TMCSpeech: A Chinese TV and movie speech dataset with character descriptions and a character-based voice generation model

Dong Liu[1], Yueqian Lin[1], Yunfei Xu[2], and Ming Li[1] ⋆

[1] Suzhou Municipal Key Laboratory of Multimodal Intelligent Systems,
Data Science Research Center, Duke Kunshan University, Kunshan, China
[2] Guangdong OPPO Mobile Telecommunications Corp., Ltd., China

Abstract. Recent research on text-guided speech synthesis has sparked considerable interest. This study explores the potential of leveraging publicly available internet video data for speech synthesis and character-based new voice generation. We introduce a multi-modal extraction pipeline for automating the creation of speech synthesis datasets, extracting accurate character speech segments and descriptions from online videos. Additionally, we propose a person-description-based controllable voice synthesis system, establishing a mapping from character descriptions to speaker representation vectors. This system transforms character descriptions into new vectors, serving as input for zero-shot VITS to generate character-specific voices. Both objective and subjective metrics affirm our approach's capability to generate previously unheard character-specific voices with acceptable naturalness. We plan to release the annotation set of TMCSPEECH [3]. Our audio samples can be accessed online [4].

Keywords: Multi-modal data processing, speech synthesis, voice generation, zero-shot text-to-speech

1 Introduction

With the rapid development of deep learning, the quality of speech synthesis has significantly improved [27]. However, current mainstream speech synthesis systems still face substantial limitations. Though they achieve high-quality synthesis for speakers within the training set, for speakers not included in the training set, synthesis quality frequently fails to meet expectations. In the thriving era of Artificial Intelligence Generated Content (AIGC), there is a growing demand for personalized voice generation, especially in scenarios such as audiobooks where each character has unique traits. Specifically, in audiobooks, current systems struggle to provide voices that match the unique characteristics of each character. For example, characters with cautious or arrogant personalities

⋆ Corresponding Author, E-mail: ming.li369@dukekunshan.edu.cn

[3] We only provide our collected original video links and our annotated labels for non-commercial research purposes. Our shared annotation set does not contain any audio or video data. It is the user's responsibility to decide whether to download the video data and whether their intended purpose with the downloaded data is allowed in their country.

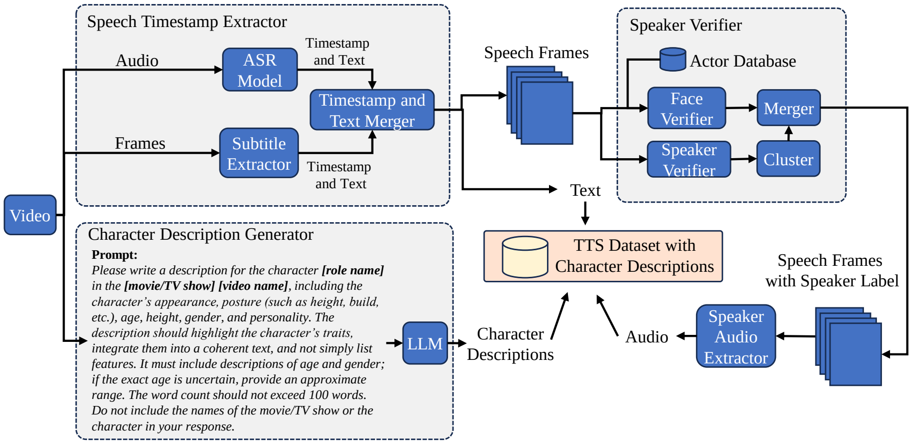
[4] https://raydonld.github.io/TMCSPEECH/

are often not accurately represented in synthesized speech, as current systems fail to capture these nuances. Therefore, the current audiobook reading experience often suffers from a lack of variety in voices, making it challenging for listeners to distinguish between characters solely based on voice, thereby reducing immersion and comprehension. In addition, recording real voices for each character in a novel, tailored to their characteristics, is impractical and economically inefficient, as it would significantly increase data collection costs and fail to meet the rapidly expanding content demands. Consequently, synthesizing voices that align with character descriptions in audiobooks efficiently becomes the focus of this research.

Recent efforts have focused on synthesizing voices absent from the training set. For example, Stanton et al. [26] proposed Tacospawn, which utilizes Mixture Density Networks (MDN) to infer the conditional distribution of the voice representation of multiple speakers in a Tacotron model under discrete label conditions, achieving the conditional generation of new voices. Bilinski et al [2]. implemented Tacospawn in a Flow-TTS based system, confirming the efficacy of GMM-based methods in characterizing voice representation distributions. However, these methods use discrete labels, and Gaussian Mixture Models have limited capabilities in modeling category boundaries, resulting in weak controllability. Recently, an increasing number of works have applied prompts to audio generation [19], [13] and style-controlled speech synthesis [16], [28] and [18]. Inspired by these works, using Prompt descriptions instead of discrete variables as parameters for voice-controllable descriptions can greatly enhance the freedom of voice generation. PromptTTS [10], proposed by Guo et al., introduced text descriptions of sound into TTS for the first time, achieving the generation of speech that meets specific requirements through natural language descriptions of sound. However, this method is not suitable for controllable voice generation, as there is a one-to-many relationship between the text description and the voice. PromptSpeaker [29], proposed by Zhang et al., introduced Glow [17] into a zero-shot VITS [15] to establish a mapping relationship between Speaker representation and Semantic representation and used Prompt Encoder to align Prompt descriptions with the Semantic Representation distribution.

Despite the achievements of the above methods in describing sound using Prompt, they all require manual annotation of audio data. This greatly limits the scale of the voice dataset with Prompt text descriptions, thereby restricting the effectiveness of controllable voice generation. Specifically, in the task of audiobooks, we need to customize the voice according to the appearance, facial features, and personality traits of the characters. The above methods are not well-suited to this requirement as they do not focus on descriptions of characters' appearances and personalities, and there is currently a lack of audiobook speech synthesis datasets with character descriptions. Therefore, the primary challenge in contemporary research lies in acquiring speech synthesis databases with detailed character descriptions and developing systems that leverage these descriptions.

To address these research challenges, we have developed an automatic multi-modal extraction pipeline and a controllable voice synthesis system based on character text descriptions. The purpose of this pipeline is to accurately extract the speech audio of each character from videos and simultaneously generate corresponding character de-



Prompt:
Please write a description for the character ***[role name]*** *in the* ***[movie/TV show] [video name]****, including the character's appearance, posture (such as height, build, etc.), age, height, gender, and personality. The description should highlight the character's traits, integrate them into a coherent text, and not simply list features. It must include descriptions of age and gender; if the exact age is uncertain, provide an approximate range. The word count should not exceed 100 words. Do not include the names of the movie/TV show or the character in your response.*

Fig. 1. The overview of our multi-modal extraction pipeline.

scriptions. This pipeline consists of four components: the speech timestamp extractor, speaker verifier, speaker audio extractor, and character description generator. Use this pipeline to process abundant online video resources, substantially reducing data collection and annotation costs. Our proposed controllable voice synthesis system, grounded in character descriptions, encompasses four primary components: a pretrained large language model, a prompt encoder, a normalizing flow model, and a zero-shot TTS system. This system can generate speaker voices that match the feature descriptions based on character traits.

2 Methods

2.1 Automatic Multi-modal Extraction Pipeline

As illustrated in Figure. 1, the multi-modal extraction pipeline consists of a speech timestamp extractor, speaker verifier, speaker audio extractor, and character description generator. The speech timestamp extractor initially isolates timestamps and corresponding texts of speech segments from video data. Next, video segments linked to these timestamps undergo analysis by the speaker verifier, which ascertains the speaker's identity by correlating facial and voice features within each segment. The extracted audio from the video is then processed by the speaker audio extractor to remove background noise and perform noise reduction, selecting high-quality audio. Lastly, the character description generator takes the roles and their corresponding video names from the video, generating character descriptions that match the characteristics of the characters. Detailed descriptions of each component within this pipeline are presented in subsequent sections.

Step 1: Speech Timestamp Extractor
The Speech Timestamp Extractor module accurately extracts timestamps and corresponding textual content from speech segments in video data. To achieve this, we employ the Paraformer-large speech recognition method [9], specifically designed for efficient extraction of speech timestamps and text from lengthy audio. Furthermore,

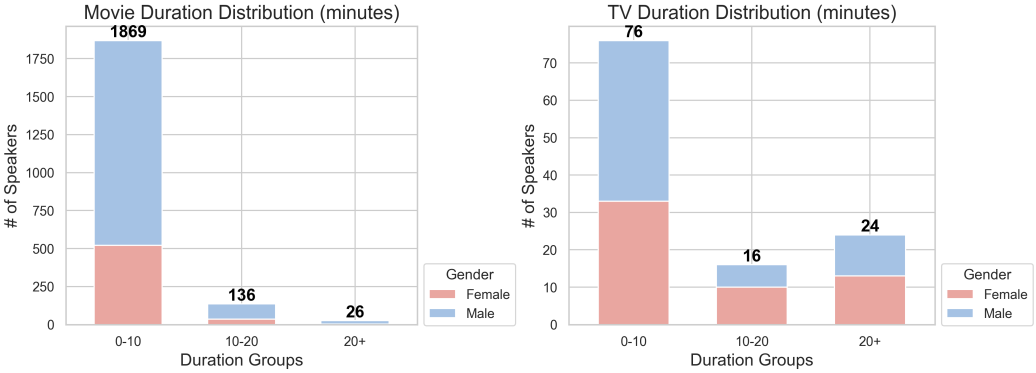
we have developed a subtitle timestamp extractor by integrating Video-SubFinder[5] and PaddleOCR[6], enabling the recognition of subtitle timestamps and text. Finally, we built a text merger, which merges video subtitles and speech recognition results based on video subtitle recognition results and timestamps. We used Intersection over Union(IOU) and Levenshtein distance[7] to sift the timestamps and text results to obtain the final timestamps and corresponding text for speech segments in the current long video.

Step 2: Speaker Verifier
The core task of the speaker verifier is to confirm the attribution of each video segment to a specific speaker. This module comprises a speech speaker verifier and a face verifier. In the speech speaker verifier, we used a speaker verification extraction model employing the ResNet101-ASP architecture from [22] for the speaker representation extraction system. The ResNet101 model [11] with residual module channel settings $[32, 64, 128, 256]$ serves as the front-end feature extractor. It is followed by an Attentive Statistics Pooling (ASP) layer [20], and a 256-dimensional fully connected layer is used as the speaker representation layer. ArcFace [6] $(s = 32, m = 0.2)$ is utilized as the classifier. The model is trained on the VoxCeleb2 dataset [4] (5994 speakers with 1092009 utterances). In the face verifier, we used the RetinaFace [5] as our Face detection model, and the IResNet [8] and ArcFace as the face recognition model. Due to the scarcity of Asian faces in existing open-source face recognition datasets, leading to inaccuracies in face recognition, we fine-tuned the face recognition model using a dataset consisting of images of 7,000 various celebrities obtained from the internet. Simultaneously, to automate the verification of characters in each film or TV show, we crawled information for 2,000 films and TV shows, including actor names, character names, dubbing names, and actor photos from Baidu Baike[8]. When using the face verifier, we register faces appearing in the video using the actor photos that we crawled online, match them with faces in the video segments, and extract speaker representation vectors from the corresponding audio segments using the speaker verification extraction model. Ultimately, we combine face verification outcomes with speaker verification results obtained through K-Means clustering to ascertain each video segment's speaker attribution.

Step 3: Speaker Audio Extractor
After successfully extracting all video segments attributed to speakers, we separate the corresponding audio and use a 5-stem spleeter [12] to extract human voices from the audio. To ensure audio quality, we only use the vocal track of the multi-channel signals derived from the spleeter toolkit.

Step 4: Character Description Generator
Drawing inspiration from Stanford Alpaca's application of large language models in data generation, we formulated a Prompt specifically for generating character descriptions in videos. Focused on processing Chinese-language video data, we utilized the

[5] https://sourceforge.net/projects/videosubfinder/
[6] https://github.com/PaddlePaddle/PaddleOCR
[7] Python implement: https://pypi.org/project/fuzzywuzzy/
[8] https://baike.baidu.com/



a) b)

Fig. 2. Data statistics crawled from the Internet.

Type	Number of Speakers	Duration (min)	Gender Ratio (Male/Female)
Movies	2032	10019.88	2.63
TV Shows	116	1706.16	1.07

Table 1. Results of TV show data processing using the multi-modal pipeline.

powerful Chinese understanding and generation capabilities provided by the Baidu Large Language Model API called ERNIE-Bot 4.0 [9]. The specific Prompt is shown in Fig. 1. This approach allows us to generate unique and distinctive character descriptions for each role.

After completing the aforementioned four steps, we obtain a speech synthesis dataset with character trait descriptions and high-quality audio.

Finally, we extracted 2,032 speakers from 783 movies, totaling 10,019.77 minutes, and 116 speakers from 3 TV Shows, totaling 1,706.16 minutes. As illustrated in Table 1, the proportion of male speakers in the extracted movie audio data is significantly higher than that of females, while the gender ratio is relatively balanced in TV show data.

Figure 2 reveals that 92% of speakers in movie data have speaking durations of less than ten minutes, with males outnumbering females. There are 136 speakers within the 10 to 20-minute range and 26 speakers with durations exceeding 20 minutes. This distribution aligns with the characteristics of movies, where each movie's duration is short, but the number of actors is high. TV show data indicates a more balanced gender ratio, with a significant proportion of speakers having durations exceeding 20 minutes, reflecting the longer speaking times of each actor in TV shows.

2.2 Controllable Voice Synthesis System Based on Character Descriptions

We present a controllable voice synthesis method based on character descriptions, as illustrated in Figure 3. The method primarily consists of four components: a pretrained large language model, a zero-shot TTS system, a reversible normalizing flow model, and a prompt encoder. In our method, we can accept two types of input to synthesize a controllable voice: the collection of movie names and role names or the character

[9] https://cloud.baidu.com/doc/WENXINWORKSHOP/s/clntwmv7t

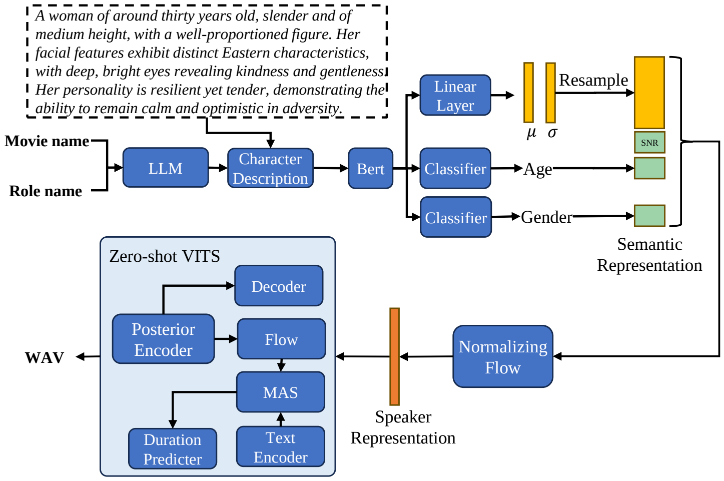

A woman of around thirty years old, slender and of medium height, with a well-proportioned figure. Her facial features exhibit distinct Eastern characteristics, with deep, bright eyes revealing kindness and gentleness Her personality is resilient yet tender, demonstrating the ability to remain calm and optimistic in adversity.
Movie name
Role name
LLM
Character Description
Bert
Linear Layer
Classifier
Classifier
μ
σ
Resample
SNR
Age
Gender
Semantic Representation
Zero-shot VITS
Decoder
Posterior Encoder
Flow
MAS
Duration Predicter
Text Encoder
WAV
Speaker Representation
Normalizing Flow

Fig. 3. Architecture of our model.

description. The pretrained large language model can generate character descriptions by providing movie names and role names. The zero-shot TTS system is designed to generate speech for a specific speaker, utilizing the speaker representation vector as input. The reversible normalizing flow model [24] decouples the speaker representation vector into a semantic representation vector consisting of quantized and non-quantized regions. The quantized region explicitly encodes attributes such as gender, age, and SNR, while the non-quantized region encompasses other non-quantifiable sub-linguistic information about the speaker. The prompt encoder identifies gender and age through the natural language description of the character and predicts the mean and variance of the non-quantized region's representation vector's prior distribution. By concatenating age, gender information, and the vector representation of the non-quantized region, the semantic representation vector is obtained after transformation through the normalizing flow model. Finally, this speaker vector is used as input to the zero-shot TTS system to synthesize speech that aligns with the character description.

Zero-shot TTS System

To synthesize speech for speakers not present in the training set, we employ the zero-shot TTS system. In recent years, the VITS structure [15] has achieved significant success in fields such as zero-shot speech synthesis [3] and voice cloning [1], [14]. Therefore, we select pre-trained VITS based on speaker representation vectors for our zero-shot TTS system. The system is built upon the VITS model, pre-trained on the AISHELL-3 dataset [23], and further fine-tuned using speech data obtained through our proposed automated multi-modal extraction pipeline. This fine-tuning process is designed to elevate the vividness and naturalness of the synthesized speech, ultimately leading to a more authentic and expressive speech synthesis outcome.

Reversible Normalizing Flow Model

To achieve stable controllability of synthesized voices concerning gender and age, ensuring consistency between synthesized voices and descriptions, we employ a reversible normalizing flow model [21]. We aim to decouple attributes such as gender and age from the speaker representation vector. We adopt the VoiceLens method proposed by

Module	Training set	Testing set
Zero-shot TTS	TV Show Dataset(Train)	Movies Dataset(Test)
Reversible Normalizing Flow Model	Movies Dataset(Train)	Movies Dataset(Test)
Prompt Encoder	Movies Dataset(Train)	Movies Dataset(Test)

Table 2. Data usage description for each module

Shi et al. [24], which utilizes conditional normalizing flow to map the voice represented by the speaker vector to a latent space. The semantic representation vector in this space can quantify attributes such as gender, age, and SNR. Hence, we choose VoiceLens as our decoupling inverse transformation model.

Prompt Encoder
The goal of the prompt encoder is to extract semantic information from the textual character description, identify the character's gender and age, and predict the mean and variance of the prior distribution of the non-quantized region of the semantic representation vector. This module consists of a pre-trained Chinese BERT module [7] and a multi-head prediction layer. The multi-head prediction layer includes three parallel linear layers, each predicting gender and age, as well as the mean and variance of the prior distribution of the non-quantized region in the text. Initially, the character description text is processed through the pre-trained Chinese BERT module to extract semantic information. Then, the semantic information is passed to the multi-head prediction layer to predict the character's gender, age, and the mean and variance of the prior distribution of the non-quantized region, which is then used to sample the vector representation of the non-quantized region of the semantic representation vector.

3 Experiments

3.1 Experiment Setting

Owing to the distinctive attributes of the tasks, we utilize TV show data featuring longer speaker's speech durations for fine-tuning the Zero-shot TTS System mentioned earlier, aiming to enhance the vividness of synthesized speech. To establish a correlation between speaker embedding and textual character descriptions, we employ movie data rich in diverse speakers for training and testing the Reversible Normalizing Flow Model and the Prompt Encoder. We excluded speakers with brief speaking durations from the multi-modal extraction pipeline's audio data, utilizing the remaining data for training and testing our model. We list the data utilized in each module in Table 2. TV Show Dataset(Train) including 45849 utterances for 18 speakers. Movies Dataset(Train) including 68640 utterances for 858 speakers. And Movies Dataset(Test) including 17160 utterances for 214 speakers.

3.2 Evaluation Metrics

This paper evaluates the proposed character description-based controllable voice synthesis system, encompassing both objective and subjective evaluations. For both evaluations, 20 speakers were selected from each category. In our objective evaluations, the speaker distance assessment was based on the x-vector method [25]. The calculation method is consistent with that described in the literature [26], [29]. The x-vector represents speaker characteristics for each audio segment. The speech speaker verifier mentioned in Section 2.1 was employed to obtain the x-vector for each audio segment.

Subsequently, we calculated the average x-vector for each speaker, obtaining a speaker-level x-vector (denoted as V). We performed distance measurements on speaker-level embedding V from three different types:

- **Speaker-level x-vector for ground-truth target speaker utterances (gt)**: we compute V_i^{gt} by averaging the x-vectors of all utterances from speaker i on the ground-truth target audio in the training set.
- **Speaker-level x-vector for synthesized speech utterances generated by using the ground-truth speaker-level x-vector (syn)**: we compute V_i^{syn} by averaging the x-vectors of all utterances from speaker i on the synthesized audio in the training set.
- **Speaker-level x-vector for generated speech utterances using character text prompts(gen)**: we compute V_i^{gen} by averaging the x-vectors of all generated speech utterances from speaker i when given the speaker prompt from testing set.

We differentiated between different speakers by computing the cosine distance between different V. The cosine distance is defined as $d(V_1, V_2) = 1 - \frac{V_1}{\|V_1\|} \cdot \frac{V_2}{\|V_2\|}$. We utilized the cosine distance between V obtained from the speech synthesis of speakers in the training set as a threshold to assess the system's performance in voice generation. We define the set of training speakers as T, and the set of generated speakers by Prompt in the testing set as G, and the following metrics are the six metrics that were computed to evaluate performance in voice generation:

- **Syn2gt-same**: Compute the speaker-level x-vector distance between synthetically generated speech V_i^{syn} and the corresponding ground truth V_i^{gt} for the same speaker in the training set. The smaller, the better.

$$\operatorname*{mean}_{i \in T} d(V_i^{syn}, V_i^{gt}) \tag{1}$$

- **Syn2gt-near**: Calculate the average distance between the training speaker V_i^{syn} and the closest ground truth training speaker V_i^{gt}. This metric is used to assess the differences between synthesized speech and the ground true speech of other speakers in the training set. The larger, the better.

$$\operatorname*{mean}_{i \in T} \min_{j \in T, i \neq j} d(V_i^{syn}, V_j^{gt}) \tag{2}$$

- **Gt2gt-near**: Calculate the average distance between the different ground truth training speakers V_i^{gt}. This metric is used to assess the differences between the ground true speech of different speakers in the training set. The larger, the better.

$$\operatorname*{mean}_{i \in T} \min_{j \in T, i \neq j} d(V_i^{gt}, V_j^{gt}) \tag{3}$$

syn2gt-same ↓	syn2gt-near ↑	syn2syn-near ↑	gen2syn-near ↑	gen2gen-near ↑	gt2gt-near ↑
0.135	0.389	0.306	0.377	0.234	0.327

Table 3. Cosine distance between different speakers

System	MOS score
Ground truth	4.80 ± 0.03
Synthetic Speakers	3.53 ± 0.07
Generated Speakers	3.35 ± 0.07

Table 4. The naturalness MOS with 95% confidence intervals

- **Syn2syn-near**: Compute the average minimum distance between synthesized speech V_i^{syn} for different speakers in the training set. This metric is applied to measure the worst-case performance in audio synthesis for different speakers within the training set. The larger, the better.

$$\underset{i \in T}{\text{mean}} \min_{j \in T, i \neq j} d(V_i^{syn}, V_j^{syn}) \tag{4}$$

- **Gen2syn-near**: Calculate the average minimum distance between audio generated from a Prompt-derived speaker V_i^{gen} in the testing set and speech synthesized V_j^{syn} for speakers in the training set. This metric assesses the worst-case performance in distances between a Prompt-generated speaker and other speakers' synthesized speech derived from the training set. The larger, the better.

$$\underset{i \in G}{\text{mean}} \min_{j \in T} d(V_i^{gen}, V_j^{syn}) \tag{5}$$

- **Gen2gen-near**: Compute the average-worst case distance between speakers generated under the same Prompt. This metric is utilized to evaluate the lower bound of the richness of speaker generation based on the same Prompt. The larger, the better.

$$\underset{i \in G}{\text{mean}} \min_{j \in G, i \neq j} d(V_i^{gen}, V_j^{gen}) \tag{6}$$

In evaluating the gender prediction capabilities of our prompt encoder within the testing set, we calculate the accuracy based on character descriptions. Accuracy A is defined as $A = \frac{C_{\text{correct}}}{N_{\text{total}}} \times 100\%$, where C_{correct} is the number of correct gender predictions and N_{total} is the total predictions made.

For the subjective evaluation, the Naturalness Mean Opinion Score (MOS) was employed to assess 60 audio samples, including real, synthetic, and test speaker audio. The synthetic speaker's speaker embedding was generated solely from character descriptions of the ground truth speakers, converting text descriptions into speaker embeddings via text embeddings, followed by zero-shot TTS generation. Conversely, test speakers' audio was produced using character descriptions from the test set, following the same conversion process. This approach allowed us to evaluate the system's effectiveness in synthesizing voices based on new, unseen descriptions, with 12 listeners participating in the evaluation.

3.3 Experimental Results

Table 3 presents the objective evaluation results of speaker similarity. Considering the robustness of our speaker verification model, the scores of syn2gt-same and syn2gt-near reflect our ability to clone the reference speaker's voice well, with good distinguishing ability between different speakers and a good speech synthetic performance. The syn2syn-near indicates that the speech generated by our synthetic model has distinctiveness between different speakers. The result of gen2syn-near suggests that our method generates new speaker voices which are much more distinct from those in the training set, demonstrating its capability to produce novel speaker voices absent in the training data. The gen2gen-near result shows that our method can generate different new voices when the same character description is used multiple times. The gt2gt-near represents the differences between ground truth speeches of different speakers. Additionally, we also conducted gender accuracy evaluation on the newly generated voices by our method. In the testing set, the gender accuracy is 98.69%, indicating that our method effectively captures gender characteristics from text inputs, enabling the TTS system to generate speech outputs with correct gender attribution.

In the subjective evaluation results provided in Table 4, our system achieves a good naturalness Mean Opinion Score (MOS) when using the genuine speaker embeddings from the testing set. The synthesized speech of new speakers generated based on character descriptions shows only a slight decrease in naturalness MOS compared to the speech generated using genuine speaker embeddings. These results indicate that our system is capable of producing high-quality new speaker voices based on textual character descriptions. Our audio samples can be accessed by visiting the following URL: https://raydonld.github.io/TMCSPEECH/

4 Acknowledgement

This research is funded in part by the National Natural Science Foundation of China (62171207), Guangdong Science and Technology Plan (2023A1111120012) and OPPO. Many thanks for the computational resource provided by the Advanced Computing East China Sub-Center.

5 Conclusions

This paper introduces an innovative multi-modal extraction pipeline efficiently designed to extract speech segments and corresponding character descriptions for each role from video data. The experimental results demonstrate that our pipeline can automatically obtain substantial quantities of accurate and high-quality character speech segments and descriptions. Additionally, we construct a controllable voice synthesis system based on character descriptions. This system establishes a one-to-many mapping relationship between character description text and speaker representation vectors, achieving the transformation from character descriptions to new speaker representation vectors. Specifically, the character description undergoes joint processing by the prompt encoder and normalization flow model, generating the input vector for the zero-shot TTS system and subsequently synthesizing speech that aligns with the character description. Experimental validation indicates that our approach is capable of generating a diverse range of voices from unseen character descriptions while maintaining a high degree of naturalness in the synthesized speech. In this work, our method still converts the character texts into speaker embeddings and then feeds them to the TTS module. In our future works, we plan to directly feed the character text prompt to the large generative TTS model with discrete tokens, hoping to achieve better speech synthesis results.

References

1. Arik, S., Chen, J., Peng, K., Ping, W., Zhou, Y.: Neural voice cloning with a few samples. Advances in Neural Information Processing Systems **31** (2018)
2. Bilinski, P., Merritt, T., Ezzerg, A., Pokora, K., Cygert, S., Yanagisawa, K., Barra-Chicote, R., Korzekwa, D.: Creating new voices using normalizing flows. arXiv preprint arXiv:2312.14569 (2023)
3. Casanova, E., Weber, J., Shulby, C.D., Junior, A.C., Gölge, E., Ponti, M.A.: Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In: International Conference on Machine Learning. pp. 2709–2720. PMLR (2022)
4. Chung, J.S., Nagrani, A., Zisserman, A.: Voxceleb2: Deep speaker recognition. arXiv preprint arXiv:1806.05622 (2018)
5. Deng, J., Guo, J., Ververas, E., Kotsia, I., Zafeiriou, S.: Retinaface: Single-shot multi-level face localisation in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5203–5212 (2020)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Duta, I.C., Liu, L., Zhu, F., Shao, L.: Improved residual networks for image and video recognition. In: 2020 25th International Conference on Pattern Recognition. pp. 9415–9422. IEEE (2021)
9. Gao, Z., Zhang, S., McLoughlin, I., Yan, Z.: Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition. arXiv preprint arXiv:2206.08317 (2022)
10. Guo, Z., Leng, Y., Wu, Y., Zhao, S., Tan, X.: Prompttts: Controllable text-to-speech with text descriptions. In: IEEE ICASSP. pp. 1–5. IEEE (2023)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hennequin, R., Khlif, A., Voituret, F., Moussallam, M.: Spleeter: a fast and efficient music source separation tool with pre-trained models. Journal of Open Source Software **5**(50), 2154 (2020). https://doi.org/10.21105/joss.02154, https://doi.org/10.21105/joss.02154, deezer Research
13. Huang, R., Huang, J., Yang, D., Ren, Y., Liu, L., Li, M., Ye, Z., Liu, J., Yin, X., Zhao, Z.: Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. arXiv preprint arXiv:2301.12661 (2023)
14. Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., Nguyen, P., Pang, R., Lopez Moreno, I., Wu, Y., et al.: Transfer learning from speaker verification to multispeaker text-to-speech synthesis. Advances in Neural Information Processing Systems **31** (2018)

15. Kim, J., Kong, J., Son, J.: Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In: International Conference on Machine Learning. pp. 5530–5540. PMLR (2021)
16. Kim, M., Cheon, S.J., Choi, B.J., Kim, J.J., Kim, N.S.: Expressive text-to-speech using style tag. arXiv preprint arXiv:2104.00436 (2021)
17. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in Neural Information Processing Systems **31** (2018)
18. Liu, G., Zhang, Y., Lei, Y., Chen, Y., Wang, R., Li, Z., Xie, L.: Promptstyle: Controllable style transfer for text-to-speech with natural language descriptions. arXiv preprint arXiv:2305.19522 (2023)
19. Liu, H., Chen, Z., Yuan, Y., Mei, X., Liu, X., Mandic, D., Wang, W., Plumbley, M.D.: Audioldm: Text-to-audio generation with latent diffusion models. arXiv preprint arXiv:2301.12503 (2023)
20. Okabe, K., Koshinaka, T., Shinoda, K.: Attentive statistics pooling for deep speaker embedding. arXiv preprint arXiv:1803.10963 (2018)
21. Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. The Journal of Machine Learning Research **22**(1), 2617–2680 (2021)
22. Qin, X., Li, N., Lin, Y., Ding, Y., Weng, C., Su, D., Li, M.: The dku-tencent system for the voxceleb speaker recognition challenge 2022. arXiv preprint arXiv:2210.05092 (2022)
23. Shi, Y., Bu, H., Xu, X., Zhang, S., Li, M.: Aishell-3: A multi-speaker mandarin tts corpus and the baselines. arXiv preprint arXiv:2010.11567 (2020)
24. Shi, Y., Li, M.: Voicelens: Controllable speaker generation and editing with flow. arXiv preprint arXiv:2309.14094 (2023)
25. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: Robust dnn embeddings for speaker recognition. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5329–5333. IEEE (2018)
26. Stanton, D., Shannon, M., Mariooryad, S., Skerry-Ryan, R., Battenberg, E., Bagby, T., Kao, D.: Speaker generation. In: IEEE ICASSP. pp. 7897–7901. IEEE (2022)
27. Tan, X., Qin, T., Soong, F., Liu, T.Y.: A survey on neural speech synthesis. arXiv preprint arXiv:2106.15561 (2021)
28. Yang, D., Liu, S., Huang, R., Lei, G., Weng, C., Meng, H., Yu, D.: Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt. arXiv preprint arXiv:2301.13662 (2023)
29. Zhang, Y., Liu, G., Lei, Y., Chen, Y., Yin, H., Xie, L., Li, Z.: Promptspeaker: Speaker generation based on text descriptions. arXiv preprint arXiv:2310.05001 (2023)