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

Anonymous ACL submission

Abstract

As large language models (LLMs) increasingly permeate daily life, there is a growing demand for interactions that mirror human conversation in real time. Traditional LLM-based chat systems are turn-based, preventing users from interacting verbally with the model while it generates output. To overcome these limitations, we introduce **duplex models**, which can receive inputs from users *while* generating outputs and adjust dynamically to instant user feedback such as interruptions. To endow model LLM architectures with such characteristics, we utilize a time-segment decoding strategy that enables the model to process inputs and generate responses pseudo-simultaneously. Furthermore, to make the LLMs proficient in handling real-time conversations, we construct a fine-tuning dataset with interleaved pieces of time-segmented input and output and include typical types of feedback in instantaneous interactions. In the experiments, we find that although the inputs and outputs are segmented into incomplete pieces, the model preserves its performance on standard benchmarks with a few steps of training. Moreover, this approach makes user-AI interactions more natural and human-like, thus greatly improving user satisfaction in our user experiments. The model and dataset will be released.

1 Introduction

Large language models (LLMs) have demonstrated impressive capabilities in various scenarios (OpenAI, 2023c,b). They are more integrated with people's daily lives, such as coding assistants (Chen et al., 2021; GitHub, 2023b,a; Microsoft, 2024; Rozière et al., 2023; Li et al., 2023), task assistants (Wang et al., 2023b; Qian et al., 2023), 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). The extraordinary capabilities of LLMs can satisfy users in many applications.

Despite ongoing advancements, interactions with LLMs often fail to mirror the real-time dynamics inherent in human conversations. We assert that the primary difference between contemporary human-LLM exchanges and human-to-human dialogues resides in the modes of interaction. In human conversations, participants simultaneously process incoming information and formulate responses, often overlapping and interjecting, thus allowing for interruptions or being interrupted. In contrast, current human-LLM interactions necessitate that one participant remains entirely passive and idle while the other generates responses. Interruptions must be artificially initiated, either by clicking a "stop" button or saying certain keywords, resulting in a communication format with LLMs that is conspicuously artificial, particularly in speech.

To address this limitation, we introduce the concept of **duplex models**. Ideally, in duplex models, the system would emulate human cognitive processes by synthesizing responses internally while simultaneously attending to incoming user inputs, akin to a person thinking while listening, and speaking while observing. However, present autoregressive models face substantial challenges in adopting a duplex configuration, as they must process a full input sentence into key-value caches before generating any new tokens, resulting in a turn-based conversation. In this paper, we propose a framework to establish a pseudo-duplex model that behaves similarly to a true duplex system without necessitating significant alterations to the foundational model architecture.

We adopt two strategies to approximate a duplex model. The first strategy involves a time-segmented decoding approach, where the model processes segments of input incrementally and generates responses based on these partial inputs. When a new input arrives, the model immediately halts its current output generation and starts a new sequence

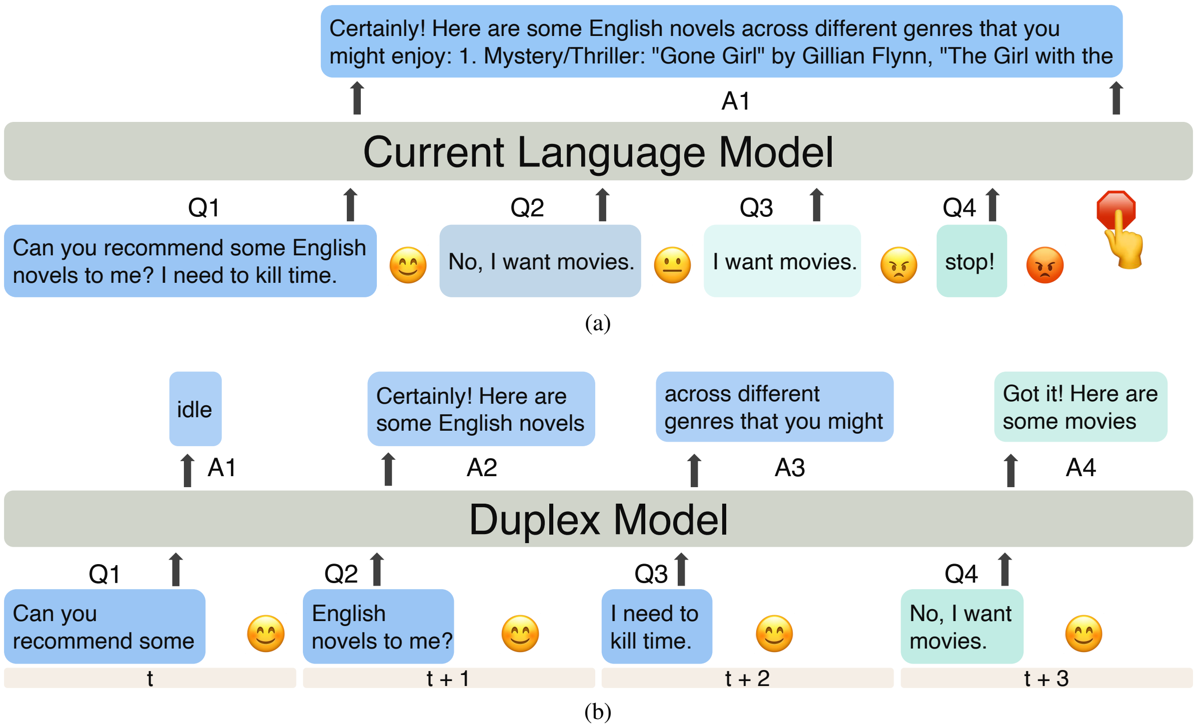

Figure 1: Illustration of the input/output processing scheme of traditional models (1a) and duplex models (1b). Traditional models receive the complete input from the user before generating the response. In contrast, duplex models process the input and generate the output in an online manner.

that integrates the additional input, enabling swift responses. The second strategy entails fine-tuning traditional models with a dataset structured in a duplex format. This dataset has two differences compared to the conventional dataset: (1) its input and output are time-segmented; (2) it includes various interactive user interruptions, such as generation termination, regeneration, and dialogue reset. Training a normal chat model on this dataset ensures that the model adeptly handles fragmented and incomplete sentence segments.

To explore the feasibility of duplex models, we develop a prototype named MiniCPM-duplex, based on MiniCPM—a robust yet lightweight small language model (Hu et al., 2024). We assess MiniCPM-duplex's performance against traditional benchmarks and confirm that the additional training does not degrade the model's performance on these benchmarks while enabling the model to dynamically respond to user inputs. Additionally, we engage 28 participants to compare the MiniCPM-duplex with the original MiniCPM. The results indicate significant improvements in human-likeness and overall satisfaction with the duplex models. Our contributions are fourfold:

- We introduce and define the concept of **duplex models**, which are designed to generate output simultaneously as they receive input.
- We devise two strategies for implementing pseudo duplex models: a time-segmented decoding strategy and a duplex-specific supervised fine-tuning (SFT) dataset.
- We confirm that segmenting time during interactions does not compromise performance, while notably enhancing the human-likeness and overall satisfaction of conversations.
- 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 language models which require that the participants specify the end of inputs and only produce outputs after the entire input is processed.

Time-Segmented Decoding Current language models struggle to function as truly duplex systems using autoregressive models. 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 "time-segmented decoding" strategy. We divide the interaction into fixed time segments and process inputs immediately within these segments to produce corresponding outputs. Instead of requiring users to specify when the model should respond, the duplex model infers responses after every k seconds. A special token (e.g., <idle>) indicates the model's decision to remain silent and wait for further inputs. If not used, the generated text 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 must intuitively determine the appropriate moments to respond to prompts from the context. Figure 1 illustrates the distinction between duplex and conventional language models.

3 Duplex Dataset

For adapting existing language models into duplex models, we construct and release a dialogue dataset, **Duplex-UltraChat**. Different from existing dialogue datasets, in Duplex-UltraChat, there are no special tokens or keywords to indicate the beginning or end of messages. Each message is split into chunks, and each dialogue example consists of alternating chunks of text between a user and an assistant. Each chunk is either the actual message of an individual or a special "idle" token to indicate that the individual has decided not to say anything yet. Each individual may also interrupt by generating a response before the other party's message is completed.

To reduce annotation costs, we choose to start from existing high-quality dialogue datasets. We split messages into chunks and heuristically inject appropriate random interruptions to simulate realistic scenarios where each individual in a dialogue may interrupt the other individual. ChatGPT (OpenAI, 2023c) then rewrites the interruptions to ensure diversity and naturalness. This dataset is based on a widely-used dialogue dataset, UltraChat (Ding et al., 2023).

During the construction of the dataset, we abide by the following two design choices: (1) user behavior is unpredictable, and (2) the assistant should be polite.

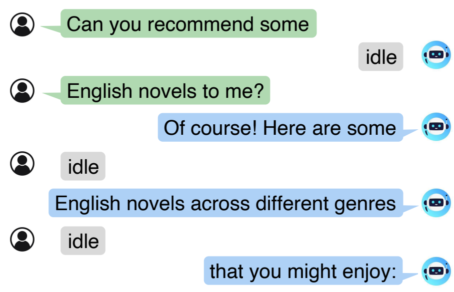

Figure 2: An example of uninterrupted dialogue in Duplex-UltraChat.

3.1 Chunk Sizes

We first establish an appropriate chunk size. Large chunk sizes result in greater response (or interruption) latencies, while smaller chunk sizes may result in exceeding long inputs (because some tokens are added between the chunks). Our preliminary survey with our transformer-based model reveals that chunking at 2-second intervals balances response latency and user experience. Assuming humans speak 110-170 words per minute [1], an appropriate chunk size is 4-6 words. Therefore, we choose to split user messages into 4, 5, or 6 words randomly, with the probability of 10%, 80%, and 10%, respectively. As for model messages, we uniform 10 tokens as a segment.

3.2 Uninterrupted Dialogue

Ordinary uninterrupted dialogue data is obtained by splitting existing dialogue messages into segments. 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 2 shows an example of basic duplex data.

3.3 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 three types of interruptions into the data, which we will describe below.

3.3.1 Generation Termination

Forced interruptions are when users directly speak out their next sentence regardless of the status of the duplex model. To generate such data, we randomly choose a location in the assistant's output, discard the remaining part of the assistant's output,

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

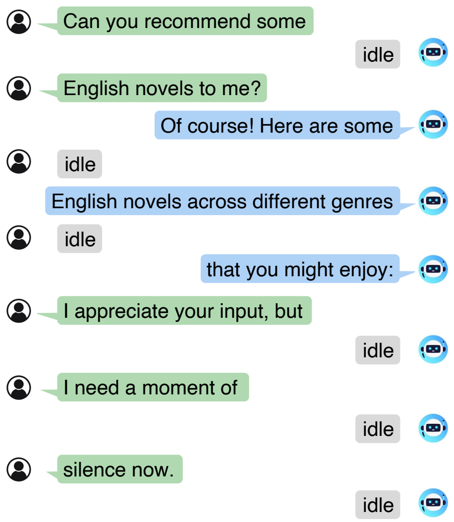

Figure 3: An example of generation termination in Duplex-UltraChat.

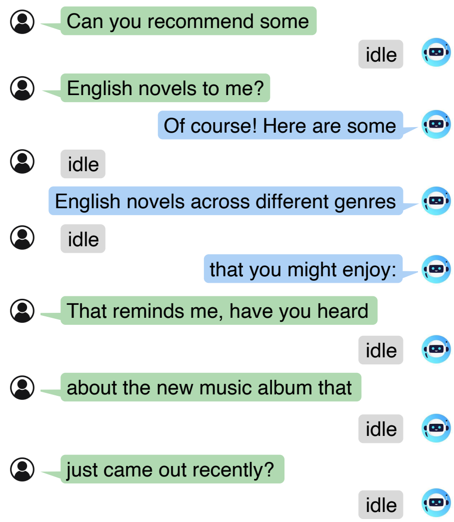

Figure 4: An example of dialogue reset in Duplex-UltraChat.

and insert a new user input at that location. Figure 3 shows one example of generation termination.

Contrary to existing dialogue data, the introduction of forced interruptions 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 forced interruptions may be regarded as impolite for many users, our dataset only contains situations where the assistant is forcibly interrupted. We define 11 transitional sentences (see Appendix A.1). We randomly choose a transitional sentence, and prefix it with the next sentence of the user as new input. This input is rewritten by ChatGPT to ensure a natural and varied transition. The target output is idle tokens because the assistant to expected to terminate its current response.

3.3.2 Regeneration

Another scenario in which 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 then prompt the model with the updated prompt. In contrast, duplex models allow the user to directly interrupt and reinput the new prompt while the LLM is generating the response. To generate such data, we randomly pick a user message and repeat it with one of 15 pre-defined transition sentences (given in Appendix A.2). This repetition message is rewritten by ChatGPT for better coherence. Then, the chat history along with the repetition message is fed to ChatGPT to generate

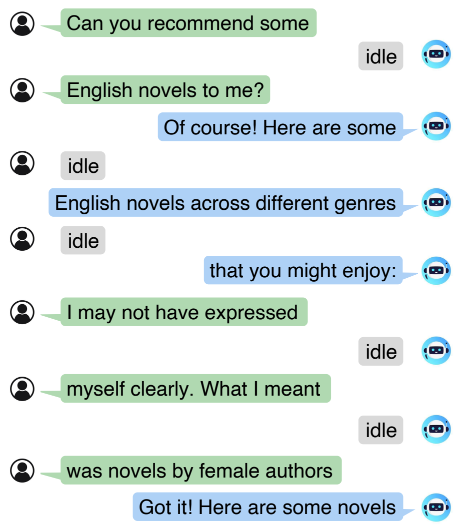

Figure 5: An example of regeneration in Duplex-UltraChat.

the annotation.

3.3.3 Dialogue Reset

Here, we consider situations where the user wants to abruptly chat on an entirely different topic while the assistant is generating output. This corresponds to the user clicking a “new chat” button in current chatbot systems. A capable chatbot should be able to infer such demand from the context.

To create such data, we random sample five dialogues in a random order, and truncate the first four dialogues at random locations before concatenation. We define 18 kinds of transitional sentences (see Appendix A.3), including one empty string. We randomly choose a transitional sentence, and prefix

Example Type	# Dialogues	Avg. # Segment Pairs	Avg. # token
Uninterrupted Dialogue	1,458,353	153.9	2,342.2
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
Total	4,033,499	136.9	2,061.1

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

it with the first sentence of the new dialogue. Each data is then rewritten by ChatGPT to ensure consistency and diversity. 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.

3.4 Data Statistics

As shown in Table 1, there are four categories of duplex data consisting of over 4M dialogues. Each piece of data has an average length of 2061.1 tokens encoded by the tokenizer of MiniCPM-duplex and 136.9 segment pairs.

4 Experimental Details

4.1 Training

We start from the publicly released MiniCPM-2.4B (Hu et al., 2024), and fine-tune it on Duplex-UltraChat 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 uses the following hyperparameters: 10^{-4} maximum learning rate, a batch size of 1280, and a maximum length of 4096. We train for 5000 steps on 64 NVIDIA A100 GPUs for 18 hours (8 machines, each with 8 GPUs).

4.2 Baseline

Since our MiniCPM-duplex is obtained by continued training of MiniCPM, we verify the effectiveness of our method by comparing it against the vanilla MiniCPM.

4.3 Evaluation

Some important aspects of duplex models cannot be captured with existing metrics for LLM-based chatbots. Therefore, in addition to evaluating the quality of responses, we also introduce other metrics that measure attributes that may provide a better user experience. We use both GPT-4 and humans as evaluators.

4.3.1 GPT-4 Evaluation

Similar to traditional LLMs, it is important to ensure the quality of response contents. To evaluate the response quality of MiniCPM-duplex, we benchmark it on AlpacaEval 2.0 [2]. This is a preference-based benchmark in which an evaluator compares the quality of the response of two models. We use GPT-4 as the evaluator and report the win rate of MiniCPM and MiniCPM-duplex against GPT-4.

To mimic real-time scenarios, we chunk each instruction from AlpacaEval 2.0 into 4-6 words and feed one chunk 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 4096. For the duplex model, the maximum new token generated per chunk is set 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. Many factors may contribute to a greater response latency. They include the speech-to-text and text-to-speech conversion time, model inference time, network latency, and the interaction strategy that the model utilizes.

[2] https://github.com/tatsu-lab/alpaca_eval

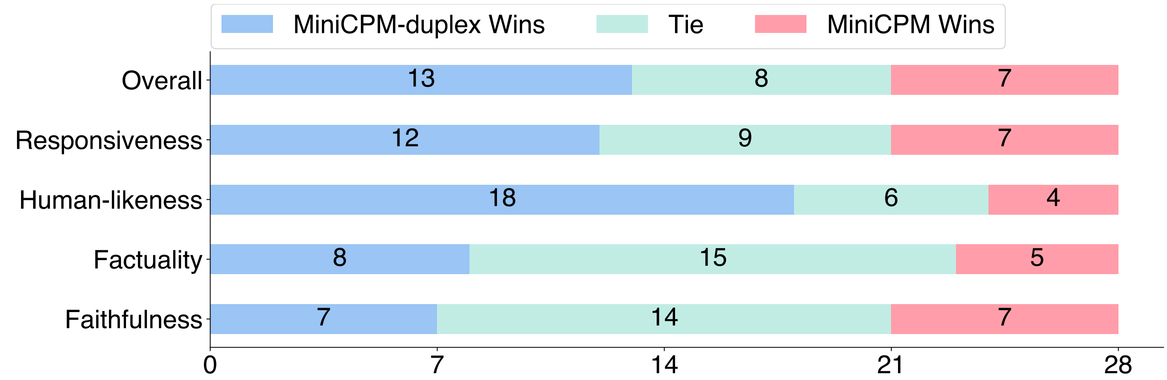

Figure 6: The human evaluation results for MiniCPM and MiniCPM-duplex in terms of response quality, factuality, faithfulness, human-likeness, and overall performance.

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

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

Factuality We also want the assistant to be factual, which is a common metric used in existing works. (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 also 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 API [3].

In the demo, users can choose to interact with the vanilla MiniCPM or our 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 speech-to-text conversion module, before being returned to the user. Meanwhile, the user has to wait until the speech response to done playing before inputting the next query. When interacting with MiniCPM-duplex, the user's speech is being processed every 1.2 seconds [4]. When the MiniCPM-duplex does not generate the idle token, the text generation will be transcribed into audio and then played out. The user voice will be captured, transcribed, and fed to the model regardless of whether the assistant is speaking.

To ensure a fair comparison, we do not disclose what the backbone language model is during interaction.

Human Evaluators Specifically, we recruit 30 participants consisting of 20 males and 10 females from 18 to 35 years old. Each participants hold a Bachelor's or Master's degree. Over 95% of the participants have used LLMs before. About 90% of them have used voice assistants, such as Siri [5], and nearly half of the participants have tried LLM-based voice assistants. Details on employment, payment, and ethical review are in Appendix C.

Before the experiment, we inform all participants that they need to engage in multiple dialogues with two different chat assistants called Model A and Model B, and will be requested to evaluate the experience after the dialogues.

During the experiment, each participant is assigned at least 10 sessions of multi-turn dialogues with each of the models. We do not specify which

[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.

[4] This interval is shorter than the 2-second interval that we used to create the dataset because preliminary tests show that the response latency was too great with 2 seconds.

[5] https://www.apple.com/siri/

Model	Length-Controlled Win Rate	Win Rate	Standard Error	Avg. Length
MiniCPM	3.59	**2.86**	0.58	1337
MiniCPM-duplex	**4.01**	2.24	**0.52**	820

Table 2: AlpacaEval 2.0 results of MiniCPM and MiniCPM-duplex. The baselines are GPT-4. The annotator is also GPT-4.

Model	Faithfulness	Factuality	Human-Likeness	Responsiveness	Overall
MiniCPM	**6.71**	6.46	5.54	6.50	5.29
MiniCPM-duplex	6.61	**6.86**	**6.04**	**7.46**	**6.21**

Table 3: The human evaluation results in faithfulness, factuality, human-likeness, response, and overall. Each metric score ranges from 0 to 10 (the higher the better). Scores are averaged on 28 surveys.

model they should interact with first. To help the participants come up with topics to chat about, we provide them with a reference note containing sample instructions from AlpacaEval 2.0 [6].

After the experiment, participants are asked to fill in a survey to score the two chat assistants.

Survey Design The survey consists of six questions. The first five questions prompt the user to rate the model based on responsiveness, faithfulness, factuality, human-likeness, and overall experience. The answer choices for these questions are scores from 0 to 10, where 0 represents disappointment, 5 represents indifference, and 10 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

GPT-4 Evaluation Results Table 2 shows the GPT-4 evaluation results on AlpacaEval 2.0. It indicates that fine-tuning a pre-trained chat model on Duplex-UltraChat does not significantly harm its performance on general benchmarks. Since MiniCPM has been trained on the UltraChat dataset, the additional training on Duplex-UltraChat does not introduce new abilities or knowledge. This explains why the performance does not improve over the base model.

Human Evaluation Results We have received 30 surveys and discarded two invalid ones, leaving 28 valid samples. Table 3 lists the average scores of both models on five metrics. The duplex model surpasses the base model by 17.39%, 14.77%, 9.03%, and 6.19% on the overall experience, responsiveness, human-likeness, and factuality respectively.

Apart from absolute scores, we compare the ratings of the two models and count the number of evaluators that rate one model higher than the other. The comparison results are shown in Figure 6. The two models come out even on faithfulness, but the duplex model wins in all other aspects, with an exceptionally large margin on 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 existing 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 each body 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 Discussion

There are many unsolved problems to tackle associated with duplex models, and we highlight some important ones below.

High-quality duplex data is urgently needed Existing dialogue datasets are inherently turn-

[6] We drop some complex instructions that are hard to express in words.

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. Six out of the 30 participants pointed out that our duplex model tends to generate long outputs, which may not be appreciated in many dialogues. Therefore, a dataset for practical and complex dialogue situations is of extreme necessity.

A new decoding strategy is needed to improve the chat experience Three participants feel that the duplex model is more likely to interrupt them, which is uncomfortable. How to balance response speed and user experience is an open problem. Furthermore, to be more human-like, the duplex model should learn to start a dialogue or topic 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, which harms the user experience. The cause is that 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.

Apart from the benefits of duplex models, we also consider their potential risks. Misinformation or toxic and harmful speech may be generated. Besides, the duplex model could help some people to commit fraud.

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 GPT-3.5 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 can speak to the chatbot by pressing and holding a button, and releasing it when they are done speaking (OpenAI, 2023a). Then ChatGPT processes the received signal and speaks out its response until it finishes or users interrupt it by pressing a button.

These implementations of voice assistants are inflexible because they require 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 human-likeness, responsiveness, factuality, 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 3.1). These limitations call for a new architecture that better supports the input-output scheme of duplex models.

References

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.

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*.

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*.

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 (Volume 1: Long Papers)*, 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*.

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 *Conference on Empirical Methods in Natural Language Processing*.

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, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nourhan Fahmy, Urvashi Bhattacharyya, W. Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jana Ebert, Tri Dao, Mayank Mishra, Alexander Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean M. Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you! *ArXiv*, abs/2305.06161.

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 (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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

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. Gpt-4 is openai's most advanced system, producing safer and more useful responses. `https://openai.com/gpt-4`.

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

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, Volume 1 (Long Papers)*, 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. *arXiv preprint arXiv:2310.10158*.

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`.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*.

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*.

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. *arXiv preprint arXiv:2305.16291*.

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 (Volume 1: Long Papers)*, 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.

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. “”
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. ""
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 Survey 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 Survey Questions

1. Score the model's response speed to evaluate whether the model can respond to your request quickly.
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. Besides, 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. All those characteristics and experience information collections are permitted by the participants 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.

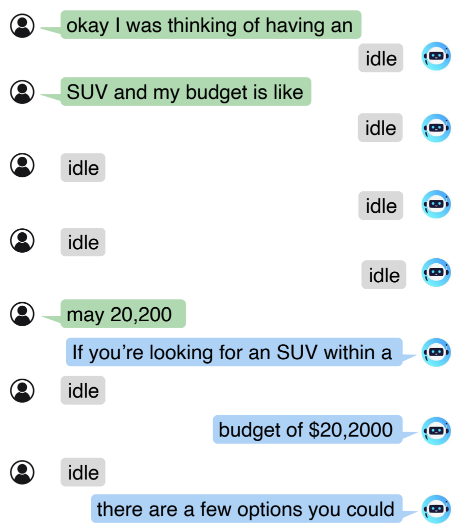

Figure 7: Case A