Beyond the Turn-Based Game: Enabling Real-Time Conversations with Duplex Models

Anonymous ACL submission

Abstract

As large language models (LLMs) increasingly permeate daily lives, there is a growing demand for real-time interactions that mirror human conversations. Traditional turn-based chat systems driven by LLMs prevent users from verbally interacting with the system while it is generating responses. To overcome these limitations, we adapt existing LLMs to *duplex models* so that these LLMs can listen for users while generating output and dynamically adjust themselves to provide users with instant feedback. Specifically, we divide the queries and responses of conversations into several time slices and then adopt a time-division-multiplexing (TDM) encoding-decoding strategy to pseudo-simultaneously process these slices. Furthermore, to make LLMs proficient enough to handle real-time conversations, we build a fine-tuning dataset consisting of alternating time slices of queries and responses as well as covering typical feedback types in instantaneous interactions. Our experiments show that although the queries and responses of conversations are segmented into incomplete slices for processing, LLMs can preserve their original performance on standard benchmarks with a few fine-tuning steps on our dataset. Automatic and human evaluation indicate that duplex models make user-AI interactions more natural and human-like, and greatly improve user satisfaction compared to vanilla LLMs. Our duplex model and dataset will be released.

1 Introduction

Large language models (LLMs) have demonstrated impressive capabilities in various scenarios (OpenAI, 2023b; Achiam et al., 2023; Touvron et al., 2023; Team et al., 2023). These large models are deeply integrated with our daily lives and their extraordinary capabilities can satisfy users in many applications, such as coding assistants (Chen et al., 2021; GitHub, 2023b,a; Microsoft, 2024; Rozière et al., 2023; Li et al., 2023b), task assistants (Wang et al., 2023b; Qian et al., 2023; OpenAI, 2024), virtual role play (Shao et al., 2023; Shanahan et al., 2023), and even emotional companions (Chaturvedi et al., 2023; Guingrich and Graziano, 2023; Pentina et al., 2023).

Despite ongoing advancements, interactions with LLMs often fail to provide users human-like interaction experience (Hill et al., 2015; Mou and Xu, 2017; Zhou et al., 2023). One reason is the turn-based nature of current chatbot implementations (Skantze, 2021), which is different from human conversations where there are many overlaps, interruptions, and silences (Zimmerman and West, 1996). Current human-LLM interactions necessitate that one participant remains entirely idle while the other generates responses. Interruptions are manually triggered with a "stop" button or by saying certain keywords, resulting in conspicuously artificial communication. In human conversations, participants simultaneously process incoming information and formulate responses, often in overlapping and interleaved contexts, thus allowing each other to interrupt or be interrupted.

To address this limitation, we introduce the concept of **duplex models**. Duplex models emulate human cognitive processes by synthesizing responses internally while simultaneously attending to incoming user inputs, akin to a person thinking while listening as well as speaking while observing. However, present autoregressive models face substantial challenges in adopting a duplex configuration, as they must process and encode a complete input message before generating any tokens, resulting in a turn-based conversation. Considering this, we propose a framework for quickly converting current LLMs into duplex models by processing queries and responses pseudo-simultaneously without significant alternations to their architectures.

Specifically, we propose a time-division-multiplexing (TDM) encoding-decoding strategy.

messages in dialogues are split into time slices and the model processes time slices of input queries incrementally and generates time slices of output responses based on these partial input slices. When a new input query arrives, the model immediately halts its current generation process and starts a new sequence that integrates the additional input, enabling swift responses. To adapt existing LLMs to this format of time slices, we build a duplex dataset for fine-tuning. The differences between our data from the conventional supervised fine-tuning (SFT) dataset are: (1) its input and output are time slices and (2) it includes various interactive user interruptions, such as generation termination, regeneration, and dialogue reset.

To demonstrate the feasibility of duplex models, we train a prototype named MiniCPM-duplex, based on MiniCPM—a robust and lightweight LLM (Hu et al., 2024). Empirical results show that MiniCPM-duplex has its original performance on general benchmarks while enabling dynamic responses to user queries. Additionally, we conduct a user study to compare the MiniCPM-duplex with the original MiniCPM. The results indicate that duplex models show significant improvements in responsiveness, human-likeness, and user satisfaction. Our contributions are fourfold:

(1) We introduce and define the concept of duplex models, which are designed to generate output simultaneously as they receive input.

(2) We propose a TDM encoding-decoding strategy and a duplex-specific SFT dataset for implementing duplex models.

(3) We confirm that segmenting time slices during interactions does not compromise performance, and notably enhances the responsiveness, human-likeness, and overall satisfaction of conversations.

(4) We release the model and dataset and provide a demo for users to experience firsthand.

2 Duplex Models

We define *duplex models* as models that can process inputs and produce outputs simultaneously, and dynamically decide when to respond. It differs from current LLMs-based chatbots where participants must specify the end of inputs and only produce outputs after processing the entire input. To convert existing LLMs into duplex models, we split conversation messages into time slices, and then propose a TDM encoding-decoding mechanism to process these slices. To enhance the processing of these time slices, we further introduce duplex alignment to adapt existing LLMs to duplex models.

2.1 Time-Division-Multiplexing Encoding-Decoding

Current autoregressive language models struggle to function as true duplex systems. During the input phase, the LLM encodes the input into key-value caches without generating any output. To leverage autoregressive models in approximating duplex models, we propose a TDM strategy. We divide the conversation interaction into time slices and process input slices immediately to produce corresponding output slices.

Instead of requiring users to specify when the model should respond, the duplex model infers responses after every k seconds, i.e., each time slice spans k seconds. A special token (e.g., <idle>) is used to indicate the model's decision to remain silent and wait for further inputs. If not used, the generated slice is delivered to the user immediately. This approach mimics human conversational patterns more closely, as humans do not use special tokens to signal the end of utterances and intuitively determine the appropriate moments to respond to inputs. Figure 1 illustrates the distinction between duplex and conventional language models.

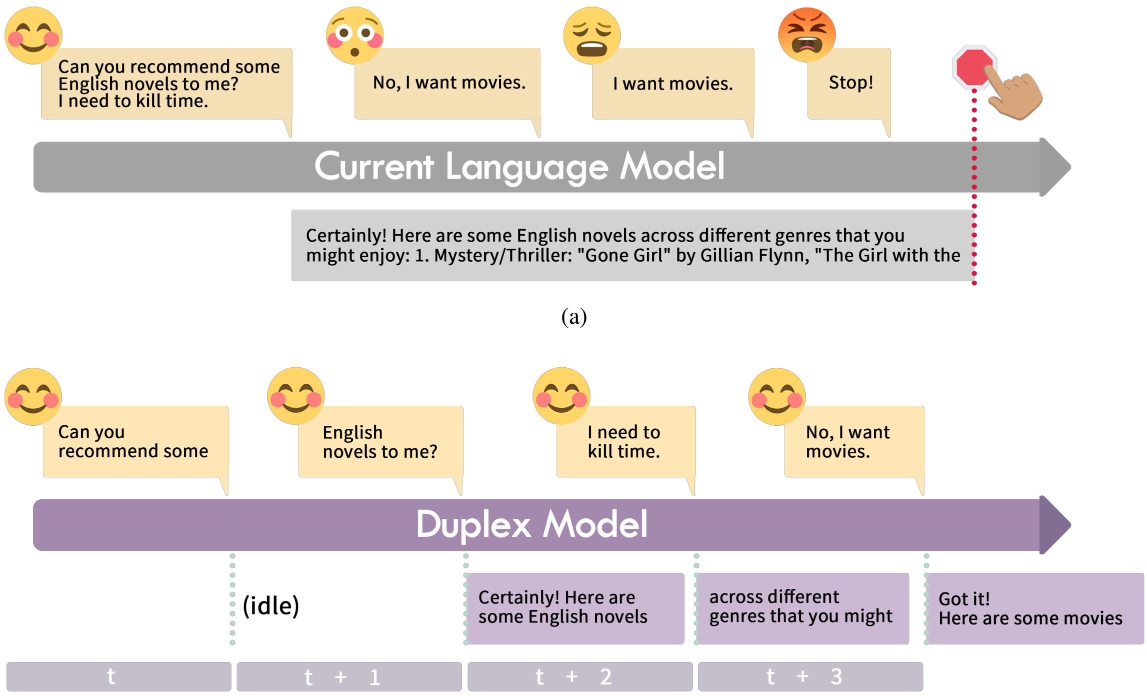
2.2 Time-Slicing Chunking

As shown in Figure 1, all the input queries and output responses of conversations are in the slice format. The size of slices has great implications for the performance of a duplex model. Large slice sizes result in greater response (or interruption) latency, while smaller slice sizes may result in unnecessarily long inputs (because some tokens are added between the chunks). Our preliminary investigation and pilot experiments with our transformer-based (Vaswani et al., 2017) models reveal that time-slicing chunking at 2-second intervals balances response latency and user experience. Assuming human beings usually speak 110-170 words per minute[1], an appropriate size of time slices is 4-6 words.

2.3 Duplex Alignment

Normal LLMs are unable to handle time slices as shown in Figure 2, so we need to fine-tune them into duplex models. To achieve this, we construct a duplex SFT duplex dataset.

[1] https://debatrix.com/en/speech-calculator/



(a)

(b)

Figure 1: Illustration of the input/output processing scheme of traditional models (1a) and duplex models (1b). Traditional models receive the complete input from the user before generating the response. In contrast, duplex models process the input and generate the output in an online manner.

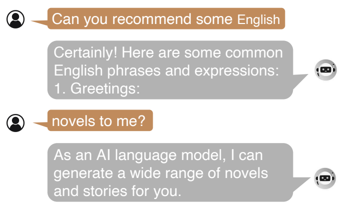


Figure 2: Responses of MiniCPM when inputs are time slices.

3 Supervised Fine-Tuning Duplex Dataset

We create **Duplex-UltraChat** for tuning current LLMs into duplex models. Different from existing dialogue datasets, Duplex-UltraChat has no special tokens or keywords to indicate the beginning or end of messages. Messages are split into time slices. A slice is either the actual message of an individual or a special "idle" token to indicate silence. Each individual may interrupt by generating a response before the other party's message is completed.

Duplex-UltraChat is derived from UltraChat (Ding et al., 2023) to reduce annotation costs. We heuristically inject appropriate random interruptions to simulate realistic scenarios. Powerful LLMs rewrite the interruptions to ensure diversity and naturalness. Each user message is randomly split into 4-6 words. Assistant messages are split into 10-token slices.

During the construction of the dataset, we abide by the following two design choices: user behavior is unpredictable and the assistant should be polite. Examples in the dataset can be categorized as uninterrupted dialogues and dialogues with interruptions. As shown in Table 1, there are six categories of duplex data consisting of over 4.8M dialogues. Each piece of data has an average length of 2,570.2 tokens encoded by the tokenizer of MiniCPM-duplex and 170.4 slice pairs.

3.1 Uninterrupted Dialogue

Basic Ordinary uninterrupted dialogue data is obtained by splitting existing dialogue messages into slices. When the user input is unfinished, the output of the duplex model should be `<idle>`. Meanwhile, when the duplex model is generating output, the user is set to quiet and its input is `<idle>`. Figure 3 shows an example of basic duplex data.

Topic Interweaving People may discuss several topics interweavingly ignoring coherence. To mimic such behavior, we interlace sentences of 3-5 dialogues while keeping their orders, and split each sentence into time slices as the basic type does.

Example Type	# Dialogues	Avg. # Slice Pairs	Avg. # Tokens
Basic	1,458,353	153.9	2,342.2
Topic Interweaving	489,065	427.7	6819.6
Generation Termination	1,468,141	89.3	1,318.0
Regeneration	806,687	171.2	2,590.4
Dialogue Reset	300,318	194.7	2,906.5
Back on Topic	327,286	199.1	2495.6
Total	4,849,850	170.4	2,570.2

Table 1: The statistics of Duplex-UltraChat. The tokens are produced by the tokenizer of our MiniCPM-duplex.

(a) Basic (b) Topic interweaving

Figure 3: An example of uninterrupted dialogue in Duplex-UltraChat.

3.2 Dialogues with Interruptions

In realistic human conversions, the individuals may start speaking before the other part is done with their message. Therefore, to simulate such scenarios, we inject four interruptions into the data as shown in Figure 4.

Generation Termination Forced interruptions are when users directly speak out their next sentence regardless of the status of the assistant. To generate such data, we randomly choose a location in an assistant message, discard the remaining part of the message, and insert a new user input at that location. We prefix the user input with one of the 11 pre-defined transitional sentences (see Appendix A.1). This input is rewritten by ChatGPT to ensure a natural and varied transition. The target output is idle tokens because the assistant is expected to terminate its current response.

Generation termination requires the assistant to learn to stop speaking when the user is forcibly interrupting it and be robust to incomplete messages in the chat history. Since this interruption may be regarded as impolite, our dataset does not contain situations where the user is interrupted.

Regeneration Another scenario where the user interrupts the assistant is when the user is dissatisfied with the current response. In conventional LLM-based chatbots, the user must first stop the generation with a button, and prompt the model with the updated prompt. In contrast, duplex models allow the user to directly interrupt and reinput the new prompt while generating outputs. To create such data, we randomly sample a user message and repeat it with one of 15 pre-defined transition sentences (given in Appendix A.2). ChatGPT rewrites this repetition message for better coherence. Then, the chat history and repetition message are fed to ChatGPT to generate the annotation.

Dialogue Reset Here, we consider situations where the user wants to chat abruptly on an entirely different topic while the assistant is generating output. To create such data, we randomly sample five dialogues and truncate the first four dialogues at random locations before concatenation. We define 18 kinds of transitional sentences in Appendix A.3, including one empty string. We randomly choose a transitional sentence, and prefix it with the first sentence of the new dialogue. Each message is then rewritten by ChatGPT. If the selected transitional sentence is the empty string, we do not rewrite the input, which simulates certain users who wish to start a new dialogue as fast as possible.

Back on Topic When the user only interrupts a question without attempting to stop the assistant or change the topic, the assistant should answer the question and then continue the unfinished statement. To construct this type of data, we randomly select a within a message from the assistant, and annotate a question about a statement by the assistant. GPT-4 (Achiam et al., 2023) is used to generate the answer to the user's question and continue the interrupted message with coherence.

(a) Termination (b) Regeneration (c) Dialogue reset (d) Back on topic

Figure 4: Some examples from Duplex-UltraChat.

4 Experimental Details

4.1 Training

We start from the public checkpoint of MiniCPM-2.4B (Hu et al., 2024)[2] and fine-tune it on Duplex-UltraChat as well as the SFT data that MiniCPM uses to obtain MiniCPM-duplex.

We make the following modifications to MiniCPM: (1) we append a special end-of-sentence token (i.e., <eos>) to each response of the duplex model, and (2) we add a special token <idle> to represent empty input or output.

The training of MiniCPM-duplex uses the following hyperparameters: 10^{-3} maximum learning rate, warmupstableexp (Hu et al., 2024) learning rate scheduler, a batch size of 800, and a maximum length of 4,096. We train for 10,000 steps on 40 NVIDIA A100 GPUs for 36 hours.

4.2 Baseline

Since our MiniCPM-duplex and MiniCPM are derived from the same checkpoint, we verify the effectiveness of our method by comparing it against the vanilla MiniCPM.

4.3 Evaluation

We evaluate the duplex model with three kinds of metrics: automatic metrics, GPT-4, and human. Automatic metrics, like accuracy and pass rate, are widely used for convenience and low cost.

4.3.1 GPT-4 Evaluation

To evaluate the multi-turn dialogue ability of MiniCPM-duplex, we benchmark it on MT-Bench (Zheng et al., 2024) with GPT-4 as the judge.

To mimic real-time scenarios, we chunk each instruction in MT-Bench into multiple 4-6 word slices and feed one slice at a time. Then we concatenate all output segments from the duplex model to form the final output. For the traditional model, we directly feed the entire prompt to the model.

Both models use the same decoding parameters: random sampling, a temperature of 0.8, a top-p value of 0.8, and a top-k value of 0. The maximum length is set to 4,096. For the duplex model, we set the maximum token generated per chunk to 10.

4.3.2 Human Evaluation

When using humans as evaluators, we consider the following four aspects.

Responsiveness This metric measures whether a model will respond to a user query and the latency if it responds, which is a perceived latency. Many factors may contribute to greater response latency, including the speech-to-text and text-to-speech conversion time, model inference time, network latency, and the interaction strategy that the model utilizes. There is no obvious difference between the actual inference latency of MiniCPM-duplex and MiniCPM.

Human-Likeness Inspired by the Turing test, we wish to develop a language model that chats in a way indistinguishable from humans. Therefore, we define human-likeness as a metric that measures the degree of the similarity of a model to humans.

Faithfulness Faithfulness is a widely used metric in the evaluation of LLMs (Arras et al., 2017; Serrano and Smith, 2019; Jain and Wallace, 2019; DeYoung et al., 2020; Adlakha et al., 2023; Chen et al., 2023b). Here, we use it to reflect the degree how the model follows a user's instruction, which

[2] `https://huggingface.co/openbmb/MiniCPM-2B-sft-bf16`, denoted MiniCPM.

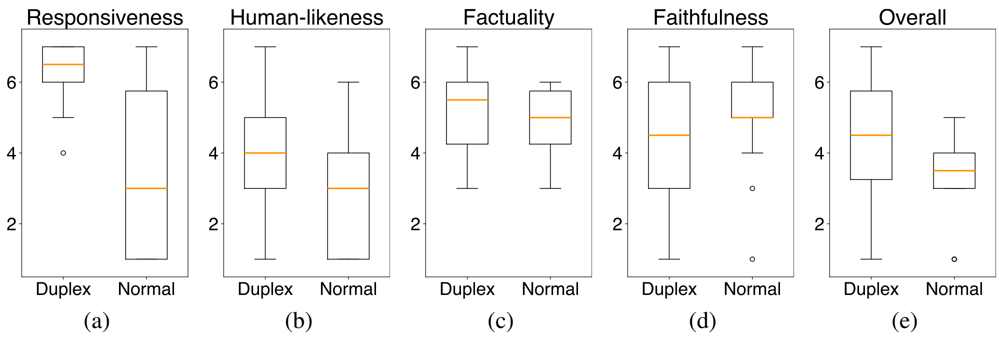


Figure 5: The human evaluation score distributions for MiniCPM and MiniCPM-duplex regarding responsiveness, human-likeness, factuality, faithfulness, and overall satisfaction.

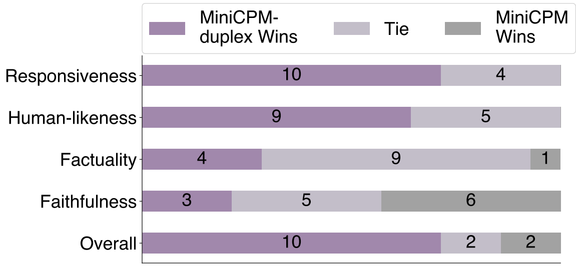


Figure 6: Win rates between MiniCPM and MiniCPM-duplex on responsiveness, human-likeness, factuality, faithfulness, and overall satisfaction.

is similar to (Adlakha et al., 2023).

Factuality This metric measures the degree of hallucination of a LLM (Rudinger et al., 2018; Tian et al., 2023; Chen et al., 2023a; Wang et al., 2023a; Nakano et al., 2021).

4.4 Interactive Demo

We implement an interactive demo with a user interface such that human evaluators can make evaluations based on actual interaction experience. In the demo, users chat with an assistant using voice. The assistant is either implemented with the vanilla MiniCPM or our MiniCPM-duplex. The conversion between speech and text is implemented with Google's cloud-based ASR and TTS API[3].

This demo supports both vanilla MiniCPM and MiniCPM-duplex. For the vanilla MiniCPM, the program automatically detects pauses in the user's voice. On each pause, the speech is converted to text, which is then sent to the model. MiniCPM performs regular text generation, and each output token is passed to the ASR module, before being returned to the user. Meanwhile, the user has to wait until the speech response is done before the next query. When interacting with MiniCPM-duplex, the user's speech is processed every 2 seconds. When the MiniCPM-duplex does not generate the idle token, the text generation will be transcribed into audio and played out. The user's voice will be captured, transcribed, and fed to the model regardless of whether the assistant speaks.

[3]Speech-to-text API: https://cloud.google.com/speech-to-text/docs/reference/rest. Text-to-speech API: https://cloud.google.com/text-to-speech/docs/reference/rest.

Benchmark	MiniCPM	MiniCPM-duplex
C-Eval	**50.52**	50.06
CMMLU	**51.30**	48.53
MMLU	53.45	**53.76**
BBH	**37.25**	36.35
HumanEval	**50.00**	49.39
MBPP	38.09	**38.30**
GSM8K	42.30	**46.10**
MATH	**10.56**	9.32
ARC-e	84.60	**85.19**
ARC-c	69.80	**70.05**
HellaSwag	**61.40**	60.79

Table 2: Performances of MiniCPM and MiniCPM-duplex on standard benchmarks.

Metric	MiniCPM	MiniCPM-duplex
Responsiveness	3.43	**6.21**
Human-Likeness	2.79	**4.00**
Factuality	4.93	**5.21**
Faithfulness	**5.14**	4.50
Overall	3.29	**4.36**

Table 3: Average human evaluation scores on responsiveness, human-likeness, factuality, faithfulness, and overall satisfaction. Higher is better.

Score	MiniCPM	MiniCPM-duplex
First turn	**7.17**	5.83
Second turn	**5.85**	4.84
Avg.	**6.51**	5.33

Table 4: MT-bench results of MiniCPM and MiniCPM-duplex. Higher is better.

4.5 User Study

Specifically, we recruit 14 participants consisting of 5 males and 9 females from 18 to 35 years old. Each participant holds a Bachelor's or Master's degree. Details on employment, payment, and ethical review are in Appendix C.

During the experiment, we rename MiniCPM-duplex as Model A, and MiniCPM as Model B to ensure anonymity. Participants are unaware of the difference between the two models beforehand. We specify the odd-numbered participants interact with Model A first, and the even-numbered ones first chat with Model B to eliminate the influence of chatting order. When finishing chatting with a model, the participant should score it and continue interacting with the other one. After the experiment, participants could modify and confirm scores for both models. Each participant is assigned at least 5 sessions of multi-turn dialogues with each model. The first sentence of sessions should be the same for both models. To help the participants come up with topics to chat about, we provide them with a reference note containing sample instructions from AlpacaEval (Li et al., 2023c).

Questionnaire Design The questionnaire consists of six questions. The first five questions prompt the user to rate the model based on responsiveness, human-likeness, faithfulness, factuality, and overall experience. The answer choices for these questions are scores from 1 to 7, where 1 represents disappointment, 4 represents indifference, and 7 represents excellence. The final question is open to suggestions on improving our duplex model. The actual questions are listed in Appendix B.2.

5 Results

Standard Benchmarks MiniCPM-duplex is benchmarked on several standard benchmarks, including multitask (C-Eval (Huang et al., 2024), CMMLU (Li et al., 2023a), MMLU (Hendrycks et al., 2020), BBH (Suzgun et al., 2023)), code (HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021)), math (GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021)), and reasoning (ARC-e, ARC-c (Clark et al., 2018), HellaSwag (Zellers et al., 2019)) with the LLM evaluation platform, UltraEval (He et al., 2024). Table 2 indicates that adapting to duplex models does not significantly harm its performance on general benchmarks.

GPT-4 Evaluation Table 4 shows the GPT-4 evaluation results on MT-Bench. MiniCPM-duplex is slightly inferior to MiniCPM mainly due to that MiniCPM-duplex tends to generate shorter responses. GPT-4 favors longer responses, whereas users prefer chat models that give concise answers.

Human Evaluation We have received 14 questionnaire. Table 3 lists the average scores of both models on five metrics. The duplex model surpasses the normal model by 81.05%, 43.37%, and 32.52% on responsiveness, human-likeness, and overall experience respectively.

Apart from absolute scores, we compare the ratings of the two models and count the number of evaluators that rate one model higher. The comparison results are shown in Figure 6. MiniCPM is more faithful than the duplex model mainly because it uses more diverse SFT data. Whereas the duplex model wins in other aspects, with an exceptionally large margin on responsiveness and human-likeness.

From these results, we conclude that duplex models can provide a better user experience in acting as the backbone model in AI assistants compared to ordinary language models.

6 Analysis & Discussion

6.1 Analysis

The superior performance of the duplex model is mainly due to its underlying receive/generate mechanism. Rather than strictly turn-based dialogue where users must explicitly signal the beginning and end of messages, duplex models behave more like human beings. Besides, the duplex model has learned when to speak at the fine-tuning stage on the Duplex-UltraChat, which makes it more human-like. Such ability is essential in passing a non-turn-based version of the Turing test, which is a more realistic test for whether a machine can be indistinguishable from humans.

6.2 Discussions

We highlight some important open problems associated with duplex models below.

High-quality duplex data is urgently needed Existing dialogue datasets are inherently turn-based, which does not represent realistic and complex human conversations. Despite some success in empirical results with our synthetically generated duplex dataset, it still lags behind the practical demands. Two out of the 14 participants pointed out that they preferred concise responses rather than tedious answers.

We manually inspect 10 out of 90 chat sessions and find that the duplex model fails to remain silent once and interrupts the user unexpectedly once, showing that there is room for improvement. Thus, high-quality duplex datasets are in urgent need.

A new decoding strategy is needed to improve the chat experience There are failed cases where the duplex model interrupted users unexpectedly. Balancing response speed and user experience is an open problem. Besides, to be more human-like, the duplex model should learn to start dialogues or topics actively.

A custom TTS system is needed to smooth the output voice The duplex model generates output chunk by chunk, which causes the output voice to be chunked. This results in incoherent intonation and volume, harming the user experience because existing TTS software does not support transcribing sequentially provided text chunks into a contiguous smooth voice. Overcoming this problem will improve the user experience considerably.

7 Related Work

7.1 Dialogue Dataset

Dialogue data can be divided into two categories: single-turn and multi-turn.

Single-Turn Self-instruct (Wang et al., 2023c) is a synthetic instruction-following dataset of over 82K instances generated by GPT-3.5. Taori et al. (2023) adopt the data construction pipeline from Wang et al. (2023c) and construct Alpaca, a dataset with 52K instances. GPT-4-LLM (Peng et al., 2023) improves the Alpaca by replacing the data generator with GPT-4. It also adopts a Chinese version of Alpaca and Unnatural Instructions (Honovich et al., 2023). Besides, there are several high-quality datasets, such as BELLE (Ji et al., 2023) and GPT-4ALL (Anand et al., 2023), among others.

Multi-Turn DailyDialog (Li et al., 2017) consists of over 13K dialogues annotated by humans, covering diverse daily conversation scenarios. Baize (Xu et al., 2023) generates multi-turn dialogues with ChatGPT by a prompting framework called self-chat where seed questions are from Quora and Stack Overflow, two popular question-answering websites. SODA (Kim et al., 2022) contains dialogues involving social commonsense. UltraChat (Ding et al., 2023) focuses on 30 meta-concepts and 20 types of materials and consists of over 1.4M dialogues.

7.2 Dialogue Models

Chat-based models have gained widespread popularity since the release of ChatGPT. Some notable chat-based LLMs include the Claude series (Anthropic, 2023, 2024), Qwen series (Qwen, 2024), the Mistral series (Jiang et al., 2023) and and LLaMa series (Touvron et al., 2023), among others. Most of these models, especially open-sourced ones, are purely text-based.

To enhance user experience, several applications support voice interaction. One instance is ChatGPT, where users press a button before speaking and indicate the end of speech with a button or pausing (OpenAI, 2023a). Then ChatGPT processes the received signal and produces a response until it finishes or users interrupt it by pressing a button. Such an implementation is unrealistic because it requires the user to specify the beginning and end of inputs. Whereas, our MiniCPM-duplex may improve this interactive experience by teaching the model to learn when to speak and when to be silent.

8 Conclusion

We have introduced the concept of duplex models and provided one implementation. To this end, we also constructed the first non-turn-based dialogue dataset, Duplex-UltraChat, by injecting diverse kinds of interruptions into existing dialogue datasets. Our model, MiniCPM-duplex, is competitive with traditional models when evaluated on ordinary benchmarks while outperforming them in terms of responsiveness, human-likeness, and overall satisfaction. We believe that this work represents an essential step toward building machines that behave more human-like beyond current turn-based conversations.

Limitations

In this paper, we propose and verify the viability of duplex models. However, our implementation, MiniCPM-duplex, is a pseudo-duplex model, since it cannot perform encoding and decoding simultaneously. Consequently, our fixed-interval decoding strategy introduces a new hyperparameter that compromises responsiveness and context length (as discussed in Section 2.2). These limitations call for a new architecture that better supports the input-output scheme of duplex models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2023. Evaluating correctness and faithfulness of instruction-following models for question answering. *arXiv preprint arXiv:2307.16877*.

Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. 2023. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. *GitHub*.

Anthropic. 2023. Introducing claude 2.1. https://www.anthropic.com/news/claude-2-1.

Anthropic. 2024. Introducing the next generation of claude. https://www.anthropic.com/news/claude-3-family.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. " what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Rijul Chaturvedi, Sanjeev Verma, Ronnie Das, and Yogesh K. Dwivedi. 2023. Social companionship with artificial intelligence: Recent trends and future avenues. *Technological Forecasting and Social Change*, 193:122634.

Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. 2023a. Beyond factuality: A comprehensive evaluation of large language models as knowledge generators. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6325–6341.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Yijie Chen, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2023b. Improving translation faithfulness of large language models via augmenting instructions. *arXiv preprint arXiv:2308.12674*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051.

GitHub. 2023a. About github copilot chat. https://docs.github.com/en/copilot/github-copilot-chat/about-github-copilot-chat.

GitHub. 2023b. Copilot. https://github.com/features/copilot.

Rose Guingrich and Michael SA Graziano. 2023. Chatbots as social companions: How people perceive consciousness, human likeness, and social health benefits in machines. *arXiv preprint arXiv:2311.10599*.

Chaoqun He, Renjie Luo, Shengding Hu, Yuanqian Zhao, Jie Zhou, Hanghao Wu, Jiajie Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Ultraeval: A lightweight platform for flexible and comprehensive evaluation for llms. *arXiv preprint arXiv:2404.07584*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Jennifer Hill, W Randolph Ford, and Ingrid G Farreras. 2015. Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. *Computers in human behavior*, 49:245–250.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 14409–14428.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. 2024. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Yunjie Ji, Yong Deng, Yan Gong, Yiping Peng, Qiang Niu, Lei Zhang, Baochang Ma, and Xiangang Li. 2023. Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases. *arXiv preprint arXiv:2303.14742*.

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*.

Hyunwoo Kim, Jack Hessel, Liwei Jiang, Peter West, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Le Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. 2022. Soda: Million-scale dialogue distillation with social commonsense contextualization. In *Proceedings of the 2022 Empirical Methods in Natural Language Processing*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023a. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Raymond Li, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, LI Jia, Jenny Chim, Qian Liu, et al. 2023b. Starcoder: may the source be with you! *Transactions on Machine Learning Research*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023c. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Microsoft. 2024. Write code without the keyboard. https://githubnext.com/projects/copilot-voice/.

Yi Mou and Kun Xu. 2017. The media inequality: Comparing the initial human-human and human-ai social interactions. *Computers in Human Behavior*, 72:432–440.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

OpenAI. 2023a. Chatgpt can now see, hear, and speak. https://openai.com/blog/chatgpt-can-now-see-hear-and-speak.

OpenAI. 2023b. Introducing chatgpt. https://openai.com/blog/chatgpt#OpenAI.

OpenAI. 2024. Hello gpt-4o. Https://openai.com/index/hello-gpt-4o/.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Iryna Pentina, Tyler Hancock, and Tianling Xie. 2023. Exploring relationship development with social chatbots: A mixed-method study of replika. *Computers in Human Behavior*, 140:107600.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Qwen. 2024. Introducing qwen1.5. https://qwenlm.github.io/blog/qwen1.5/.

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

Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–744.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature*, 623(7987):493–498.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. https://crfm.stanford.edu/2023/03/13/alpaca.html.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023b. Voyager: An open-ended embodied agent with large language models. In *Intrinsically-Motivated and Open-Ended Learning Workshop@ NeurIPS2023*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 13484–13508.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6268–6278, Singapore. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Qi Zhou, Bin Li, Lei Han, and Min Jou. 2023. Talking to a bot or a wall? how chatbots vs. human agents affect anticipated communication quality. *Computers in Human Behavior*, 143:107674.

Dean H Zimmerman and Candace West. 1996. 9. sex roles, interruptions and silences in conversation. In *Towards a Critical Sociolinguistics*, page 211. John Benjamins.

A Transition Sentences

To generate a sentence with coherent context, we utilize ChatGPT to rewrite the template below, which replaces {sentence_a} and {sentence_b} with one transition sentence and new content respectively.

> Fuse the two sentences smoothly and replace [topic] with the topic of sentence two.
>
> Sentence one "{sentence_a}"
>
> Sentence two "{sentence_b}"
>
> Give me your answer only, no other words. Give me your answer only, no other words.

A.1 Generation Termination Transition Sentences

1. <Empty string>
2. I need to cut you off right now; this is urgent.
3. Excuse me, I need to interject for a moment.
4. Sorry to interrupt, but I have something important to add.
5. Excuse me, may I interrupt for a moment?
6. I'm sorry to break in, but there's something important I need to address.
7. I apologize for interrupting, but I'd like to interject for a moment.
8. I'm sorry to interrupt, but I have a quick point to make.
9. I appreciate your input, but I need a moment of silence now.
10. I'm sorry to interrupt, but I really need some quiet time to focus.
11. Enough talking! I need you to be quiet now.

A.2 Regeneration Transition Sentences

1. I may not have expressed myself clearly. What I meant was [topic]
2. I think there might be a bit of confusion. Let me clarify [topic]

3. I appreciate your input, but I was hoping for more details on [topic]
4. I think there might be a misunderstanding. What I'm really looking for is [topic]
5. I may not have explained myself clearly. Let me rephrase the question. What are your thoughts on [topic]?
6. Actually, the correct information is [topic]. Could you share your perspective on that?
7. I'm a bit confused because what you mentioned contradicts the information I have. Can we go over this again?
8. I'm sorry, but that information seems to be incorrect. Let me clarify the question, and please provide the accurate details regarding [topic].
9. I'm sorry, but that's not accurate. The correct information is [topic]. It's essential to have the correct details for our discussion.
10. I appreciate your effort in responding, but I think there might be a misunderstanding. What I intended to convey was [topic]. Let's revisit the topic to ensure we're on the same page.
11. I see there might be some confusion. Let me clarify my point further to ensure we're on the same page. What I meant was [topic]. Can we discuss this to make sure we have a mutual understanding?
12. There seems to be a misunderstanding. I meant [topic]. Let's align our understanding.
13. No.
14. Oh, No.
15. No, you are wrong.

A.3 Dialogue Reset Transition Sentences

1. <Empty string>
2. That's interesting, and speaking of [topic], have you ever...?
3. I was just thinking about [topic], what are your thoughts on that?
4. That's fascinating! On a different note, have you ever thought about [topic]?
5. I was just reading about [topic]. What are your thoughts on that?
6. By the way, speaking of something else.
7. That reminds me, have you heard about [topic]?
8. Can we shift gears for a moment and talk about [topic]?
9. I've been curious about [topic]. Have you ever considered it?
10. I was thinking about [topic]. What are your thoughts on that?
11. Now, shifting gears to a different subject, have you ever explored [topic]
12. Moving on to a different topic, have you ever considered [topic]
13. Changing the subject, have you ever thought about [topic]
14. Switching gears, let's talk about [topic]
15. On a different note, have you ever thought about [topic]
16. Speaking of which, have you ever considered exploring [topic]
17. Changing the subject, let's now delve into [topic]
18. Shifting gears a bit, let's talk about [topic]

B Questionnaire Details

B.1 Subject Instruction

Before the experiment, we inform each participant of the subject instruction. The whole instruction is listed below:

1. This experiment requires subjects to have conversations with chat models. The content does not involve any dangerous remarks or have an impact on the subjects' physical and mental health.
2. This test includes two parts: chatting and interacting with the models and filling out the questionnaire.

3. The models are voice input and output modes that support multiple rounds of dialogue. At the end of each dialogue, you can press the new conversation button to start a new round of conversation.

4. The models are English models and only support English dialogue.

5. There are two types of models, A and B. You must have at least 10 conversations with each model.

6. We have included some questions to start the conversation, just for reference.

7. This test mainly evaluates the performance of the two models in terms of response speed, human-likeness, faithfulness, factuality, and overall experience.

8. After the chat, fill out the questionnaire.

B.2 Questionnaire

1. Score the model's response speed to evaluate whether the model can respond to your request.

2. Score the faithfulness of the model's answers to evaluate whether the model understands your question, follows your instructions, and whether the answer is relevant to your chat topic.

3. Score the factuality of the model's answers and evaluate whether the content of the answers is correct.

4. Score the human-likeness of the model's answers and evaluate whether the conversation process between you and the model is close to the feeling of daily communication between people and whether the conversation process is smooth.

5. Score the overall experience of the model.

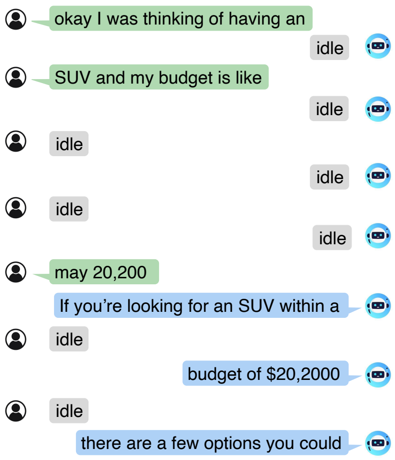
C Explanation of Ethical Concerns

All participants are recruited from a partner company. Those experiments are conducted during their working hours and we do not pay them additionally.

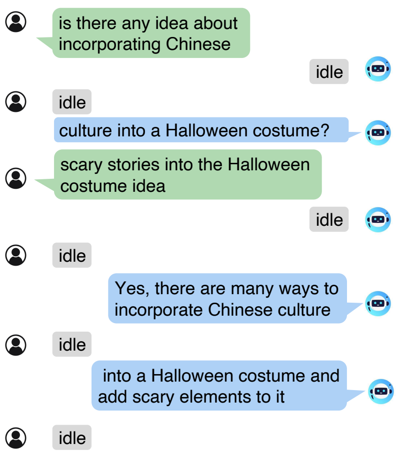
In the human-evaluation experiment, we collect basic demographic characteristics information: gender, age, and educational qualification. We also collect their knowledge and usage of LLMs and voice assistants, which is tightly related to our research topic. As for the evaluation of the two chat models, we utilize their experience. The participants permit all those characteristics and experience information collection for research purposes only.

D Case Demonstration

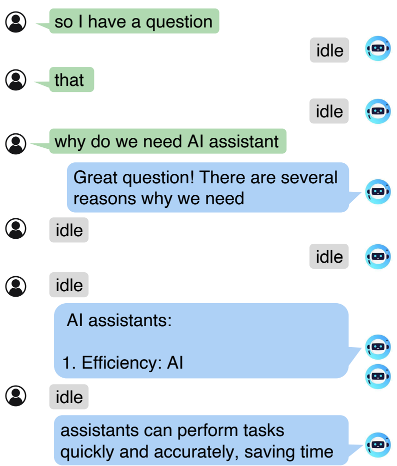
Here are some cases of conversation segments between the MiniCPM-duplex and human users. In Figure 7, the duplex model generates a response until it obtains enough information from the user.



(a) Case A



(b) Case B



(c) Case C

Figure 7: User study cases.