study. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2021, Punta Cana, Dominican Republic, November 11, 2021*, pages 322–336. Association for Computational Linguistics.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy V, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Moustafa-Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you! *CoRR*, abs/2305.06161.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside bert's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 241–253. Association for Computational Linguistics.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings Bioinform.*, 23(6).

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, pages 1–9.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics.

Shivin Thukral, Kunal Kukreja, and Christian Kavouras. 2021. Probing language models for understanding of temporal expressions. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2021, Punta Cana, Dominican Republic, November 11, 2021*, pages 396–406. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Yue Wang, Weishi Wang, Shafiq R. Joty, and Steven C. H. Hoi. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8696–8708. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David S. Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *CoRR*, abs/2303.17564.

A Appendix

A.1 Training Contact-Center Instruction-Tuned LLM

Numerous closed-source (Brown et al., 2020; OpenAI, 2023) and open-source (Touvron et al., 2023) general purpose LLMs have demonstrated abilities to address a diverse range of tasks in natural language processing. However, specialised models like CodeT5 (Wang et al., 2021), StarCoder (Li et al., 2023), Med-PaLM (Singhal et al., 2023), BioGPT (Luo et al., 2022), Galactica (Taylor et al., 2022), BloombergGPT (Wu et al., 2023) emphasize the significance of domain-specific models in achieving exceptional performance within fields like coding, bio-medicine, science, and finance. These models excel at producing high-quality outputs and tackling domain-specific challenges, illustrating the need of tailored LMs in diverse domains.

Inspired by the above works, we leverage inhouse dataset[2] of conversational interactions between agents and customers to train a CC-specific LLM (CC-LLM) to model the properties of CC conversations. Due to the spontaneous nature of these conversations, the data is often nuanced with characteristics such as multi-party speakers, disfluencies, overtalks, call transfers, etc. Furthermore, the data is obtained post transcription from an automatic speech recognition (ASR) system, thus introducing the challenge of dealing with ASR errors such as insertions, deletions, and substitutions, in turn establishing the need for a model robust to the conversational properties. In this work, we adopt an approach of instruction fine-tuning (Wei et al., 2022; Longpre et al., 2023), which is fine-tuning the language model on a mixture of tasks expressed via natural language instructions.

The process of fine-tuning a LM for contact-center applications involves three main components: a contact-center dataset, instructions specific to contact center use-cases, and a language model. To curate the contact-center dataset, we collect ASR transcripts of English conversations between agents and customers from various sectors, such as e-commerce, ed-tech, logistics, etc. We observe an average word-error-rate (WER) of 14.3 on these transcripts. The next step is to gather the instructions and their corresponding responses from the collected calls. We employ three processes to obtain these instructions:

- Initially, we utilize our previously annotated data from use-cases such as sentiment detection, intent classification, entity recognition, and question answering. We reformat this data into triplets containing an *{instruction, input, output}*. The instructions and outputs for these tasks are aggregated through a semi-automatic process involving human intervention. We leverage the human-in-the-loop approach to generate instructions and corresponding responses for the given task.
- Following this, we expand the instructions by employing a paraphrasing process. This allows us to generate multiple styles of the same instructions, thereby increasing the diversity of the instruction set.
- In addition to using the annotated data from the past, we also gather new sets of instructions by instructing human annotators to generate relevant questions that can be asked and answered during a call. Similar to the previous step, we expand these generated instructions using the paraphrasing process.

To assist the annotators in generating these tasks, we provide them with a list of insights that we aim to extract from the calls to address various use-cases. Examples of such insights include understanding and tracking customer and agent behaviors, following the steps taken in the call to resolve customer issues, and identifying different objections raised by the customers. The overall corpus is constructed with a diversity of full call transcripts, segmented call transcripts and individual speaker turns. On an average each task-instruction is paraphrased into 50 alternate instruction to make the model generalizable to unseen variations.

Here are some important statistics on the internally curated contact-center dataset:

- Total corpus size: 110030
- Number of tasks: 59
- Number of instructions: 2468

Some example tasks considered in the dataset include *reason for call*, *call summarization*, *segmented call summarization*, *confirmed next steps*,

[2] We cannot release the dataset due to proprietary reasons.

Task	Task Instruction
Call Reason	What is the primary call intent
Call Summarization	Summarize the dialog
Segmented Call Summarization	Summarize a segmented portion of the dialog
Confirmed Next Steps	List the confirmed next steps if any in the dialog
Question-Answering (QA)	Answer the question based on context present in the dialog
Entity Extraction	List the entities present in the dialog
Topic Segmentation	Segment the dialog into coherent topics
Text Rewriting (QA)	Rewrite a given piece of text in a fluent and grammatically correct form
Sentiment Classification	Classify the sentiment of the customer in the call among positive, negative and neutral.

Table 3: Definitions of representative tasks considered in the internally curated contact-center dataset. These tasks were utilized as the downstream tasks for RQ1 (Section 5.1).

Question-Answering (QA), *entity extraction*, *topic segmentation*, *text rewriting*, *sentiment classification*. Refer to Table 3 for instructions used for these tasks.

Further, we fine-tune OOB-LLMs that are free for commercial use on the curated dataset. Specifically, we obtain CC-Flan-T5 model by fine-tuning the corresponding sized OOB-Flan-T5 model, and obtain CC-Llama model by fine-tuning the corresponding sized OOB-Llama-Instruct model. The models were trained on 8×A100 40GB GPUs (p4d.24x larger) using Deepspeed [3] library. The models were fine-tuned for a total of 2 epochs. The training time per epoch for Flan-T5 (11B) model is 32 hours, while it take 17 hours to train Llama (13B) for each epoch.

A.2 Implementation Details for Probing Setup (RQ2, RQ3, RQ4)

In this section, we provide a detailed account of the implementation specifics related to our investigation into LLMs fine-tuned on CC instructions.

- **Representation Extraction**: To initiate the process, we extract representations from the LLMs, harnessing their hidden states to encapsulate the contextual nuances present in the transcripts as well as instructions which are indicative of the tasks they are expected to perform as demonstrated in a previous study (Amini and Ciaramita, 2023). Our approach is different from the authors in the sense that we use a linear probe as opposed to an attentional probe which is explained in more detail later in this section. For encoder-decoder models, we tap into the final encoder layer to obtain representations for each token within the input prompt. We adopt a suitable aggregation method depending on the characteristics of the specific probing task. For single-token probing tasks, we use the representation of the target token. For other tasks, we obtain an average of representations of all input tokens. On the other hand, in decoder-only models, we utilize the last hidden layer of the decoder block. The aggregation approach for decoder-only models aligns with encoder-decoder models for single-token probing tasks but relies on the last token's representation for other tasks. This difference stems from encoder-decoder models being bidirectional, making each token representation contextual to the entire sequence. In contrast, decoder models process tokens sequentially from left to right, making each token's representation contextual only to the tokens before it. Therefore, we consider the last token's representation as it encompasses information from entire sequence.

  For encoder-decoder models, the embedding dimension spans 512, 1024, 2048, and 4096 tokens, while for decoder-only models, it encompasses 32001 and 65024 tokens. The different embedding dimensions for the two classes of models stems from the difference in model architectures and context lengths employed during pre-training and fine-tuning. We employed a context length of 512 for all models when extracting representations due to the input prompts having a maximum sequence length of 507 tokens across probing tasks. All models receive an input consisting of a prompt, which is a combination of transcript generated from the input dialog, and an instruction that defines the probing task being conducted.

- **Hyperparameters**: Post representation extraction, we employ a Multilayer Perceptron

[3] https://github.com/microsoft/DeepSpeed

(MLP) comprising a single hidden layer, utilizing the extracted representations as feature inputs for probing. We adopt a *sigmoid* and *softmax* activation function for binary and multi-class classification respectively. We perform a hyper-parameter sweep over the range - number of neurons in the hidden layer $\in \{50, 100, 150, 200\}$, learning rate $\in \{1e-3, 1e-2, 5e-2\}$, batch size $\in \{4, 8, 16, 32, 64\}$ and choose the best setting as evaluated on eval set. Additionally, we employ Adam optimizer with a dropout rate of 0.3, incorporate a weight decay of 0.00001, and set the maximum number of epochs to 20. Moreover, all experiments include early stopping and check-pointing for the best model.

- **Compute Infra**: Our experiments comprising representation extraction and probe classifier training were conducted on an AWS cloud instance, specifically, the p4d.24xlarge instance, equipped with eight GPUs, each boasting 40 GB of memory. The process of extracting representations is computationally intensive, chiefly because of the substantial embedding dimensionality. On average, a single run of the representation extraction job for decoder-only models of size 13 billion parameters demands 8-10 hours for completion, whereas the corresponding timeframe for encoder-decoder models of size 11 billion parameters is considerably shorter, ranging from 1-2 hours. In contrast, training of probing classifiers present a lighter computational load and general taking around 0.5 hours for each classifier.

- **Sample instructions used for contact-center probing tasks**
  - Disfluency Detection: Is the given spoken utterance disfluent?
  - Pause Classification: Does the speaker take long pauses while speaking?
  - Overtalk Detection: Are two speakers talking over each other?
  - Question Classification: Did the speaker ask any question?
  - Speaker Role: Who among the agent or customer is the speaker for a given utterance?
  - Response Length: Is the expected response to current utterance is short or long?
  - Turn Taking: Has speaker completed its turn?
  - Token Multi: What is the error category of word *{ref_word}* among insertion error, substitution error or no error?
  - Token Binary: Is the word *{ref_word}* correct word in the given input

  As mentioned in the previous section, these instructions are concatenated with the input (dialog or turn transcript) to obtain the representations for training the probing classifiers.

A.3 Annotation process for evaluating model responses on contact center specific downstream tasks in RQ1

In the execution of this study, an annotation protocol was established, aimed at quantifying the quality of the response on the parameters of consistency, relevance, and fluency of responses generated by the large language models. Annotation guidelines were crafted, incorporating examples to illustrate the application of quality metrics, ensuring uniformity in annotator interpretation and application of these criteria.

To prepare for this task, 7 in-house annotators were subjected to a two-week training, designed to familiarize them with the nuances of instruction following large language models and interpretation of the response quality against the input of a call transcript and an instruction. This training utilized a dataset distinct from the evaluation corpus to prevent overlap and bias. Throughout the annotation process, the origins of the model outputs were anonymized to preclude annotator bias towards any specific model. Annotation agreement was monitored and evaluated through a cross-annotator review mechanism, yielding a Fleiss' Kappa score of 0.59. This score signifies moderate inter-annotator agreement, validating the reliability of the annotation process post-training.

Upon completion of the training week, the evaluation corpus was allocated among the annotators, where each annotator had to go through all data points across all models. The final response quality was judged on the basis of majority vote of the labels provided by the annotators.