SpeechRole: A Large-Scale Dataset and Benchmark for Evaluating Speech Role-Playing Agents

Anonymous submission

Figure 1: Distribution of eight sub-tasks across three major task categories in SpeechRole.

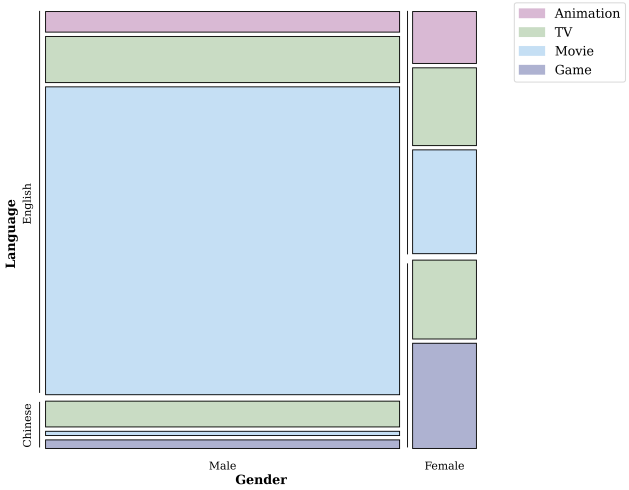

Figure 2: Distribution of the 98 roles in SpeechRole-Data by language, gender, and data source.

A SpeechRole-Data Construction

A.1 Text Data Construction

Role Selection. A total of 98 representative roles are meticulously curated for SpeechRole-Data, selected from two high-quality datasets: ChatHaruhi (Li et al. 2023) and RoleLLM (Wang et al. 2024). Specifically, 18 roles from games, movies, and television series are chosen from ChatHaruhi, while 80 roles from animation, movies, and television series are drawn from RoleLLM. The selected roles are balanced in terms of language, gender, and personality traits, thereby ensuring diversity in both vocal expression and character identity (see Figure 2). To facilitate rich and character-consistent speech modeling, particular emphasis is placed on roles with distinctive vocal characteristics. Table 2 and Table 3 summarizes the attributes of all roles, including their categories, languages, genders, source materials, and dataset splits.

Role Metadata Extraction. For each character, we extract structured metadata, including: **(1) Role Profile,** such as temperament, behavioral tendencies, values, and personal preferences; **(2) Background,** including character settings, social identity, relationships, and living environment; and **(3) Character Lines,** including narration, monologues, and dialogue excerpts involving the target character in games or audiovisual works. These elements collectively define distinctive character profiles and support vivid, context-rich speech role-playing. Additionally, the dialogue excerpts serve as the basis for guiding dialogue generation and evaluating character consistency.

Dialogue Generation. We use `gpt-4.1-2025-04-14` (OpenAI 2023) to generate dialogues between users and roles based on role metadata. For some roles with lengthy scripts, the content is divided into segments and processed sequentially. GPT-4.1 is required to provide relevant script segments while generating dialogues to serve as references for SRPA in answering questions. For each role, 800 single-turn dialogues and 800 multi-turn dialogue data are generated.

Dialogue Filtering. The data are filtered through a multi-stage process to ensure structural consistency, contextual relevance, and semantic diversity. First, only dialogues that conform to a clear turn-taking pattern are retained. In these dialogues, the user initiates with a question, and the role

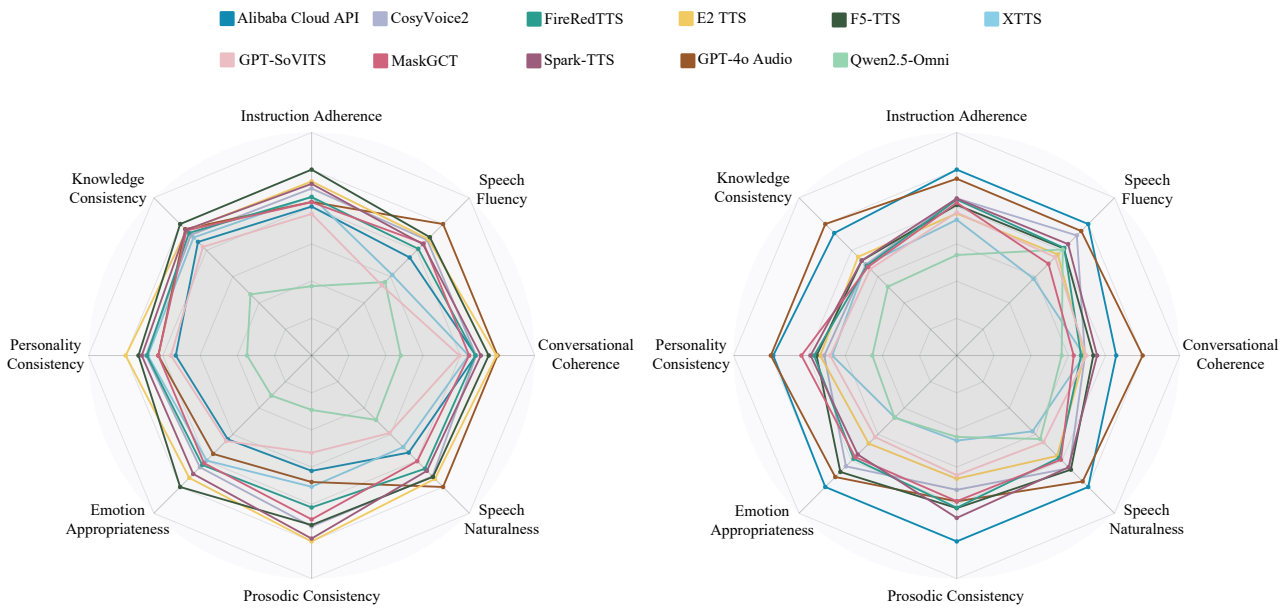

Figure 3: Main evaluation results of 11 SRPAs on SpeechRole-Eval. Left: English results. Right: Chinese results.Each metric is displayed with an interval range of 0.26, with all evaluation results normalized such that the maximum value across metrics is scaled to 1.30.

responds, with each party speaking in alternating turns. To ensure alignment between the dialogue and its corresponding context, semantic embeddings are computed for both texts. For English data, the `all_MiniLM_L6_v2` (Reimers and Gurevych 2019) model is used, and samples with a similarity score below 0.4 are excluded. The same procedure is applied to Chinese data using the `text2vec-bge-large-chinese` (Xu 2023) model, with a similarity score threshold of 0.45.

To further eliminate semantic redundancy, which is commonly observed in GPT-4.1-generated dialogues, near-duplicate samples are removed based on the similarity of their dialogue embeddings. Thresholds are set to 0.85 for Chinese and 0.9 for English. This de-duplication step is conducted separately for single-turn and multi-turn dialogues for each role. After filtering, the dataset contains 53,902 single-turn dialogues and 57,944 multi-turn dialogues.

A.2 Role Voice Collection

Role Voice Extraction. To achieve authentic voice role-playing, real-life performance audio for each character is collected as reference material. For characters from Genshin Impact, in-game character audio is collected with permission from miHoYo. For other characters, audio is obtained from actual film and television works. Films and television series featuring these characters are gathered, and a total of 105 video files with an approximate duration of 181 hours are downloaded from the internet, covering all target characters.

During the preprocessing stage, all extracted speech segments undergo anonymization and fragmentation to ensure that they contain only brief, decontextualized utterances, with no retention of the original video or narrative context. As a result, no identifiable or copyrighted content remains, thus mitigating potential intellectual property risks and ensuring adherence to the principles of fair use for research purposes.

We first employ a batch processing pipeline based on `ffmpeg` (FFmpeg Developers 2024) to extract the raw audio streams from the collected video files. To ensure consistency, all audio is converted to the standard mono 16 kHz WAV format. However, since these recordings often contain background music, noise, and overlapping speech, additional processing is required.

Role Voice Processing. To obtain clean audio segments from specific speakers, we employ a structured pipeline based on the open-source `Emilia` (He et al. 2024) framework. This process includes source separation to remove background noise, speaker separation to distinguish individual speakers, and voice activity detection for fine-grained segmentation. The resulting audio segments, each lasting between 3 and 10 seconds and containing only a single speaker, are well-suited for subsequent modeling.

Finally, we use an automatic speech recognition model to transcribe each segmented audio fragment and evaluate speech quality using the `DNSMOS P.835 OVRL` (Reddy, Gopal, and Cutler 2022) metric. In order to ensure the quality of speech data, we filter out speech with DNMOS less than 3. The resulting structured dataset provides, for each

Metrics	IA	SF	CC	SN	PC	EA	PeC	KC	Overall
MSE	0.1020	0.0413	0.1426	0.0581	0.2539	0.1262	0.1690	0.1147	0.1260
Pearson	0.7304	0.6366	0.6076	0.8093	0.6316	0.8881	0.6677	0.7581	0.7162

Table 1: Evaluation results of correlation between automatic and human scoring.

audio entry, the following information: start and end times, transcription, speaker index, language, and DNSMOS-based audio quality score.

Speaker Identification. The extracted audio data contains only the speaker's index and does not include identity information. To identify the speaker, we use both `gpt-4.1-2025-04-14` and `DeepSeek-V3-0324` (DeepSeek-AI et al. 2024) models to determine the target character's speaker number based on all transcribed texts. We integrate the results from both models and verify them manually to obtain the speech corresponding to the target character.

Role Voice Ranking. To select the most representative audio sample as a reference for subsequent role-playing, we rank all speech samples of each speaker based on similarity. Specifically, we extract the speaker feature vector of each audio sample using the `CAM++` (Wang et al. 2023) model and calculate the cosine similarity between each sample and all other samples. A higher average similarity score indicates that the audio sample better represents the overall speaker characteristics of the role. Finally, we sort all audio samples for each role in descending order according to their similarity scores and select the top-ranked sample as the reference audio.

B Human Correlation Verification Details

To evaluate the consistency between automatic scoring and human scoring, we randomly selected 20 samples from the dataset, including 5 single-turn dialogue samples and 15 multi-turn dialogue samples. For each sample, we randomly selected two model-generated speech responses for evaluation.

Gemini scored each model response and its corresponding reference response on a scale of 1 to 10 based on 8 predefined scoring metrics. The final score for a model was calculated as the ratio of the model response score to the reference response score.

For human evaluation, four trained evaluators were employed to directly compare the responses of the two models based on the same 8 scoring metrics. The evaluation criteria were as follows: the superior model received 1 point, the inferior model 0 points, and if the performance was deemed equivalent, each received 0.5 points. The final human score was determined by averaging the scores from all evaluators.

To quantify the agreement between automated and human evaluation methods, we computed two key metrics: mean squared error (MSE) and Pearson correlation coefficient. The MSE was calculated by first fitting a linear regression between the score differences of paired model responses from human evaluators and gemini. The predicted gemini score differences were then derived from this regression model, and the MSE was computed between these predicted values and the actual gemini score differences. For the Pearson correlation analysis, we computed the linear correlation coefficient between the human evaluators' score differences and gemini's score differences across all paired comparisons.

Table 1 reports the detailed results, the overall MSE of 0.1260 indicates relatively minor deviations between the two scoring methods. The high Pearson correlation coefficient further confirms the reliability of the automated scoring system. The evaluation results collectively demonstrate strong agreement between automatic and human scoring across all metrics, indicating that the automated scoring system can effectively approximate human judgment.

C Additional data statistics

Role Type Distribution Figure 2 presents the distribution of characters by language (Chinese and English), source type (game, film, television series, and animation), and gender. English characters, particularly those from films, constitute the majority in the dataset. In contrast, the Chinese subset exhibits a more balanced distribution across various source types. This diversity provides extensive coverage of voices and contexts for role-playing.

Task Type Distribution User prompts can generally be categorized into three major task categories, which can be further divided into eight subtask types, as illustrated in the figure 1. The description and proportion of each subtask type are as follows:

1. Internal Reasoning
   - Opinion and Emotion Inquiry: Questions about opinions, emotions, inner feelings, or reactions. (14.73%)
   - Reasoning and Motivation: Questions about why someone did something, their motivations, or decision-making processes. (14.73%)
   - Reflection and Change: Questions about how practices/opinions have changed. (9.55%)
2. Experiential Narration
   - Action Description and Retrospection: Questions about what happened, how something was done, or detailed retrospectives. (14.72%)
   - Skills and Abilities: Questions about how to accomplish something, difficulties with skills, or ability changes. (6.38%)
   - Historical and Background Description: Questions about history or background of places/organizations/people. (11.55%)

3. Social Communication
    - Judgment of Others and Events: Questions about relationships, event evaluations, or interpersonal interactions. (14.73%)
    - Advice and Life Experience: Questions seeking advice, coping methods, or life experiences. (13.61%)

Evaluation Results Demonstration Figure 3 visually presents the evaluation results of all SRPAs on the SpeechRole-Eval benchmark.

D Prompt Templates

Prompts for Batch Data Generation Figure 4 and Figure 5 respectively present the prompts used for generating single-turn dialogue data for English and Chinese characters. Figure 6 and Figure 7 respectively present the prompts used for generating multi-turn dialogue data for English and Chinese characters.

Prompts for Automated Judgement Figure 8 displays the judgment prompt designed for automatically assessing audio responses generated by the two models.

E SpeechRole-Agent training settings

Figure 9 presents the distributed training script for fine-tuning the Qwen2.5-Omni-7B model on 8×H20 GPUs. The configuration employs bfloat16 mixed-precision training with a learning rate of 1e-4 and gradient accumulation. The implementation includes periodic evaluation (every 500 steps) and checkpoint management.

References

DeepSeek-AI; Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; Dai, D.; Guo, D.; Yang, D.; Chen, D.; Ji, D.; Li, E.; Lin, F.; Dai, F.; Luo, F.; Hao, G.; Chen, G.; Li, G.; Zhang, H.; Bao, H.; Xu, H.; Wang, H.; Zhang, H.; Ding, H.; Xin, H.; Gao, H.; Li, H.; Qu, H.; Cai, J. L.; Liang, J.; Guo, J.; Ni, J.; Li, J.; Wang, J.; Chen, J.; Chen, J.; Yuan, J.; Qiu, J.; Li, J.; Song, J.; Dong, K.; Hu, K.; Gao, K.; Guan, K.; Huang, K.; Yu, K.; Wang, L.; Zhang, L.; Xu, L.; Xia, L.; Zhao, L.; Wang, L.; Zhang, L.; Li, M.; Wang, M.; Zhang, M.; Zhang, M.; Tang, M.; Li, M.; Tian, N.; Huang, P.; Wang, P.; Zhang, P.; Wang, Q.; Zhu, Q.; Chen, Q.; Du, Q.; Chen, R. J.; Jin, R. L.; Ge, R.; Zhang, R.; Pan, R.; Wang, R.; Xu, R.; Zhang, R.; Chen, R.; Li, S. S.; Lu, S.; Zhou, S.; Chen, S.; Wu, S.; Ye, S.; Ye, S.; Ma, S.; Wang, S.; Zhou, S.; Yu, S.; Zhou, S.; Pan, S.; Wang, T.; Yun, T.; Pei, T.; Sun, T.; Xiao, W. L.; and Zeng, W. 2024. DeepSeek-V3 Technical Report. *CoRR*, abs/2412.19437.

FFmpeg Developers. 2024. FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video. https://ffmpeg.org/.

He, H.; Shang, Z.; Wang, C.; Li, X.; Gu, Y.; Hua, H.; Liu, L.; Yang, C.; Li, J.; Shi, P.; Wang, Y.; Chen, K.; Zhang, P.; and Wu, Z. 2024. Emilia: An Extensive, Multilingual, and Diverse Speech Dataset For Large-Scale Speech Generation. In *IEEE Spoken Language Technology Workshop, SLT 2024, Macao, December 2-5, 2024*, 885–890. IEEE.

Li, C.; Leng, Z.; Yan, C.; Shen, J.; Wang, H.; Mi, W.; Fei, Y.; Feng, X.; Yan, S.; Wang, H.; Zhan, L.; Jia, Y.; Wu, P.; and Sun, H. 2023. ChatHaruhi: Reviving Anime Character in Reality via Large Language Model. *CoRR*, abs/2308.09597.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Reddy, C. K. A.; Gopal, V.; and Cutler, R. 2022. Dnsmos P.835: A Non-Intrusive Perceptual Objective Speech Quality Metric to Evaluate Noise Suppressors. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, 886–890. IEEE.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 3980–3990. Association for Computational Linguistics.

Wang, H.; Zheng, S.; Chen, Y.; Cheng, L.; and Chen, Q. 2023. CAM++: A Fast and Efficient Network for Speaker Verification Using Context-Aware Masking. In Harte, N.; Carson-Berndsen, J.; and Jones, G., eds., *24th Annual Conference of the International Speech Communication Association, Interspeech 2023, Dublin, Ireland, August 20-24, 2023*, 5301–5305. ISCA.

Wang, N.; Peng, Z.; Que, H.; Liu, J.; Zhou, W.; Wu, Y.; Guo, H.; Gan, R.; Ni, Z.; Yang, J.; Zhang, M.; Zhang, Z.; Ouyang, W.; Xu, K.; Huang, W.; Fu, J.; and Peng, J. 2024. RoleLLM: Benchmarking, Eliciting, and Enhancing Role-Playing Abilities of Large Language Models. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, 14743–14777. Association for Computational Linguistics.

Xu, M. 2023. text2vec: A Tool for Text to Vector.

Role Name	Category	Language	Gender	Source	Split
hutao	Game	Chinese	Female	Genshin Impact	train/test
raidenShogun	Game	Chinese	Female	Genshin Impact	train/test
wanderer	Game	Chinese	Male	Genshin Impact	train/test
ayaka	Game	Chinese	Female	Genshin Impact	dev/test
zhongli	Game	Chinese	Male	Genshin Impact	train/test
liyunlong	TV	Chinese	Male	Drawing Sword	dev/test
wangduoyu	Movie	Chinese	Male	Hello Mr. Billionaire	train/test
weixiaobao	TV	Chinese	Male	The Deer and the Cauldron	train/test
jiumozhi	TV	Chinese	Male	Demi-Gods and Semi-Devils	train/test
wangyuyan	TV	Chinese	Female	Demi-Gods and Semi-Devils	train/test
Luna	Movie	English	Female	Harry Potter	dev/test
Penny	TV	English	Female	The Big Bang Theory	dev/test
zhangwuji	TV	Chinese	Male	The Heaven Sword and Dragon Saber	train/test
zhaomin	TV	Chinese	Female	The Heaven Sword and Dragon Saber	train/test
huangrong	TV	Chinese	Female	The Legend of the Condor Heroes	train/test
guojing	TV	Chinese	Male	The Legend of the Condor Heroes	dev/test
wukong	TV	Chinese	Male	Journey to the West	train/test
HAL 9000	Movie	English	Male	2001: A Space Odyssey	train/test
Colonel Nathan R. Jessep	Movie	English	Male	A Few Good Men	train/test
Antonio Salieri	Movie	English	Male	Amadeus	train/test
Stifler	Movie	English	Male	American Pie	train/test
Paul Vitti	Movie	English	Male	Analyze That	train/test
Alvy Singer	Movie	English	Male	Annie Hall	train/test
Violet Weston	Movie	English	Female	August: Osage County	train/test
Willie Soke	Movie	English	Male	Bad Santa	train/test
Gaston	Animation	English	Male	Beauty and the Beast	train/test
The Dude	Movie	English	Male	The Big Lebowski	train/test
Paul Conroy	Movie	English	Male	Buried	train/test
Truman Capote	Movie	English	Male	Capote	train/test
Mater	Animation	English	Male	Cars 2	train/test
Andrew Detmer	Movie	English	Male	Chronicle	train/test
Coriolanus	Movie	English	Male	Coriolanus	train/test
John Keating	Movie	English	Male	Dead Poets Society	dev/test
Wade Wilson	Movie	English	Male	Deadpool	dev/test
Jim Morrison	Movie	English	Male	The Doors	train/test
Queen Elizabeth I	Movie	English	Female	Elizabeth: The Golden Age	dev/test
Jeff Spicoli	Movie	English	Male	Fast Times at Ridgemont High	train/test
Fred Flintstone	Animation	English	Male	The Flintstones	train/test
Freddy Krueger	Movie	English	Male	Freddy Vs.Jason	train/test
Tyrion Lannister	TV	English	Male	Game of Thrones	train/test
James Brown	Movie	English	Male	Get on Up	train/test
Walt Kowalski	Movie	English	Male	Gran Torino	train/test
John Coffey	Movie	English	Male	The Green Mile	train/test
Theodore Twombly	Movie	English	Male	Her	dev/test
Gregory House	TV	English	Male	House M.D.	dev/test
Sonny	Movie	English	Male	I, Robot	train/test
Colonel Hans Landa	Movie	English	Male	Inglourious Basterds	train/test
Judge Dredd	Movie	English	Male	Judge Dredd	dev/test
Juno MacGuff	Movie	English	Female	Juno	train/test
Professor G.H. Dorr	Movie	English	Male	The Ladykillers	train/test

Table 2: Summary of character attributes and dataset partitioning (part 1).

Role Name	Category	Language	Gender	Source	Split
Fletcher Reede	Movie	English	Male	Liar Liar	train/test
Abraham Lincoln	Movie	English	Male	Lincoln	train/test
Frank T.J. Mackey	Movie	English	Male	Magnolia	train/test
Leonard Shelby	Movie	English	Male	Memento	train/test
Harvey Milk	Movie	English	Male	Milk	train/test
Randle McMurphy	Movie	English	Male	One Flew Over the Cuckoo's Nest	train/test
Jack Sparrow	Movie	English	Male	Pirates of the Caribbean: Dead Man's Chest	dev/test
John Dillinger	Movie	English	Male	Public Enemies	train/test
Lestat de Lioncourt	Movie	English	Male	The Queen of Damned	train/test
Tyler Hawkins	Movie	English	Male	Remember Me	dev/test
James Carter	Movie	English	Male	Rush Hour 2	train/test
Jigsaw	Movie	English	Male	Saw	train/test
John Doe	Movie	English	Male	Se7en	train/test
Sherlock Holmes	Movie	English	Male	Sherlock Holmes	dev/test
Shrek	Animation	English	Male	Shrek	train/test
Pat Solitano	Movie	English	Male	Silver Linings Playbook	train/test
Karl Childers	Movie	English	Male	Sling Blade	train/test
Bruno Antony	Movie	English	Male	Strangers on a Train	train/test
Seth	Movie	English	Male	Superbad	train/test
Caden Cotard	Movie	English	Male	Synecdoche, New York	train/test
Travis Bickle	Movie	English	Male	Taxi Driver	train/test
Stanley Ipkiss	Movie	English	Male	The Mask	dev/test
Lyn Cassady	Movie	English	Male	The Men Who Stare at Goats	train/test
Michael Scott	TV	English	Male	The Office	dev/test
Robert Angier	Movie	English	Male	The Prestige	dev/test
Dr. Frank-N-Furter	Movie	English	Male	The Rocky Horror Picture Show	train/test
Jack Torrance	Movie	English	Male	The Shining	train/test
Tom Ripley	Movie	English	Male	The Talented Mr. Ripley	train/test
D_Artagnan	Movie	English	Male	The Three Musketeers	train/test
Thor	Movie	English	Male	Thor: Ragnarok	train/test
James Bond	Movie	English	Male	Tomorrow Never Dies	dev/test
Mark Renton	Movie	English	Male	Trainspotting	train/test
David Aames	Movie	English	Male	Vanilla Sky	train/test
Rorschach	Movie	English	Male	Watchmen	train/test
Jordan Belfort	Movie	English	Male	The Wolf of Wall Street	train/test
Logan	Movie	English	Male	X-Men Origins: Wolverine	dev/test
Judy Hoops	Animation	English	Female	Zootopia	train/test
Doctor Who	TV	English	Male	Doctor Who	train/test
Raylan Givens	TV	English	Male	Justified	train/test
Mary Sibley	TV	English	Female	Salem	train/test
Lucifer Morningstar	TV	English	Male	Lucifer	train/test
Twilight Sparkle	Animation	English	Female	My Little Pony: Friendship is Magic	dev/test
Oliver Queen	TV	English	Male	Arrow	train/test
Klaus Mikaelson	TV	English	Male	The Originals	train/test
Queen Catherine	TV	English	Female	Reign	train/test
Dr. Hannibal Lecter	TV	English	Male	Hannibal	train/test
Coach Eric Taylor	Movie	English	Male	Friday Night Lights	train/test
yaemiko	Game	Chinese	Female	Genshin Impact	train/test

Table 3: Summary of character attributes and dataset partitioning (part 2).

Prompt for Batch Generation of Single-Turn Data, English

System: Your task is to generate {BATCH_SIZE} single-turn dialogues between a user and the character {role_name} from the script {script_name}. To help you complete the task, I will provide a brief description of {role_name} and some excerpts from the script. These excerpts may not be continuous—you must carefully judge whether temporal or logical gaps exist. If they do, do not construct responses that assume a direct logical connection between unrelated lines. Please follow the instructions below strictly:
1. Each dialogue should consist of a single Q&A turn, forming a natural, in-character, and context-aware role-playing interaction.
2. The dialogue should be in the following format:

```
{{
    "user": "User's question",
    "role": "Character's response"
}}
```

3. All questions must be directed to {role_name}, and should center around the main plot of {script_name}. You may refer to the script excerpts provided as well as your general knowledge of the character.
4. All outputs must be formatted as a JSON array. Each dialogue should be an object with two fields:
   - dialogue: a list containing exactly one Q&A pair;
   - context: the script excerpts that were used or referenced in generating this dialogue.
5. You must generate {BATCH_SIZE} such dialogues.
6. All character responses must remain faithful to their personality and the context of the story.
7. Ensure each question and answer brings unique value, avoiding repetitive content across samples.
Now I will give you one example. This example is not from {script_name} and not about {role_name}, but serves to illustrate the format and completeness of a single-turn dialogue:
[Example]

```
[
    {{
    "dialogue": [
        {{
        "user": "Tony, why did you stop manufacturing weapons even though it made your company so successful?",
        "role": "Because I saw firsthand what my weapons were doing in the wrong hands. That ambush in Afghanistan—it changed everything. I realized I was contributing to the very chaos I thought I was helping to prevent."
        }}
    ],
    "context": "TONY STARK: I saw young American soldiers killed by the very weapons I created to protect them. I can't ignore that.\n(Flashback: Tony being ambushed by Stark Industries missiles in Afghanistan)\nTONY: I'm not a hero. I'm just trying to fix what I broke."
    }}
]
```

Prompt:
[Character Name and Description]
The character is {role_name} from {script_name}, the character description is: {role_description}
[Script Content]
{script}
[JSON format]

```
[
    {{
    "dialogue": [
        {{
        "user": "",
        "role": ""
        }}
    ]
    "context": "",
    }},
    ......
]
```

[Question Design ({BATCH_SIZE} single-turn dialogues, no semantic repetition, all questions must be directed to {role_name}; each dialogue must consist of exactly 1 user-role interaction pair, and each Q&A must be self-contained, logically sound, and in-character)]

Figure 4: Prompt structure for batch single-turn English dialogue generation.

Prompt for Batch Generation of Single-Turn Data, Chinese

System: 你的任务是设计{BATCH_SIZE}段用户与{script_name}中的{role_name}之间的单轮对话。为了帮助你更好地完成任务，我会提供{role_name}的简要人物描述和部分剧本内容。这些剧本片段可能不连续，你需要根据上下文判断是否存在时间或逻辑跳跃。如果不连续，请不要构建上下句之间有强逻辑衔接的对话。
请严格遵循以下设计规则：
1. 每段对话应为一轮完整问答，由用户向{role_name}提问，角色作答，形成自然、符合剧情背景的角色扮演对话。
2. 每轮结构为：
{{
"user": "用户提的问题",
"role": "角色的回答"
}}
3. 所有问题必须是用户直接向{role_name}提问，并围绕{script_name}的核心情节展开。你可以结合剧本片段和你对角色的常识知识设计问题。
4. 所有输出必须是一个符合 JSON 结构的数组，每段对话为一个对象，对象中包含两个字段：
- dialogue：一个包含一组问答的列表（即仅一个 user-role 对）；
- context：该段对话中所参考或引用的剧本原文内容。
5. 你需要输出共{BATCH_SIZE}段对话。
6. 所有角色回答必须符合人设，语言风格需贴合剧本风格与时代背景。
7. 对话应尽量避免重复提问或答复，确保每段都有独立的信息价值。
接下来我会给你一个样例，这个样例与{role_name}和{script_name}无关，仅用于展示格式和完整性：
[样例]
[
{{
"dialogue": [
{{
"user": "姜子牙，你为什么要讨伐商纣王？这是一个冒天下之大不韪的决定。",
"role": "我并非一时冲动起兵。商纣王荒淫无道，沉迷酒色，任用奸佞，焚书坑儒，滥杀忠良，百姓民不聊生。作为一名辅佐天下的臣子，我若对他的暴行视而不见，那才是失职。讨伐他，并非为我一人之私，而是为了拯救黎民百姓于水火。"
}}
],
"context": "姜子牙：「纣王以鹿台为宫，役民无数，掠财千金，尽为享乐。我观民间疲敝，不忍再等。」\n姬发：「父王身死国灭，子民苦难，我夜不能寐。师父，若你不举兵，我愿亲率义军，誓为天下请命。」\n 姜子牙沉思片刻：「兵者不祥之器，非不得已不得动之。然今日不动，百姓将无明日。此战，虽难，却势在必行。」
"
}}
]

Prompt:
[Character Name and Description]
The character is {role_name} from {script_name}, the character description is: {role_description}
[Script Content]
{script}
[JSON format]
[
{{
"dialogue": [
{{
"user": "",
"role": ""
}}
]
"context": "",
}},
.....
]
[Question Design ({BATCH_SIZE} single-turn dialogues, no semantic repetition, all questions must be directed to {role_name}; each dialogue must consist of exactly 1 user-role interaction pair, and each Q&A must be self-contained, logically sound, and in-character)]

Figure 5: Prompt structure for batch single-turn Chinese dialogue generation.

Prompt for Batch Generation of Multi-Turn Data, English

System: Your task is to generate {BATCH SIZE} multi-turn dialogues between a user and the character {role name} from the script {script name}. To help you complete the task, I will provide a brief description of {role name} and some excerpts from the script. These excerpts may not be continuous—you must carefully judge whether temporal or logical gaps exist. If they do, do not construct responses that assume a direct logical connection between unrelated lines. Please follow the instructions below strictly:
1. Each dialogue should consist of 2 to 4 turns of Q&A, forming a natural, in-character, and context-aware role-playing conversation.
2. Each turn should be in the following format, alternating between user questions and character answers:

```
{{
    "user": "User's question",
    "role": "Character's response"
}}
```

3. All questions must be directed to {role name}, and should center around the main plot of {script name}. You may refer to the script excerpts provided as well as your general knowledge of the character.
4. All outputs must be formatted as a JSON array. Each dialogue should be an object with two fields:
- dialogue: a list of Q&A turns in order;
- context: the script excerpts that were used or referenced in generating this dialogue.
5. You must generate {BATCH SIZE} such dialogues.
6. The number of turns per dialogue can vary (2–4), but the exchange must be coherent and the responses must stay in character.
7. Aim to produce 3 or 4 turns per dialogue when appropriate, as long as the flow remains logical and engaging.
Now I will give you one example. This example is not from {script name} and not about {role name}, but serves to illustrate the format and definition of completeness.
[Example]

```
[
    {{
    "dialogue": [
        {{
        "user": "Tony, why did you stop manufacturing weapons even though it made your company so successful?",
        "role": "Because I saw firsthand what my weapons were doing in the wrong hands. That ambush in Afghanistan—it changed everything. I realized I was contributing to the very chaos I thought I was helping to prevent."
        }},
        ......
    ],
    "context": "TONY STARK: I saw young American soldiers killed by the very weapons I created to protect them. I can't ignore that.\nRHODEY: You're not just a weapons manufacturer anymore.\n ....."
    }}
]
```

Prompt:
[Character Name and Description]
The character is {role name} from {script name}, the character description is: {role description}
[Script Content]
{script}
[JSON format]

```
[
    {{
    "dialogue": [
        {{
        "user": "",
        "role": ""
        }},
        ......
    ]
    "context": "",
    }},
    ......
]
```

[Question Design ({BATCH_SIZE} multi-turn dialogues, no semantic repetition, all questions must be directed to {role_name}; each dialogue must consist of 2 to 4 user-role interaction rounds, and you are encouraged to generate 3 or 4 rounds whenever possible, as long as the continuation remains logical and in-character)]

Figure 6: Prompt structure for batch multi-turn English dialogue generation.

Prompt for Batch Generation of Multi-Turn Data, Chinese

System: 你的任务是设计{BATCH_SIZE}段用户与{script_name}中的{role_name}之间的多轮对话。为了帮助你更好地完成任务，我会提供{role_name}的简要人物描述和部分剧本内容。这些剧本片段可能不连续，你需要根据上下文判断是否存在时间或逻辑跳跃。如果不连续，请不要构建上下句之间有强逻辑衔接的对话。
请严格遵循以下设计规则：
1.每段对话应为一组 2 到 4 轮的连续问答，形成自然的、符合剧情背景的角色扮演对话。
2.对话应由用户提问、角色回答交替构成，每轮结构为：
{{
"user": "用户提的问题",
"role": "角色的回答"
}}
3.请确保所有问题都是用户直接向{role_name}提问，并围绕{script_name}的核心情节展开。你可以结合剧本片段和你对角色的常识知识设计问题。
4. 所有输出必须是一个符合 JSON 结构的数组，每段对话为一个对象，对象中包含两个字段：
- dialogue：一个按轮次排列的问答列表；
- context：该段对话中所参考或引用的剧本原文内容。
5.你需要输出共{BATCH_SIZE}段对话。
6.每段对话的轮次不必相同（可为 2-4 轮），但都应逻辑自然，角色回答应符合人设。
7. 请尽量生成 3 或 4 轮对话，只要情节自然、角色一致即可，不必刻意缩短对话。
接下来我会给你一个样例，这个样例与{role_name}和{script_name}无关，主要作用是让你明白完整性的定义以及输出的格式。
[样例]
[
{{
"dialogue": [
{{
"user": "姜子牙，你为什么要讨伐商纣王？这是一个冒天下之大不韪的决定。",
"role": "我并非一时冲动起兵。商纣王荒淫无道，沉迷酒色，任用奸佞，焚书坑儒，滥杀忠良，百姓民不聊生。作为一名辅佐天下的臣子，我若对他的暴行视而不见，那才是失职。讨伐他，并非为我一人之私，而是为了拯救黎民百姓于水火。"
}},
.....
],
"context": "姜子牙：「纣王以鹿台为宫，役民无数，掠财千金，尽为享乐。我观民间疲敝，不忍再等。」\n姬发：「父王身死国灭，子民苦难，我夜不能寐。师父，若你不举兵，我愿亲率义军，誓为天下请命。」\n....."
}}
]

Prompt:
[Character Name and Description]
The character is {role name} from {script name}, the character description is: {role description}
[Script Content]
{script}
[JSON format]
[
{{
"dialogue": [
{{
"user": "",
"role": ""
}},
.....
]
"context": "",
}},
.....
]
[Question Design ({BATCH_SIZE} multi-turn dialogues, no semantic repetition, all questions must be directed to {role_name}; each dialogue must consist of 2 to 4 user-role interaction rounds, and you are encouraged to generate 3 or 4 rounds whenever possible, as long as the continuation remains logical and in-character)]

Figure 7: Prompt structure for batch multi-turn Chinese dialogue generation.

Prompt for Automated Judgement

System:
You are a helpful assistant.

Prompt:
[Question Start]\n\n{question}\n\n## **[Question End]**\n\n\n## **[Model A's response start, with one answer audio in each round]**

[Audio Responses of Model A]

[Model A's Response End]\n\n\n## **[Model B's response start, with one answer audio in each round]**

[Audio Responses of Model B]

[Model B's Response End]\n\n\n## **[Instruction]**\n\nThe task instruction of the two models is to directly role-play as {role name}.\n\nPlease evaluate the following aspect of each model's response:\n{metrics[metric]}

Please provide a brief qualitative evaluation for the relative performance of the two models, followed by paired quantitative scores from 1 to 10, where 1 indicates poor performance and 10 indicates excellent performance.\n\nThe output should be in the following format:\nQualitative Evaluation, [Scores]: ({the score of Model A}, {the score of Model B})\n\nPlease ensure that your evaluations are unbiased and that the order in which the responses were presented does not affect your judgment.

Figure 8: Prompt structure for Automated Judgement.

Script Configuration for SpeechRole-Agent Training

```
OUTPUT_DIR="checkpoint"
MODEL_PATH="Qwen2.5-Omni-7B"

export MODELSCOPE_CACHE="ms_cache"
mkdir -p $MODELSCOPE_CACHE

torchrun --nproc_per_node=8 swift sft \
    --model "$MODEL_PATH" \
    --train_type lora \
    --dataset 'train.jsonl' \
    --torch_dtype bfloat16 \
    --num_train_epochs 1 \
    --per_device_train_batch_size 2 \
    --per_device_eval_batch_size 1 \
    --learning_rate 1e-4 \
    --lora_rank 8 \
    --lora_alpha 32 \
    --target_modules all-linear \
    --gradient_accumulation_steps 1 \
    --eval_steps 500 \
    --save_steps 500 \
    --save_total_limit 2 \
    --logging_steps 5 \
    --output_dir "$OUTPUT_DIR" \
    --warmup_ratio 0.05 \
    --dataloader_num_workers 4 \
    &> train.log
```

Figure 9: Training script configuration for fine-tuning the Qwen2.5-Omni-7B model.