Recognizing and Predicting Business Communication Outcomes Using Local LLMs

Wenbo Wang
EECS
University of Missouri
Columbia, MO, USA
wwr34@mail.missouri.edu

Can Li
EECS
University of Missouri
Columbia, MO, USA
lican@mail.missouri.edu

Lingshu Hu
EECS
University of Missouri
Columbia, MO, USA
lhu@mail.missouri.edu

Bin Pang
EECS
University of Missouri
Columbia, MO, USA
bpnrc@mail.missouri.edu

Bitty Balducci
Carson College of Business
Washington State University
Pullman, WA, USA
bitty.balducci@wsu.edu

Detelina Marinova
Robert J. Trulaske, Sr. College of Business, University of Missouri
Columbia, MO, USA
marinovad@missouri.edu

Matthew Gordon
Department of English
University of Missouri
Columbia, MO, USA
gordonmj@missouri.edu

Yi Shang
EECS
University of Missouri
Columbia, MO, USA
shangy@missouri.edu

***Abstract*—In this paper, we use machine learning methods based on three popular open-source large language models (LLMs) that can run efficiently on local computers to recognize and predict business communication outcomes. The methods include zero-shot Alpaca-Lora-7B, BigBird with fine-tuning, Alpaca-Lora-7B with prompting and LoRA-based fine-tuning, and Llama2-70B-Chat with one-stage and two-stage prompting. Our experimental results on a real-world dataset showed promising results of LLMs for both communication outcome recognition and prediction tasks. On recognizing communication outcomes, Alpaca-Lora-7B with prompt engineering and LoRa-based fine-tuning performed the best achieving 94.66% accuracy while BigBird with fine-tuning is closely behind with 94.27% accuracy, which, in turn, outperformed prompt engineering on LLMs. On predicting communication outcomes based on partial conversations, BigBird fine-tuned using the beginning 70% of each conversations achieved more than 85% prediction accuracy based on the first 70% of each conversation.**

***Keywords*—machine learning, local LLMs, conversation outcome recognition, conversation outcome prediction, business calls.**

I. Introduction

In the realm of business, a cold call traditionally denotes an unsolicited outreach to potential customers over the phone with the aim of promoting goods or services. While such endeavors are often labor-intensive and perceived as intrusive by recipients, they remain indispensable for many businesses and constitute a significant portion of their customer acquisition. Therefore, enhancing the efficiency of this process offers advantages for firms that span diverse industries.

In recent years, machine learning (ML) methods, in particular deep learning models and large language models (LLMs), have made significant breakthroughs in natural language processing (NLP). A seminal deep learning architecture, called transformers [1], was very successful in many NLP tasks, like sentiment analysis [2], formality detection [3], and text augmentation [4]. Further, LLMs and generative AI possess significant capability in language understanding tasks, such as misinformation detection [5][6], politeness prediction [7], and anomaly detection [8]. Harnessing the power of AI and LLMs to recognize and predict the outcomes of business conversations is a transformative approach in understanding dyadic interactions. For example, cold call interactions involve a myriad of factors that influence their success or failure including the nuances of language, sentiment, and contextual cues. Such factors make cold call interactions inherently complex to analyze and predict, yet deep learning models, equipped with the ability to understand and interpret intricate patterns, offer a promising solution to this age-old challenge.

Even though the advent of LLMs has sparked interest among businesses in leveraging AI methods to analyze, automate and enhance customer service and acquisition procedures, there are concerns about data security and privacy. The use of cloud-based AI tools could result in leakage of proprietary data to competitors. Therefore, many companies have implemented stringent regulations that dictate where data can be stored and processed. Fortunately, there are many open-source LLMs that can run on local computers to process proprietary data. However, firms must tailor these models for specific applications through fine-tuning or prompt engineering to fully realize the benefits of these tools. To investigate the feasibility and performance of local LLMs in recognizing and predicting business call outcomes, we applied machine learning methods based on three popular open-source local LLMs (BigBird, Alpaca-Lora-7B, Llama2-70B-Chat) to transcriptions of real-world business cold calls. Specifically, the methods include zero-shot Alpaca-Lora-7B, BigBird with fine-tuning, Alpaca-Lora-7B with prompting and LoRA-based fine-tuning, and Llama2-70B-Chat with one-stage and two-stage prompting. Our experimental results showed promising results of LLMs for both communication outcome recognition and prediction tasks. Specially, the main contributions of this paper include: 1) Applied new local LLMs with various prompting engineering and fine-tuning techniques to recognize outcomes of business calls and achieved very high recognition accuracy, over 94%, on a real-world dataset. 2) Applied a top-performing model, BigBird with fine-tuning, to predict communication outcomes based on partial conversations and achieved high prediction accuracy. For example, BigBird trained using the first 70% portion of conversations achieved more than 85% prediction accuracy based on the first 70% of a conversation.

2835-5776/24/$31.00 ©2024 IEEE
DOI 10.1109/IRI62200.2024.00042

II. Related Work

A. Business Conversation Outcome Recognition and Prediction

There are many prior studies that applied machine learning methods to understand and predict outcomes in business calls. Gopagoni et al. [9] applied various traditional classification algorithms, such as logistic regression, SVM, and decision tree, to understand the relationship between calls and customer profiles and predicted sales call success rate. They found logistic regression outperformed other classifiers and could effectively explain the data set. Feng et al. [10] proposed a machine learning method to predict the success of bank telemarketing sales of time deposits. They used both prediction accuracy and average profit in a dynamic ensemble selection method using meta-training. Vo et al. [11] developed a customer churn prediction model based on the contents of phone communications in a call-center context with millions of conversations. Their model accurately predicted the risk of client churning and generated meaningful insights using customers' personality traits and attributes. Peng et al. [12] proposed a deep learning model to predict conversation outcomes using audio features extracted from speaker turns between salesperson and customers. Some previous work used multimodality for conversation outcome prediction. Qin et al. [13] proposed a bidirectional long short-term memory (BiLSTM) network to predict financial risk based on conference calls by using both vocal and verbal features. Li et al. [14] employed a multimodal transformer network that used both vocal and verbal cues to predict conversation outcomes in business calls. Their result showed that verbal features are more important than vocal features. The main difference of our work from the previous work is the application of new local LLMs and prompt engineering and fine-tuning techniques to recognize and predict outcomes of real-world business calls, while achieving high recognition and prediction accuracy.

B. LLMs and ML Methods Used in This Paper

BigBird [15] is a leading deep learning model from Google, an improved version of BERT [16] that can process significantly longer sequences. It uses sparse attention, global attention and random attention, which approximates full attention and is computationally efficient in handling extended sequences. BigBird outperformed previous models, such as BERT or RoBERTa, in various long document NLP tasks, such as question answering and summarization [17].

LoRA, an acronym for Low-Rank Adaptation (of Large Language Models), is a machine learning technique that involves freezing the weights of pre-trained models and introducing trainable rank decomposition matrices into every layer of the transformer architecture [18]. This approach significantly reduces the number of trainable parameters for downstream tasks.

Llama 2, a family of generative text models released by Meta's GenAI, was trained on a diverse dataset from publicly accessible sources. Spanning a parameter range from 7 billion to 70 billion, these models employ an optimized auto-regressive transformer architecture with additional grouped-query attention [19]. Llama2-70B-Chat is the largest chat-optimized versions that used supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to align with human preferences.

Alpaca is a fined-tuned version of Llama model. It was fine-tuned based on 52K instruction-following demonstrations generated in the style of self-instruct using the text-davinci-003 dataset [20]. On the self-instruct evaluation set, Alpaca performed similarly to OpenAI's text-davinci-003 model, yet was more compact and cost-effective. Despite Alpaca's success, the model was not publicly available. Subsequently, researchers adopted the same approach to create the Alpaca-LoRA model by incorporating the LoRA technique to enhance training efficiency.

III. Problem Formulation and Proposed Methods

This section begins with a problem formulation followed by our proposed machine learning methods and LLMs.

A. Problem Formulation

We define the conversation outcome *recognition* problem as follows. A salesperson makes an unsolicited call to a business customer in the middle of their workday with the goal of arranging an in-person meeting. Therefore, each call has either a positive or negative outcome. Positive calls consist of those in which the customer is interested, follow-up steps are discussed, or a meeting is scheduled. Negative calls consist of those in which the customer is not interested, and no meeting is scheduled. More specifically, in this paper, a conversation is labeled positive if it ends with one of the following results: 1) A meeting is scheduled. 2) The customer asks the salesperson to call them back later, especially when the customer seems interested in the salesperson's offer but needs to have someone else present before making a decision. The call-back timeframe could be long or short, anywhere from the same day to a year or two in the future or might not be mention at all. 3) The customer asks the salesperson to send information without agreeing to a meeting. They could ask for an email, or for the salesperson to drop off information at their business. 4) The customer cannot commit to a meeting date/time at that moment but intends to set a meeting when they can. 5) The customer drives the conversation, either re-directing the course of the conversation in some way or requesting contact information so that they can be the one to follow up if they so choose and set the meeting parameters after the call ends. 6) The customer states that they have already met with their representative recently so there is no need for a follow-up. Similarly, we define the conversation outcome *prediction* problem as follows. At a certain point of a business call, predict the outcome of the call using the transcript of the conversation from the beginning of the call up to that point, i.e., use partial conversation to predict the outcome, as either positive or negative.

In this paper, we focus on LLMs that can be run locally to protect the privacy and confidentiality of the business data. Therefore, we do not consider cloud-based AI services, such as OpenAI's ChatGPT.

B. Prompt Engineering Methods

When working with LLMs, prompt engineering is the practice of designing effective prompts (instruction inputs) for LLMs to produce high-quality outputs. Unfortunately, other than a few general guidelines, there is no fixed process one can follow to guarantee high-quality prompts.

In this work, we started with a simple prompt to ask LLMs to recognize or predict the outcome of a call: *"Is the customer positive or negative towards the conversation?"* A key advantage of LLMs is their ability to provide the reasoning behind its prediction, which is very useful to correct errors and generate better answers. Thus, the second prompt we tried was: *"Is the customer positive or negative towards the conversation, why?"* The results of LLMs in response to this prompt revealed 2 issues: a) LLMs sometimes only used the beginning of a conversation to make a prediction, which could be wrong when a conversation was positive in the beginning but has since turned negative at the end, or the beginning of the conversation was unrelated content such as a discussion about the weather. b) LLMs sometimes misunderstood the meaning of parts of the conversation. For example, a customer said "I appreciate it" simply to be polite, but LLMs interpreted it as positive. To address these two issues, we tried the 3rd prompt to be more specific and detailed: *"Is customer's final attitude towards the conversation positive or negative? What did representative say make customer feel this way?"* To further improve outcome recognition and prediction accuracy, we explicitly told LLMs what to ignore and added words such as "greetings" "farewells" "friendliness" "gratitude" and "general niceties" on the list of textual cues to ignore. In addition, based on the outcome criteria as defined in the "Problem Formulation" section, we created detailed prompts in if-then rule style: *"If a meeting is arranged, then customer's attitude is positive. If a meeting is not arranged, but customer is willing to be in touch with representative later, or customer is open to more information, or they already had a meeting, then customer's attitude is positive, otherwise customer's attitude is negative."* Finally, we tested different variations of these prompts and picked those that yielded the best results.

C. BigBird with Fine-Tuning

BERT from Google, or "Bidirectional Encoder Representations from Transformers," excelled in tasks that require contextual understanding of entire sequences. A limitation of BERT is that it can only take input sequences up to 512 tokens in length, which is significantly shorter than many business conversations in the real world. BigBird from Google improved over BERT and can handle much longer sequences. BigBird reduces time complexity from $O(n^2)$ in BERT to $O(n)$ and increases the maximum input token length from 512 to 4096 [15], which can accommodate the full length of cold call transcriptions in our real-world dataset. In the experiments, we fine-tuned BigBird using the training set of our dataset.

D. Llama2-70B-Chat with Prompt Engineering

Given the impracticality of fine-tuning Llama2-70B-Chat on consumer-grade hardware, we used prompt engineering to enhance its performance, which involved providing appropriate instruction text to facilitate the model's comprehension of the tasks. For our problem, a single prompt to Llama2-70B-Chat might not be enough for LLMs to perform well. It sometimes generated "neutral" or other irrelevant output despite explicit instructions to answer either positive or negative. We proposed a two-stage prompting method to achieve better performance.

In the two-stage prompting method, we used two prompts in sequence. First, we used a prompt to query Llama2-70B-Chat and if its response was neither positive nor negative, then we used another prompt to query Llama2-70B-Chat again. We tried two variants: a) a simple prompt followed by a more descriptive prompt, and b) a more descriptive prompt followed by a simple prompt. The idea of the first variant is to use a simple prompt to get answers on easy cases and leave the hard case to a more descriptive prompt. The idea of the second variant is to use a descriptive prompt to get more accurate answer when possible, and when Llama2-70B-Chat failed to give an answer on the descriptive prompt, a simple prompt was used as a backup.

E. Alpaca-Lora-7B with prompt engineering and LoRA-based fine-tuning

LoRA, an acronym for Low-Rank Adaptation of LLMs, is a machine learning method that freezes a pre-trained model's weights and injects trainable rank decomposition matrices into each layer of its transformer architecture [18]. This method greatly reduced the number of trainable parameters and allowed us to fine-tune the Alpaca-Lora-7B model, which was an order of magnitude larger than the BigBird model, on consumer-grade hardware. We also used prompt engineering techniques to improve recognition accuracy.

IV. Experimental Results

This section begins with a description of the real-world dataset we created and used in our experiments, followed by experimental results of the proposed methods.

A. Dataset Preparation

We obtained a proprietary dataset of customer calls from a company under a non-disclosure agreement. The dataset consists of 1,310 cold-call conversation audio recordings between salespeople and business customers. The audio recordings were then transcribed into text. Based on the outcome of each conversation, trained labelers coded each conversation as either positive or negative. We then randomly split the dataset into 80% of the conversations for training and 20% for testing. Therefore, our training set contained 1048 conversations, out of which 600 or 57% were labeled positive and 448 or 43% negative. Our test set contained 262 conversations, out of which 144 or 55% were labeled positive and 118 or 45% negative.

B. Outcome Recognition Performance of Proposed Methods

We applied 5 methods based on 3 models (BigBird, Llama2-70B-chat, and Alpaca-Lora-7B) to the conversation outcome recognition task in this experiment.

1) **Zero-shot Alpaca-Lora-7B**. As the baseline we queried the Alpaca-Lora-7B model, one of the popular local LLMs that the general public uses, directly on the test set. For each conversation in the test set, we asked this prompt: *"Is customer's final attitude towards the conversation positive or negative?"*

2) **BigBird with fine-tuning**. We fine-tuned the BigBird Base model using our training set with the default parameters as following: max encoder length 4096, num_attention_heads 12, num_hidden_layers 12, hidden_size 768, intermediate_size 3072, block_size 64, train_batch_size 2, AdamWeightDecay optimizer with initial learning rate 1e-5, and training steps 16,000.

3) **Llama2-70B-Chat with various prompts**. We tested different prompts ranging from simple and short to long and detailed, as described in the "Prompt Engineering Method" section. Overall, the detailed ones performed better. The best prompt is "*If a meeting is arranged, customer's attitude is positive. If a meeting is not arranged, but customer is willing to be in touch with representative later, or customer is open to more information, or they already had a meeting, customer's attitude is positive, otherwise customer's attitude is negative. Ignore greetings, farewells, friendliness, gratitudes, and general niceties, is customer's final attitude towards the conversation positive or negative? What did representative say make customer feel this way?"* which led to 85.11% acccuracy on the test set. Sometimes Llama2-70B-Chat failed to recognize the correct outcome of a conversation. The following is a false positive example where Llama2-70B-Chat answered positive on a negative conversation. *"Positive. The conversation ends with the representative saying, "Take care" and the customer responding with "Okay, bye". This indicates a positive ending to the conversation and suggests that the customer's final attitude towards it is also positive."* The following is a false negative example where Llama2-70B-Chat answered negative on a positive conversation. *"The customer's attitude is negative. They did not schedule a meeting with the representative and stated they were busy. The representative did not do anything wrong to make the customer feel negatively. The customer seemed very busy and preoccupied with other things."*

4) **Llama2-70B-Chat with 2-stage prompting.** In this method, we used two prompts in sequence. We tested various combinations of simple to descriptive prompts. Table I shows results from using the following three prompts, ranging from simple to more descriptive, as representative results of this method. Simple prompt Q1:*"Is customer's final attitude towards the conversation positive or negative, why?"* Medium prompt Q2:*"If a meeting is arranged, customer's attitude is positive. If a meeting is not arranged, but customer is willing to be in touch with representative later, or customer is open to more information, or they already had a meeting, customer's attitude is positive, otherwise customer's attitude is negative. Is customer's final attitude towards the conversation positive or negative? What did representative say make customer feel this way?"* Most descriptive Q3:*"If a meeting is arranged, customer's attitude is positive. If a meeting is not arranged, but customer is willing to be in touch with representative later, or customer is open to more information, or they already had a meeting, customer's attitude is positive, otherwise customer's attitude is negative. Ignore greetings, farewells, friendliness, gratitudes, and general niceties, is customer's final attitude towards the conversation positive or negative? What did representative say make customer feel this way?"*

TABLE I. EXPERIMENTAL RESULTS OF LLAMA2-70B-CHAT WITH DIFFERENT PROMPT COMBINATIONS IN THE 2-STAGE PROMPTING METHOD ON THE TEST SET.

Stage 1	Stage 2	TP	TN	FN	FP	Error	Test Accuracy
Q3	Q2	130	96	12	22	2	**86.26%**
Q3	Q1	130	95	12	23	2	85.88%
Q1	Q2	135	84	9	33	1	83.59%
Q1	Q3	128	88	16	28	2	82.44%
Q2	Q1	139	74	3	45	1	81.30%
Q2	Q3	139	74	3	44	2	81.30%

TP: number of true positive conversations; TN: number of true negative conversations; FN: number of false negative conversations; FP: number of false positive conversations; Error: number of conversations where the model did not provide a valid answer because it was out of memory, generated invalid output or answered "neutral".

Table I shows our experimental results of Llama2-70B-Chat with different prompt combinations using the two-stage prompting method on the test set. The result shows that using the most descriptive prompt Q3 first yielded the best results. Using Q3 followed by the medium prompt Q2 produced the best accuracy results, 86.26%, which was slightly better than Q3+Q1. Interestingly, using the simple prompt Q1 first generated better performance compared to using the medium prompt Q2 first. Q1+Q2 was slightly better than Q1+Q3, because Q3 was more likely to fail than Q2 in the cases when Q1 failed.

5) **Alpaca-Lora-7B with prompt engineering and LoRA-based fine-tuning**. We used engineered prompt *"If a meeting is arranged, customer's attitude is positive. If a meeting is not arranged, but customer is willing to be in touch with representative later, or customer is open to more information, or they already had a meeting, customer's attitude is positive, otherwise customer's attitude is negative. Is customer's final attitude towards the conversation positive or negative?"* to fine-tune Alpaca-Lora-7B model for 96 epochs (adding more epochs does not yield statistically significant improvement) with default parameters as following: batch size 128, micro batch size 4, num epochs, learning rate 3e-4, LoRA rank 8, LoRA alpha (the higher alpha value, the higher LoRA layers impact the base model) 16, LoRA dropout 0.05. The same prompt was also used for inference.

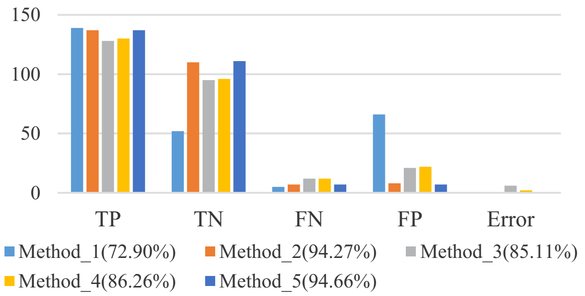

Fig. 1. Outcome recognition accuracy of the five proposed methods and their outputs on the conversations in the test set in terms of true positive (TP), true negative (TN), false negative (FN), false positive (FP), and errors. The number in the parenthesis after each method is the outcome recognition accuracy calculated as (TP+TN)/total number of conversations.

Finally, in Fig. 1, we compare the test results of our 5 proposed methods using the real-world dataset. The figure shows outcome recognition accuracies of the five methods and their output on the conversations in the test set in terms of true positive (TP), true negative (TN), false negative (FN), false positive (FP), and errors.

Method 2 (BigBird with fine-tuning) and Method 5 (Alpaca-Lora-7B with prompt engineering and LoRA-based fine-tuning) performed the best, achieving accuracy above 94%. Method 3 (Llama2-70B-Chat with prompting) and Method 4 (Llama2-70B-Chat with 2-stage prompting) were comparable, at around 85% accuracy, showing that 2-stage prompting was not much better than one-stage prompting. Method 1 (Zero-shot Alpaca-Lora-7B) was the worst, at 73% accuracy, due to being a smaller LLM and simple prompt. In summary, fine-tuned models performed much better than models with prompt engineering, which, in turn, outperformed zero-shot LLMs.

C. Predicting Conversation Outcome Based on Partial Conversations

In business conversations, especially cold calls, if an unfavorable outcome has been predicted early, the sales representative could be advised to end the conversation and move on to the next customer to maximize sales efficiency and reduce customer's irritation. To achieve this, we evaluated conversation outcome predictions based on partial conversations using a top performing LLM from our prior experiments: BigBird with fine-tuning. We used the same training and test set as in the outcome recognition experiments. For each conversation in the training and test set, we extracted the first 10%, 20%, up to 100% of the conversation. Then, we trained BigBird models using the extracted partial conversations. Next, the trained models were applied to the partial conversations in the test set to predict the outcome of a conversation based on the first 5%, 10%, 15%, up to 100% of the conversation, respectively. The accuracy of test performance is (TP+TN)/(TP+TN +FP+FN).

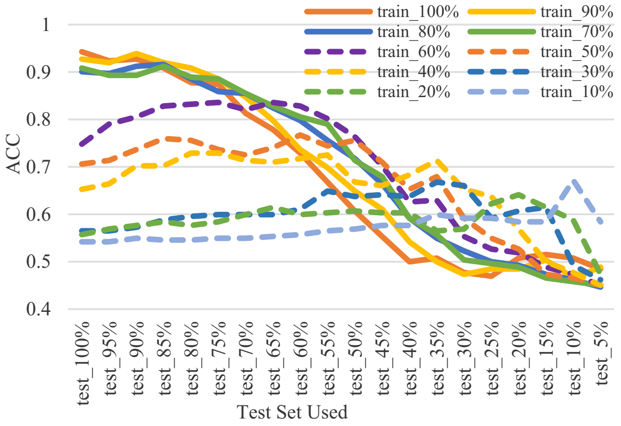

Fig. 2. Prediction accuracies of BigBird models fine-tuned with different partial conversations serving as training and testing inputs.

Fig. 2 shows the outcome prediction accuracies of BigBird models fine-tuned using different partial conversations on different partial test conversations. For example, the curve *"train_80%"* represents the accuracy of the BigBird model trained using the first 80% of each conversation in the training set ran on different ratios of partial conversations in the test set. This curve's accuracy value at "*test_70%"* is 0.85, meaning that this model achieved 85% accuracy when using the first 70% of each conversation in the test set. Note that in test set, 55% conversations were positive and 45% negative. Thus, prediction accuracies below 55% were worse than simply predicting all of them as positive.

The results show that BigBird models trained using larger partial conversations (70% or more) performed much better than the ones trained using smaller partial conversations (60% or less) on larger partial conversations (70% or more) in the test set. For example, BigBird trained using the first 70% of each conversation in the training set achieved 85% outcome prediction accuracy just using the first 70% of each conversation in the test set. It achieved more than 90% prediction accuracy when using the first 85% of each conversation in the test set. In comparison, BigBird models trained using 30% or less partial conversations achieved less than 60% prediction accuracy on 70% partial conversations. On the other hand, BigBird trained using smaller partial conversations (e.g., 30% or less) performed better than other models trained using larger partial conversations (e.g., 70% or more) when making predictions based on small partial conversations (30% or less), achieving prediction accuracy around 60%. Overall, BigBird models trained using larger partial conversations (80% or more) achieved very high accuracy on larger partial conversations (80% or more).

To improve overall robustness, more BigBird models are trained using a mixture of partial conversations. We created 5 mixed training sets by combining different partial conversations, as shown in Table II. For example, in training set 1, each conversation in the training set produced 10 partial conversations, corresponding to the first 10% of the conversation, the first 20% of the conversation, and so on. These partial conversations were all used to train a BigBird model. In total, 5 BigBird models were trained, each using one of the 5 training sets. Then, the 5 trained models were used to predict the outcomes of conversations using the first 5%, 10%, 15%, ..., up to 100% of each conversation in the test set. The prediction accuracy results are shown in Fig. 3.

TABLE II. FIVE DIFFERENT TRAINING SETS FOR TRAINING BIGBIRD MODELS.

Training set	Increment	% of partial conversations included
1	10%	100%, 90%, 80%, …, 20%, 10%
2	20%	100%, 80%, 60%, 40%, 20%
3	30%	100%, 70%, 40%, 10%
4	40%	100%, 60%, 20%
5	50%	100%, 50%

Fig. 3 shows that the BigBird model trained using the smallest dataset (#5) obtained the best performance on 40% or more partial conversations. Overall, the inclusion of smaller proportions of a conversation in the training set, e.g., 20% or less, reduced model performance because the beginning part of conversations are noisy and contain little useful information for the outcome of a conversation.

Comparing the accuracy of the BigBird model trained using a mix of partial conversations of 50% ratio and whole (100%)

conversations in Fig. 3 with that of the BigBird model trained using whole conversations in Fig. 2, we observed that the model trained using whole conversations performed better when 70% or more of the conversations served as inputs for testing, but worse on when 20% to 60% of the conversation served as inputs for testing. The reason is that the training examples of 50% of the conversation contain less information than those that include more of the conversation and were reducing prediction accuracy based on larger portions of the conversation. But these shorter partial conversations in the training set show the model what to look for when handling the test cases of 20% to 60% of the conversation, which is an advantage the model that only trained on the full conversation does not possess.

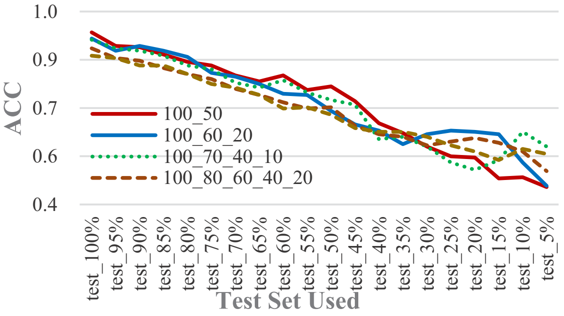

Fig. 3. Prediction accuracies of 5 BigBird models fine-tuned using different mixed partial conversation training sets on different partial conversations in the test set.

V. Conlcusion

In this paper, we investigated the feasibility of adopting local LLMs with prompt engineering and fine-tuning techniques to recognize the outcome of business calls based on complete call transcripts and to predict the outcome of business calls based on transcripts of partial conversations. The methods used 3 state-of-the-art open-source LLMs: BigBird, Alpaca-Lora-7B, and Llama2-70B-Chat. In experimental evaluation of their performances using a real-world dataset, we found that fine-tuned models performed much better than models with prompt engineering, which, in turns, outperformed zero-shot LLMs. On outcome recognition tasks, two fine-tuned models, Alpaca-Lora-7B with prompt engineering and LoRa-based fine-tuning and BigBird with fine-tuning achieved over 94% accuracy, much higher than the 85% accuracy achieved by Llama2-70B-Chat with various prompt engineering techniques, even although Llama2-70B-Chat is much larger than Lora-7B or BigBird. On outcome prediction based on partial conversations, BigBird models fine-tuned using moderate partial conversations achieved high prediction accuracy. For example, BigBird fine-tuned using the beginning 70% portion of conversations achieved more than 85% prediction accuracy based on the beginning 70% of each conversation in testing. Also, using a mixture of multiple training sets (i.e. 100%+50%) could improve the robustness of trained models compared to using a single training set (i.e. 100%). These results show that LLMs have the potential to become an effective tool in recognizing business call outcomes and predicting call outcome in the middle of a business call. For future work we will acquire data from more companies to test the robustness of different methods.

Acknowledgment

ChatGPT was used to improve the spelling, grammar, and overall readability.

References

[1] A. Vaswani et al., “Attention is all you need,” Adv Neural Inf Process Syst, vol. 30, 2017.

[2] S. Tabinda Kokab, S. Asghar, and S. Naz, “Transformer-based deep learning models for the sentiment analysis of social media data,” Array, vol. 14, 2022.

[3] C. Li et al., “Deep Formality: Sentence Formality Prediction with Deep Learning,” in 2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI), 2022.

[4] L. Hu, C. Li, W. Wang, B. Pang, and Y. Shang, “Performance Evaluation of Text Augmentation Methods with BERT on Small-sized, Imbalanced Datasets,” in 2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI), 2022.

[5] J. A. Leite, O. Razuvayevskaya, K. Bontcheva, and C. Scarton, “Detecting misinformation with llm-predicted credibility signals and weak supervision,” arXiv preprint arXiv:2309.07601, 2023.

[6] C. Chen and K. Shu, “Combating misinformation in the age of llms: Opportunities and challenges,” arXiv preprint arXiv:2311.05656, 2023.

[7] C. Li et al., “How Well Can Language Models Understand Politeness?” in 2023 IEEE Conference on Artificial Intelligence (CAI), 2023.

[8] A. Elhafsi, R. Sinha, C. Agia, E. Schmerling, I. A. D. Nesnas, and M. Pavone, “Semantic anomaly detection with large language models,” Auton Robots, vol. 47, no. 8, pp. 1035–1055, 2023.

[9] D. R. Gopagoni, P. V Lakshmi, and P. Siripurapu, “Predicting the Sales Conversion Rate of Car Insurance Promotional Calls,” in Rising Threats in Expert Applications and Solutions, V. S. Rathore, N. Dey, V. Piuri, R. Babo, Z. Polkowski, and J. M. R. S. Tavares, Eds., Singapore: Springer Singapore, pp. 321–329, 2021.

[10] Y. Feng, Y. Yin, D. Wang, and L. Dhamotharan, “A dynamic ensemble selection method for bank telemarketing sales prediction,” J Bus Res, vol. 139, pp. 368–382, 2022.

[11] N. N. Y. Vo, S. Liu, X. Li, and G. Xu, “Leveraging unstructured call log data for customer churn prediction,” Knowl Based Syst, vol. 212, p. 106586, 2021.

[12] Z. Peng, W. Wang, B. Balducci, D. Marinova, and Y. Shang, “Toward Predicting Communication Effectiveness,” in 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), 2018.

[13] Y. Qin and Y. Yang, “What You Say and How You Say It Matters: Predicting Stock Volatility Using Verbal and Vocal Cues,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019.

[14] C. Li, W. Wang, B. Balducci, D. Marinova, and Y. Shang, “Predicting Conversation Outcomes Using Multimodal Transformer,” in 2021 International Joint Conference on Neural Networks (IJCNN), 2021.

[15] M. Zaheer et al., “Big Bird: Transformers for Longer Sequences,” in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., pp. 17283–17297, 2020.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.

[17] Y. Liu et al., “Roberta: A robustly optimized bert pretraining approach,” arXiv preprint arXiv:1907.11692, 2019.

[18] Y. Yu et al., “Low-Rank Adaptation of Large Language Model Rescoring for Parameter-Efficient Speech Recognition,” in 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2023.

[19] H. Touvron et al., “Llama 2: Open foundation and fine-tuned chat models,” arXiv preprint arXiv:2307.09288, 2023.

[20] R. Taori et al., “Alpaca: A strong, replicable instruction-following model,” Stanford Center for Research on Foundation Models. https://crfm.stanford.edu/2023/03/13/alpaca.html, vol. 3, no. 6, p. 7, 2023.