The HW-TSC's Speech-to-Speech Translation System for IWSLT 2023

Minghan Wang, Yinglu Li, Jiaxin Guo, Zongyao Li, Hengchao Shang, Daimeng Wei, Chang Su, Min Zhang, Shimin Tao, Hao Yang
[1]Huawei Translation Services Center, Beijing, China
{wangminghan,liyinglu,guojiaxin1,lizongyao,shanghengchao,
weidaimeng,suchang8,zhangmin186,taoshimin,yanghao30}@huawei.com

Abstract

This paper describes our work on the IWSLT2023 Speech-to-Speech task. Our proposed cascaded system consists of an ensemble of Conformer and S2T-Transformer-based ASR models, a Transformer-based MT model, and a Diffusion-based TTS model. Our primary focus in this competition was to investigate the modeling ability of the Diffusion model for TTS tasks in high-resource scenarios and the role of TTS in the overall S2S task. To this end, we proposed DTS, an end-to-end diffusion-based TTS model that takes raw text as input and generates waveform by iteratively denoising on pure Gaussian noise. Compared to previous TTS models, the speech generated by DTS is more natural and performs better in code-switching scenarios. As the training process is end-to-end, it is relatively straightforward. Our experiments demonstrate that DTS outperforms other TTS models on the GigaS2S benchmark, and also brings positive gain for the entire S2S system.

1 Introduction

Compared to previous iterations of the IWSLT-S2S task (Anastasopoulos et al., 2022; Guo et al., 2022), this year's task (Agarwal et al., 2023) is distinct, particularly in terms of data. The official training dataset provided is GigaS2S (Chen et al., 2021; Ye et al., 2022), which is substantially larger than previous S2S datasets, with a data size of 10,000 hours. Although the target text and speech are generated by MT and TTS systems, their quality is relatively high, making them suitable for initiating research on end-to-end S2S or TTS models in high-resource scenarios.

Our strategy is similar to that of last year (Guo et al., 2022), where we used a cascaded S2S system, but our research focus has shifted. In last year's work, we primarily studied the role of ASR and MT in the S2S system and attempted to optimize the context consistency of translation results. In this year's competition, we shifted our research focus to the TTS component. Therefore, we directly used the ASR and MT systems in our offline ST track (Wang et al., 2022a,b). Additionally, we no longer considered the issue of context consistency during inference.

Given the unprecedented success of the Diffusion Model (Ho et al., 2020; Rombach et al., 2022) in image generation over the past few years, we sought to explore its potential in speech synthesis. Thus, we proposed an end-to-end Diffusion TTS (DTS) model. Unlike previous TTS models, such as FastSpeech2 (Ren et al., 2021), which use phonemes as input and use a duration predictor to determine the duration and generate mel-spectrograms, DTS uses raw text as input, predicts the total audio length, and generates the waveform by iteratively denoising the output.

The structure of this paper is as follows: We first introduce the dataset used in this task, followed by a brief introduction to the ASR and MT models used. Then, we provide a detailed description of our proposed DTS model. Finally, we showcase the performance of each model on the GigaS2S dataset.

2 Method

2.1 Dataset

To train the ASR model, we combined five datasets and added corresponding domain tags to enable the model to generate speech in the desired style (Wang et al., 2022b). For the MT model, we aggregated all available en-de, en-zh, and en-ja translation data allowed for constrained offline tasks and added language tags to train a multilingual model. Finally, for the TTS model, we utilized the Chinese text and speech pairs from GigaS2S (Ye et al., 2022).

2.2 ASR

We trained our ASR models using a combination of five datasets: MuST-C V2, LibriSpeech, TED-

Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023), pages 277–282

Dataset	Number of Utterance	Duration(hrs)
LibriSpeech	281,241	960.85
MuST-C	340,421	590.67
IWSLT	170,229	254.41
CoVoST	1362,422	1802.52
TEDLIUM3	268,214	453.42

Table 1: Data statistics of our ASR corpora

LIUM 3, CoVoST, and IWSLT. Table 1 provides statistics for these datasets. Our model uses an 80-dimensional filterbank feature, with input samples restricted to a frame size between 50 to 3000 and a token limit of 150 to ensure that the Transformer model's encoder and decoder can process sequences of limited size.

To identify outliers, we calculated the speech speed of each sample based on the transcript length and frame size. We excluded samples with speeds outside the range of $\mu(\tau) \pm 4 \times \sigma(\tau)$, where $\tau = \frac{\text{\# frames}}{\text{\# tokens}}$.

We utilized an ensemble of two models to improve ASR performance: Conformer (Gulati et al., 2020) and S2TTransformer (Synnaeve et al., 2019). The encoder of Conformer incorporates a macaron structure at each layer based on the S2TTransformer's encoder to enhance speech encoding capability. Our ensemble method involves averaging the probabilities output by both decoders at each decoding step during beam-search. To control the model's generation style, we added prefix tags corresponding to the COVOST dataset for inference, making the model's inference style closer to GigaS2S transcripts.

2.3 MT

For MT, we utilized the multilingual Transformer model that we developed for the offline track, training it on en-zh, en-de, and en-ja datasets. To ensure high-quality pairs, we first cleaned and removed duplicates from the data, then filtered it using LaBSE (Feng et al., 2022) to select domain-specific data. During training, we employed R-Drop (Liang et al., 2021) for additional regularization. Our Transformer (Vaswani et al., 2017) model consisted of a 25-layer encoder and a 6-layer decoder with a dimension of 1024 and an FFN dimension of 4096.

2.4 TTS

2.4.1 Modeling

The Denoising Diffusion Model (DDM) (Ho et al., 2020) models a continuous process of iteratively denoising Gaussian noise to restore the original sample. The model consists of two processes: the forward process of adding noise and the reverse process of denoising. These continuous processes are assumed to have Markovian properties and can be decomposed into T conditional distributions through a Markov chain, with x_0 representing the original data (raw waveform in the TTS task) and x_T representing pure noise.

In the forward process of DDM, $q(x_{1:T}|x_0, c)$ is decomposed into a Markov process of T steps and conditioned on the input text c:

$$q(x_{1:T}|x_0, c) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \quad (1)$$

$$q(x_t|x_{t-1}, c) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}). \quad (2)$$

The sampling of x_t given x_{t-1} can be expressed as:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon, \quad (3)$$

where $\epsilon \in \mathcal{N}(0, \mathbf{I})$, and $\beta_t \in [0, 1]$ is a noise scheduler related to t. Therefore, each step of the forward process is adding a certain amount of Gaussian noise to the previously corrupted speech x_{t-1}. Finally, x_0 ultimately evolves into white noise that follows a Gaussian distribution. An important characteristic of the forward process is that $x_t \sim q(x_t|x_0, c)$ for any t has a closed form:

$$q(x_t|x_0, c) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}\} \quad (4)$$

$\alpha_t = 1 - \beta_t$, and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, so we can efficiently obtain x_t for any t from x_0 during training.

In the reverse process, the denoising process is similar to the forward process, and is also described as a T-step Markov process:

$$p_\theta(x_{0:T}, c) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \quad (5)$$

$$p_\theta(x_{t-1}|x_t, c) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t, c), \sigma_t^2\mathbf{I}), \quad (6)$$

where the μ_θ can be learned by neural networks and $\sigma_t^2 = 1 - \alpha_t$.

The training objective of the Diffusion Model is to maximize the log-likelihood of $p(x_0|c)$, which

is intractable, so optimization on the variational bound is used instead. (Ho et al., 2020) further simplify it to an unweighted version of L2 regression loss with respect to $\hat{\epsilon}$ and added noise ϵ. In our work, we predict the x_0 with the model instead of the noise:

$$L(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[||\hat{x}_\theta(x_t, t, c) - x_0||\right] \quad (7)$$

Here, t is uniformly sampled from the interval $[0, T]$.

During inference, the model iteratively samples x_{t-1} from x_t:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t, c) + \sigma_t \mathbf{z}\right) \quad (8)$$

$$\epsilon_\theta = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(x_t - \sqrt{\bar{\alpha}_t}\hat{x}_\theta(x_t, t, c)\right) \quad (9)$$

where $\sigma_t = \sqrt{1-\alpha_t}$ and $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. In our experiments, to allow for flexible determination of the maximum step T, we choose to use a continuous t ranging from 0 to 1. During training, t is uniformly sampled, and we use the cosine noise scheduler (Nichol and Dhariwal, 2021).

In addition to modeling the denoising process, DTS also needs to predict the length of the target audio in advance, as DTS is essentially a non-autoregressive (NAR) model. However, unlike previous TTS models that predict the duration of each phoneme, we directly model the total number of frames in the target audio, which is more convenient. Specifically, we use the text representation after average pooling, denoted as $\mathbf{h}_c$, as the input to the classifier ϕ to predict the length distribution. Then, we calculate the cross-entropy loss with the frame number N_{x_0} of x_0.

$$L_{\text{length}} = CE(\phi(\mathbf{h}_c; \theta), N_{x_0}) \quad (10)$$

2.4.2 Model Architecture

The DTS model is essentially a parameterized denoising function $\hat{x}(x_t, t, c)$ which takes x_t, t as input, conditions on c, and predicts the x_0 for the sampling of x_{t-1}. The model makes some modifications on top of the Transformer model to make it more suitable for speech synthesis. As shown in Figure 1, the main modifications are as follows:

- On top of the Encoder, we add a two-layer FFN network to predict the length of the target audio.

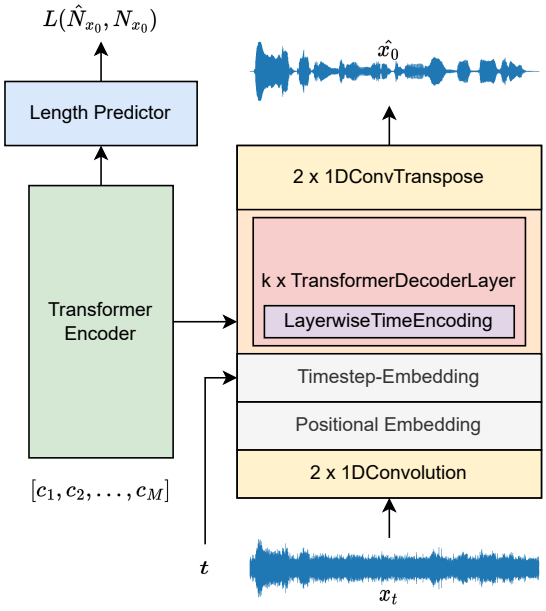

Figure 1: The architecture of DTS model, which takes $C = [c_1, ..., c_M]$ as the encoder input to predict the frame length N. For the decoder, it takes x_t and t as input, conditions on C to predict x_0 for the sampling of x_{t-1} according to Eq 8 and 9.

- In the input part of the Decoder, we use two 1D convolutions with a proper setting of kernel size, stride, and padding, so the sequence length before and after convolution remains unchanged.

- As the Diffusion model depends on the time step t, we additionally introduce a Timestep Embedding, and use the same implementation as (Ho et al., 2020).

- To make the time step encoding more comprehensive, we add Layerwise time encoding at each layer and added to the encoded hidden states from the last layer.

- In the output part of the decoder, we add 2 1D deconvolutions to restore the hidden state back to the waveform. We use deconvolution because we found that using only linear projection leads to a lack of dependency between the generated waveform and the previous waveform, resulting in noticeable jitter, which can be significantly eliminated by using deconvolution.

Model	WER-all-punct	WER-all	WER-code-switch	WER-zh
FastSpeech 2	13.18	10.75	15.70	8.37
DTS-Mel	13.32	10.28	15.66	7.69
DTS-Wave	**12.68**	**9.82**	**15.33**	**7.17**

Table 2: This table shows the performance of our TTS models on the GigaS2S dev set, using ground truth transcripts as input. We compare our models against FastSpeech 2 (Ren et al., 2021), which serves as the baseline. Additionally, we present a DTS model trained to predict mel-spectrograms (DTS-Mel) for comparison with DTS for waveform (DTS-Wave). The table reports the word error rate (WER) for the entire set with punctuation (WER-all-punct), WER for all samples without punctuation (WER-all), WER for code-switch samples without punctuation (WER-code-switch), and WER for Chinese-only samples without punctuation (WER-zh). The results indicate that DTS-Wave outperforms the other models, achieving the lowest WER values in all categories.

Model	WER	WER-no-punct
S2TTransformer	22.67	18.15
Conformer	22.42	17.80
Ensemble	21.57	16.92

Table 3: The performance of our two independent ASR models and the ensemble of them with or without punctuation.

Model Input	BLEU	ChrF
ASR output	29.0	25.4
Ground Truth	30.7	27.3

Table 4: The performance of our MT models with ground truth input and asr outputs as the input.

Model	BLEU	ChrF
FastSpeech2	21.8	22.7
DTS-Mel	22.3	23.1
DTS-Wave	22.7	23.4

Table 5: The overall cascade performance evaluated by BLEU and ChrF.

3 Experiment

3.1 Experimental Setup

For the ASR and MT parts of our S2S system, we directly used the same setting as in the Offline track. For the TTS part, we trained the model on the GigaS2S dataset for 360k steps, with a maximum learning rate of 1e-4, warmup of 20000 steps, and a batch size of 32 samples per GPU. The maximum and minimum audio lengths were restricted to 25 seconds and 0.5 seconds, respectively. The model has 12 layers in the encoder and 16 layers in the decoder, with a hidden dimension of 512 and an FFN dimension of 2048. DTS can directly generate waveforms, but since audio waveforms are usually long, we pre-segment them into equally sized non-overlapping frames. In this way, the model learns to generate the waveform frame by frame, and we only need to flatten these frames to get the final output. In our experiments, we used a frame length of 1200 and a sampling rate of 24000. When inference, we set the sampling step to 100. In addition to the raw waveform, DTS can also learn to generate mel-spectrogram, simply by changing wave frames to spectrogram frames. This is also evaluated in our experiment.

3.2 Experimental Results

In the experiments, we tested the performance of each module in our S2S system separately. In addition to testing with the cascaded results as input, we also conducted independent tests with ground truth input. For the three modules, we mainly used the dev set of GigaS2S for evaluation. In terms of evaluation metrics, for ASR and MT, we used WER, BLEU and ChrF, respectively. For TTS, we used a Whisper-medium (Radford et al., 2022) model to transcribe the TTS-generated audio back into the text for automatic evaluation and calculated WER.

ASR Results We evaluated the results of two ASR models trained on the same corpus separately, as well as the ensemble version. As shown in Table 3, the ensemble results were slightly better.

MT Results In the evaluation of MT, we considered two scenarios: using ground truth transcripts as input and using the output of the previous ASR module as input. The experimental results showed that the robustness of MT was relatively good, even if there were errors in the ASR output, the difference in BLEU score was not significant as shown

in Table 4.

TTS Results In the TTS experiments, because the development set of GigaS2S contains code-switching samples, we evaluated not only the WER of the entire set but also separately evaluated the cases without the code-switching. As for the models, we chose FastSpeech 2 as the baseline. In addition, we trained an additional DTS based on mel-spectrogram for comparison with the waveform-based DTS. Both FS2 and DTS-mel used the Griffin-lim vocoder. As shown in Table 2, DTS-Wave outperformed the other two models, especially on Chinese monolingual data.

Full Pipeline Results In addition to testing each module separately, we also tested the final metrics of the entire pipeline. We compared the difference between the speech generated by the three TTS models with the MT results as input by computing the BLEU and ChrF with the ground truth translation. Table 5 shows that there is a difference that existed, but it is not significant. Therefore, we can conclude that the quality of the speech generated by TTS does affect the final performance of S2S system in terms of automatic evaluation, but the impact is still limited.

4 Conclusion

In this paper, we present the system we developed for the IWSLT2023 speech-to-speech competition. The system includes relatively simple and effective ASR and MT modules, as well as a TTS module proposed by us based on the Diffusion Model. In the experiments, we demonstrate that the denoising diffusion process can effectively learn end-to-end TTS task, simplifying both training and inference. However, its generation speed is relatively slow. In our future work, we will continue to optimize its quality and generation efficiency, and further explore the application of diffusion in end-to-end S2S tasks.

References

Milind Agarwal, Sweta Agrawal, Antonios Anastasopoulos, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, William Chen, Khalid Choukri, Alexandra Chronopoulou, Thierry Declerck, Qianqian Dong, Yannick Estève, Kevin Duh, Marcello Federico, Souhir Gahbiche, Benjamin Hsu, John Judge, Tom Ko, Rishu Kumar, Xutail Ma, Prashant Mathur, Evgeny Matusov, Paul McNamee, John P. McCrae, Kenton Murray, Matteo Negri, Jan Niehues, Xing Niu, Atul Ojha Kr., John E. Ortega, Proyag Pal, Juan Pino, Lonneke van der Plas, Elijah Rippeth, Elizabeth Salesky, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Brian Thompson, Marco Turchi, Alex Waibel, Mingxuan Wang, and Rodolfo Zevallos. 2023. Findings of the IWSLT 2023 Evaluation Campaign. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*. Association for Computational Linguistics.

Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondrej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gahbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Vera Kloudová, Surafel Melaku Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nadejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Miguel Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alexander Waibel, Changhan Wang, and Shinji Watanabe. 2022. Findings of the IWSLT 2022 evaluation campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation, IWSLT@ACL 2022, Dublin, Ireland (in-person and online), May 26-27, 2022*, pages 98–157. Association for Computational Linguistics.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Zhao You, and Zhiyong Yan. 2021. Gigaspeech: An evolving, multi-domain ASR corpus with 10, 000 hours of transcribed audio. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pages 3670–3674. ISCA.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 878–891. Association for Computational Linguistics.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 5036–5040. ISCA.

Jiaxin Guo, Yinglu Li, Minghan Wang, Xiaosong Qiao, Yuxia Wang, Hengchao Shang, Chang Su, Yimeng Chen, Min Zhang, Shimin Tao, Hao Yang, and Ying

Qin. 2022. The hw-tsc's speech to speech translation system for IWSLT 2022 evaluation. In *Proceedings of the 19th International Conference on Spoken Language Translation, IWSLT@ACL 2022, Dublin, Ireland (in-person and online), May 26-27, 2022*, pages 293–297. Association for Computational Linguistics.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 10890–10905.

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *CoRR*, abs/2212.04356.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE.

Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Edouard Grave, Tatiana Likhomanenko, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. 2019. End-to-end ASR: from supervised to semi-supervised learning with modern architectures. *CoRR*, abs/1911.08460.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Minghan Wang, Jiaxin Guo, Yinglu Li, Xiaosong Qiao, Yuxia Wang, Zongyao Li, Chang Su, Yimeng Chen, Min Zhang, Shimin Tao, Hao Yang, and Ying Qin. 2022a. The hw-tsc's simultaneous speech translation system for IWSLT 2022 evaluation. In *Proceedings of the 19th International Conference on Spoken Language Translation, IWSLT@ACL 2022, Dublin, Ireland (in-person and online), May 26-27, 2022*, pages 247–254. Association for Computational Linguistics.

Minghan Wang, Jiaxin Guo, Xiaosong Qiao, Yuxia Wang, Daimeng Wei, Chang Su, Yimeng Chen, Min Zhang, Shimin Tao, Hao Yang, and Ying Qin. 2022b. The hw-tsc's offline speech translation system for IWSLT 2022 evaluation. In *Proceedings of the 19th International Conference on Spoken Language Translation, IWSLT@ACL 2022, Dublin, Ireland (in-person and online), May 26-27, 2022*, pages 239–246. Association for Computational Linguistics.

Rong Ye, Chengqi Zhao, Tom Ko, Chutong Meng, Tao Wang, Mingxuan Wang, and Jun Cao. 2022. Gigast: A 10, 000-hour pseudo speech translation corpus. *CoRR*, abs/2204.03939.