MLLP-VRAIN UPV systems for the IWSLT 2022 Simultaneous Speech Translation and Speech-to-Speech Translation tasks

Javier Iranzo-Sánchez and **Javier Jorge** and **Alejandro Pérez-González-de-Martos**
Adrià Giménez and **Gonçal V. Garcés Díaz-Munío** and **Pau Baquero-Arnal**
Joan Albert Silvestre-Cerdà and **Jorge Civera** and **Albert Sanchis** and **Alfons Juan**
Machine Learning and Language Processing Group
Valencian Research Institute for Artificial Intelligence
Universitat Politècnica de València
Camí de Vera s/n, 46022 València, Spain

Abstract

This work describes the participation of the MLLP-VRAIN research group in the two shared tasks of the IWSLT 2022 conference: Simultaneous Speech Translation and Speech-to-Speech Translation. We present our streaming-ready ASR, MT and TTS systems for Speech Translation and Synthesis from English into German. Our submission combines these systems by means of a cascade approach paying special attention to data preparation and decoding for streaming inference.

1 Introduction

In this paper we describe the participation of the MLLP-VRAIN research group in the shared tasks of the *19th International Conference on Spoken Language Translation (IWSLT)*. We participated in two shared tasks: the *Simultaneous Speech Translation* and the (offline) *Speech-to-Speech Translation* tasks. The translation pair for both tasks was English to German. Our submission follows the cascade approach, with individual ASR, MT and TTS components. We use common ASR and MT models for both tasks, with additional latency restrictions for the Simultaneous task. In short, for the Simultaneous S2T task our system comprises a one-pass decoder ASR system based on the HMM-DNN approach with a chunk-based BLSTM AM combined with a Transformer LM, followed by a multi-k Transformer-based MT system. Regarding the S2S translation task, the aforementioned systems are followed by a non-autoregressive Conformer-based text-to-spectogram module, ending with a multi-band UnivNet neural vocoder to convert from the spectogram to the final audio wave.

This paper is structured as follows. Section 2 describes our participation in the *Simultaneous Speech Translation (ST)* task: the architecture and design decisions of the ASR and MT components in our cascade system, and the evaluation of the individual components as well as the speech translation system as a whole. Section 3 describes our participation in the *Speech-to-Speech (S2S) Translation* task, paying special attention to the speaker-adaptive TTS system specifically developed for this task. Our conclusions for the shared task are drawn in Section 4.

2 Simultaneous Speech Translation

2.1 ASR System Description

The acoustic model (AM) was trained using 3649 hours from resources listed in Table 4 in Appendix A. The evaluation sets were those provided with MuST-C v2.0: *tst-HE*, *tst-COMMON* and *dev*, for the English-German language pair. To train the AM we follow our training recipe for the DNN-HMM model, thoroughly described in Jorge et al. (2022). After this training pipeline we end up with a BLSTM network with 8 bidirectional hidden layers and 512 LSTM cells per layer and direction, with 10861 output labels (sub-phonetic units), trained with TensorFlow (Abadi et al., 2015). During inference, to enable streaming recognition, we perform a chunking-based processing of the input to carry out both feature normalization and feature scoring, as also described in Jorge et al. (2022).

Regarding the language model (LM), we trained a count-based model (n-gram) and a neural-based model (Transformer LM, TLM). For the former, we trained a 4-gram LM with KenLM (Heafield, 2011) using 1.3G sentences and 17G of running words (see Table 5 in Appendix A for a complete list of resources). For the latter, in order to alleviate the training time for this neural model, we selected a subset with the WIT3, MuST-C, and a random sample from the rest of the data up to 1G words. This TLM was trained using an adapted version of the FairSeq toolkit (Ott et al., 2019). The architecture is based on a 24-layer network with 768 units per layer, 4096-unit feed-forward neural

Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022), pages 255 - 264

network, 12 attention heads, and an embedding of 768 dimensions. These models were trained until convergence with batches limited to 512 tokens. Parameters were updated every 32 batches. During inference, Variance Regularization was applied to speed up the computation of TLM scores (Baquero-Arnal et al., 2020). Regarding the selected vocabulary, it comprises 300K words, with an OOV rate of about 0.3% on the selected dev sets. Lastly, we combined these acoustic and language models to perform a one-pass streaming recognition with our internal decoder implemented in TLK (del Agua et al., 2014).

2.2 MT System Description

The MT system must be ready to translate unpunctuated, lowercase ASR transcriptions. To prepare the MT system for this, the source side of the training data is pre-processed using the same approach as that applied to the LM training data (Iranzo-Sánchez et al., 2020a). Subword segmentation is based on the SentencePiece described in Kudo and Richardson (2018). Internally, 40k BPE operations are used, jointly learned on the source and target data, and the white-space sentence word separator symbol is used as a suffix to ease the decoding.

Most of our efforts this year have been focused on data preparation, selection and filtering. We have considered the following setups for training our models:

- *Baseline* data setup: For this configuration, we use all of the WMT20 news translation task data (Barrault et al., 2020), Europarl-ST (Iranzo-Sánchez et al., 2020b), MuST-C v2 (Di Gangi et al., 2019) and the TED corpus (Cettolo et al., 2012a), for a total of 48M sentence pairs used for training.

- *WMT21*: We use WMT21 news translation task (Akhbardeh et al., 2021) data instead of WMT20, for a total of 97M sentence pairs used for training.

- *OpenSubtitles*: Add the OpenSubtitles 2018 (Lison and Tiedemann, 2016) to the training data. This adds an additional 22M sentence pairs to the training data.

- *Bicleaner*: We use the Bicleaner and Bifixer tools (Ramírez-Sánchez et al., 2020) to filter the training data. We use the v1.4 pre-trained model published by the Bitextor team to score the sentences, and we do not run the LM component during filtering. We filter the sentences using two values for the filtering threshold, 0.3 and 0.5, so sentences with a score lower than the threshold are discarded before training.

- *Clean ups.*: In order to increase the proportion of clean data used by the model during training, we take those parallel corpora that contain document-level information (TED, news-commentary, Wikititles, rapid, Europarl, Europarl-ST and MuST-C), and upsample them by a factor of 5. Our expectation is that corpora which contain entire documents can be more reliable that sentence pairs extracted from other sources.

- *[ASR]-half*: Using this configuration, we prepend a new special token [ASR] to the source text sequence to be translated during inference. Additionally, during training, only half of the data is pre-processed following the ASR recipe, and we append the special [ASR] tag to it. The other half of the data keeps its original casing and punctuation. Ideally, this would allow the model to learn how to translate ASR output, while at the same time having access to some information about capitalization and casing during training. This setup is inspired in Zhao et al. (2021), but the authors used a different pre-processing schema.

All our models are based on the Transformer BIG architecture (Vaswani et al., 2017). We use the Adam optimizer, learning rate 5e-4 with an inverse square root decay, and train for a total of 1M batches of 16k tokens each. After training finishes, we carry out domain adaptation by finetuning on the MuST-C train data for 5000 updates or until the dev perplexity stops improving.

For training simultaneous MT models, we use the multi-k approach (Elbayad et al., 2020), because it achieves competitive results while at the same time provides us with the flexibility of adjusting the latency at inference time. By default, a random k is used for each batch, sampled between 1 and the length of the longest sentence included in the batch. We also tried training with a smaller k upper bound to check whether the quality improves in low-latency scenarios.

During decoding, we use beam search with a beam size of 6 for the offline model, whereas we

Table 1: PPL and WER figures for the *dev* and *tst-HE/CO(MMON)* sets with 4-gram model and TLM.

		dev	*tst-HE*	*tst-CO*
PPL	4-gram	117	117	106
	TLM	54	54	55
WER	4-gram	7.8	7.2	9.5
	TLM	5.8	5.3	7.3

Table 2: WER figures varying the window size (in ms) of the look-ahead context of the chunk-based BLSTM.

look-ahead window	250	500	1000	1500
dev	6.9	5.8	5.6	5.6
tst-HE	6.6	5.3	5.1	5.0
tst-COMMON	9.3	7.3	7.0	7.1

use speculative beam-search (Zheng et al., 2019) with a beam size of 4 for simultaneous models. Higher beam values significantly increased decoding costs for a negligible increase in quality. In order to speed-up decoding, we first compute how many w words we need to generate based on the wait-k policy. Then, we carry out speculative beam-search by generating hypothesis with a maximum length of $w \cdot a + b + 1$ subwords, where a and b are two hyperparameters optimized on the dev set. If this first search does not generate the w words we need, we carry out a second search with a maximum hypothesis length of 150 subwords.

2.3 ASR System Evaluation

First, we carried out a comparative evaluation in terms of perplexity (PPL) and Word Error Rate (WER) between the 4-gram model and the TLM on the MuST-C.v2 dev set *dev* and the test sets, *tst-HE* and *tst-COMMON*. Table 1 shows PPL and WER figures on dev and test sets having validated and fine-tuned hyperparameters on the dev set. It is worth noting how roughly halving perplexity involves a consistent WER reduction of about 23-25%.

Next, with the best setup from the previous experiment (using TLM) we performed another set of evaluations to explore the impact of the size of the window for the acoustic look-ahead context on WER. For this comparison, we considered values of 250, 500, 1000, and 1500 ms of future context for the chunk-based BLSTM. Table 2 illustrates the resulting WER when the look-ahead context is modified. As expected, providing more future context allows the model to deliver more accurate scores, reducing the WER. Indeed, increasing this context results in a WER reduction of about 20% the cost of increasing the latency from 250 to 1000 ms.

2.4 MT System Evaluation

As in the ASR system, we also use the MuST-C.v2 dev set in order to validate and fine-tune hyperparameters. Additionally, we report results on the MuST-C.v2 tst-COMMON set, as well as on the IWSLT 2015 and 2018 test sets, using the BLEU score (Papineni et al., 2002).

Table 3 shows BLEU figures of a conventional offline system and a range of simultaneous multi-k systems trained on the data setups described in Section 2.2. These results correspond to the fine-tuned models using the in-domain MuST-C data, which results in a consistent improvement across all training setup. For the sake of comparison on the Baseline data setup between the offline and simultaneous system, the simultaneous multi-k system was evaluated when running inference in offline mode ($k = 100$). The ranking of training data setups for multi-k systems with $k \in \{1, 3, 6, 15\}$ on inference time was the same.

As observed in Table 3, the unidirectional encoder used for training the multi-k system (system #2) results in a small quality degradation when compared with the offline model (system #1), similarly to what was observed in (Iranzo-Sánchez et al., 2022). Adding OpenSubtitles to the data (system #3) shows some improvements across the evaluation sets. The use of the *[ASR]-half* pre-processing scheme (system 4) shows a promising 1.7 BLEU increase on MuST-C tst-COMMON, but it does not convey to other evaluation sets. Other tentative configurations using the *[ASR]-half* approach did not improve over non-*[ASR]-half* results.

With regards to systems using WMT21 data (systems #5-7), it is surprising to see that the additional data does not seem to improve results across the board, even if we use filtering, when compared to the baseline data configuration. Additional experiments are needed on this regard, but a possible explanation is that the smaller baseline dataset is more in-domain than the larger WMT21 set, perhaps due to the speech corpora being a bigger portion of the training data.

Based on our intuition behind the results provided by systems #5-7, we ran an additional experiment combining the WMT21 with data upsampling

Table 3: BLEU scores of offline and multi-k MT systems for different training data setups on MuST-C.v2 *dev* and *tst*-CO(MMON), and IWSLT 2015 and 2018 test sets.

#	System	*dev*	*tst*-CO	tst2015	tst2018
1	Offline Baseline	33.0	33.8	33.4	31.6
2	Multi-k Baseline	32.2	32.8	32.3	30.7
3	+ OpenSubtitles	**32.3**	33.3	**33.2**	30.7
4	+ [ASR]-half	31.4	**34.5**	30.4	28.8
5	+ WMT21	31.9	32.6	32.5	30.2
6	+ Bicleaner (tr=0.3)	31.7	32.6	32.5	31.0
7	+ Bicleaner (tr=0.5)	31.8	32.3	32.8	30.9
8	+ Clean ups. & OpenSubtitles	32.2	32.9	32.6	**31.1**

and the OpenSubtitles2018 corpora (system #8, see Section 2.2). This configuration obtained better results than systems #4-7, and even outperformed system #2 on tst2018. Based on the results on the dev set, we selected systems #3 and #8 for further experimentation.

The default implementation of the multi-k system samples a random k each batch, with a maximum k value of the longest sentence in the batch. In our case, we discard before training all sentences longer than 100 words. This means that the model trains across multiple latency regimes, and in some batches is actually training with the same restrictions as an offline model. Thus, it might be beneficial to train with a smaller upper value of k, in order to encourage better translation quality for low-latency regimes. We trained a new system #3 with a maximum k of 20 subwords and study its trade-off between latency measured as Average Lagging (AL) (Ma et al., 2019) and BLEU compared with the conventional system #3 (maximum k=100) in Figure 1. As shown, no performance improvement at low latency when training with a smaller k threshold is observed, and therefore we decided not to use the multi-k system trained with maximum $k = 20$.

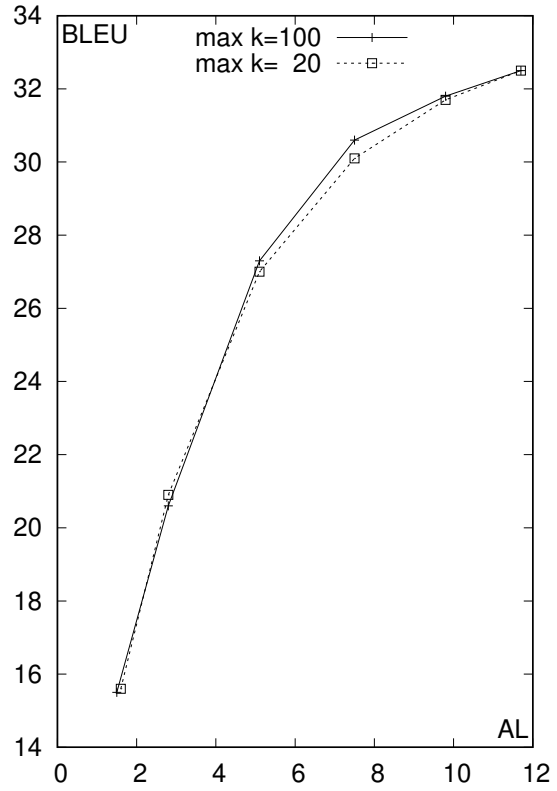

Figure 1: BLEU versus AL for maximum values of $k \in \{20, 100\}$ for multi-k system #3 measured on MuST-C.v2 *tst*-COMMON.

2.5 Simultaneous S2T System Evaluation

Based on the previously described ASR and MT systems, we now move into optimizing the decoding hyper-parameters of the joint cascade system. For the ASR component, we optimized the pruning parameters, that is, the grammar scale factor, the beam and the number of active hypotheses at both sub-phonetic and word level, as well as the recombination limit and the look-ahead acoustic context. As described before all experiments were carried out using the TLM model, since no differences on computational AL were found between both language models. For the MT component, we optimized the inference time k, and the a and b hyperparameters of the speculative beam search.

The goal is to obtain the best hyperparameter combination that satisfies the AL thresholds defined in the simultaneous task, $1000, 2000$, and 4000. Our cascade systems operates approximately at Real-Time Factor of 0.5, so we first run a wide hyperparameter sweep using *tst*-HE, which is a smaller dataset than *tst*-COMMON. The results are

shown in Figure 2.

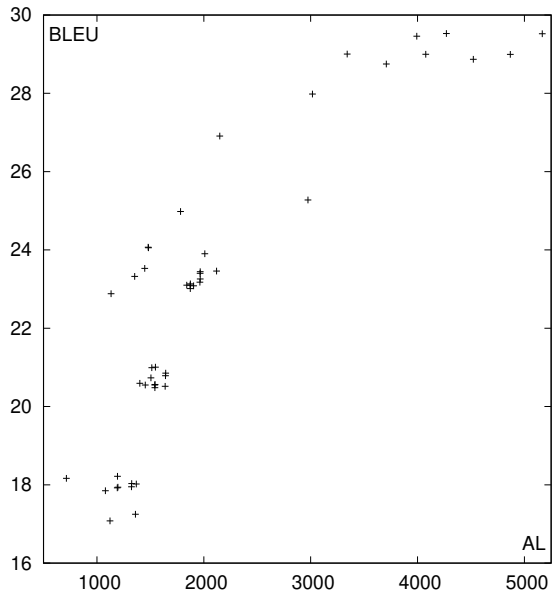

Figure 2: BLEU vs AL for different hyperparameter configurations of our simultaneous ST system measured on MuST-C.v2 *tst*-HE.

It can be observed how the choice of hyperparameters is critical in order to maximize the quality of the system, as there are differences of up to 4 BLEU points between systems that have the same latency. We found it significantly hard to obtain a system with AL$\leq$ 1000, as our ASR decoder with a TLM takes a long time to consolidate hypothesis. We came up with a strategy in order to be able to submit a low-latency system, so that every time a new transcribed word is consolidated, we also send the unconsolidated part of the top scoring hypothesis to the MT system. Using this strategy, our hope is that if the unconsolidated hypothesis do not show a lot of variation, the latency of the cascade system can be significantly reduced in exchange for a small degradation of translation quality. We tested this strategy as well as our best performing systems (#3 and #8) on tst-COMMON, and report BLEU versus AL in Figure 3.

Figure 3 shows how we were able to stay below the AL$=$ 1000 threshold thanks to using the ASR unconsolidated hypothesis. Based on these results, our final submission to the shared task are shown in Figure 3 as filled points, with system #8 submitted as *System 1, Primary*, and system #3 submitted as *System 2, Contrastive*.

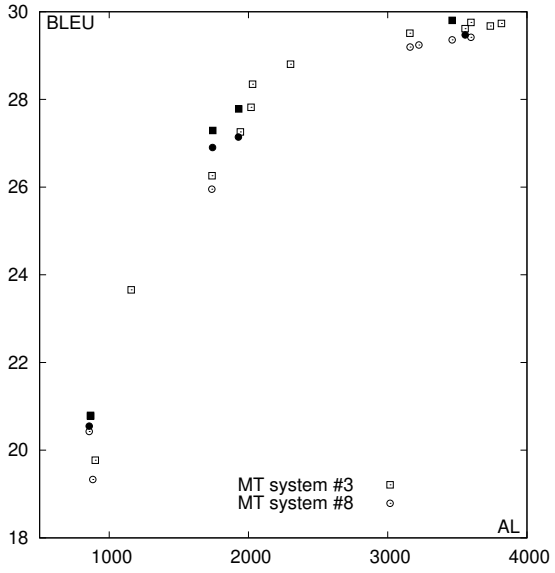

Figure 3: BLEU vs AL for different configurations of simultaneous ST systems measured on MuST-C.v2 tst-COMMON. Filled points were included in our submission to the shared task.

3 Speech-to-Speech Translation

In this section we describe our submission to the Speech-to-Speech translation track, in which we include a speaker-adaptive TTS module to our previously described cascaded Speech Translation system. Thus, we reuse the ASR and MT models developed for the Simultaneous Speech Translation task, though imposing a less restrictive pruning setup. This involves, in brief, more look-ahead context and a wider search space for the ASR system described in Section 2.1, and using the offline MT system instead of the simultaneous multi-k MT system referred to in Section 2.2. Therefore, the remaining of this section will describe the additional TTS module included to carry out the final text-to-audio conversion of the S2S pipeline.

3.1 TTS System Description

In the context of the S2S translation task, for many applications the TTS module should not only be able to produce high quality natural sounding synthetic speech in a predefined set of voices, but ideally also be capable of mimicking the voice characteristics of the original speaker in the target language (e.g. male or female). To that end, our proposed TTS model follows the transfer learning approach to zero-shot speaker adaptation or multi-speaker TTS (Doddipatla et al., 2017; Jia et al., 2018; Cooper et al., 2020; Casanova et al., 2021),

where an auxiliary speaker encoder model trained on a speaker classification task is leveraged to compute speaker embeddings from reference utterances both during training and inference.

Our speaker encoder model follows the modified ResNet-34 residual network architecture (He et al., 2016) from Chung et al. (2018), which is being widely used for speaker recognition tasks with excellent results (Xie et al., 2019; Chung et al., 2020b). However, similar to Chung et al. (2020a) we halve the number of filters in each residual block with respect to the original ResNet-34 architecture to reduce computational costs and avoid over-fitting when trained on relatively small datasets. The model is trained on a speaker classification task on the TED-LIUM v3 dataset (Hernandez et al., 2018), which contains 452 hours of transcribed speech data from 2351 TED conference talks given by 2028 unique speakers. To reduce class imbalance, we limit the number of audio segments per speaker to 50. We trim leading and trailing silence, apply a pre-emphasis filter with a coefficient of 0.97 and extract 64-dim log-mel spectrograms from training samples. During training, we also perform on-the-fly audio data augmentation such as randomly adding Gaussian noise, reverberations, dynamic range compression and frequency masking in order to help generalization to different audio recording conditions. Mean and variance normalization is performed by adding an instance normalization layer to the spectrogram inputs. The model is trained to minimize the Angular Prototypical loss (Chung et al., 2020b), in which we set $M = 2$ where M is the number of samples per speaker in each mini-batch. We use the Adam optimizer with a fixed learning rate of 0.0005 and train the model for 100K steps using a mini-batch size of 300 samples (150 different speakers), each comprising 2.5 seconds.

Our TTS model follows the two-stage approach to end-to-end neural text-to-speech. It is comprised of a non-autoregressive Conformer-based *text-to-spectrogram* network and a *spectrogram-to-wave* multi-band UnivNet (Jang et al., 2021; Yang et al., 2020) neural vocoder. We extract phoneme durations by means of a forced-aligner auto-encoder model trained on the same data as in de Martos et al. (2021). The Conformer encoder and decoder blocks follow the modifications proposed in Liu et al. (2021). First, the Swish activation function is replaced with ReLU for better generalization, particularly on long sentences. Second, the depthwise convolution is placed before the self-attention module for faster convergence. Finally, the linear layers in feed-forward modules are replaced by convolution layers.

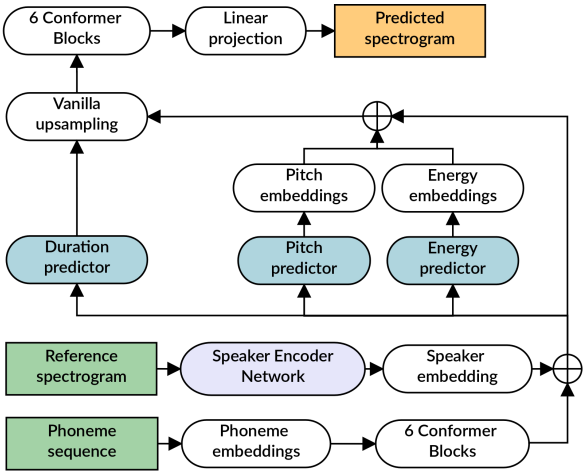

Figure 4: Speaker-adaptive Conformer text-to-spectrogram network architecture.

Figure 4 depicts the speaker-adaptive text-to-spectrogram network architecture. The encoder and decoder modules consist of 6 Conformer blocks with attention dimension 384 and a kernel size of 1536 for convolutional feed-forward modules. The speaker encoder model is used to extract 256-dim speaker embeddings which are linearly projected and added to the encoder hidden states. The variance adaptor modules (duration, pitch and energy predictors) follow the convolutional architecture in Ren et al. (2021) with 2, 5 and 2 layers, respectively. The pitch prediction is done similarly as in Łańcucki (2020), where frame-wise F_0 values are first converted to the logarithmic domain and averaged over every input symbol using phoneme durations. Then, predicted (ground truth during training) phoneme-level pitch values are projected and added to the encoder hidden states by means of a 1-D convolution.

The text-to-spectrogram model is trained on the LibriVoxDeEn dataset (Beilharz et al., 2020), comprising 547 hours (487 hours after silence trimming) of sentence-aligned audios from German audio books. We down-sample all audios to 16kHz and compute 100-bin log-mel spectrograms with Hann windowing, 50ms window length, 12.5ms hop size and 1024 point Fourier transform. Phoneme sequences are extracted from normal-

ized text transcriptions using the eSpeak NG[1] tool. Frame-wise pitch (F_0) values are estimated using the WORLD vocoder toolkit (MORISE et al., 2016; Morise et al., 2009). The model is optimized to minimize a combination of the $\ell 1$ loss and the SSIM (Structural SIMilarity index measure) (Wang et al., 2004) between reference and predicted spectrograms. Additionally, auxiliary $\ell 1$ losses are used also for the duration, pitch and energy variance prediction modules between reference and predicted values. An auxiliary $\ell 1$ loss between standard deviation values of target and predicted pitch contours (F_0 values) is used to encourage the pitch predictor produce less flattened prosody as the result of training on a huge variety of speakers. We train the model using the Adam optimizer for 500K steps on a NVIDIA RTX 3090 GPU with a batch size of 12 and a learning rate of 0.0001 with a linear ramp up for the first 5000 steps.

Finally, a 4-band UnivNet vocoder is trained to generate 24kHz audios from 16kHz spectrograms. UnivNet is a recent GAN-based vocoder that has been shown to produce high quality speech of comparable quality to best performing GAN vocoders such as HiFi-GAN (Su et al., 2020) while bringing an improved inference speed of about $1.5\times$. The model is trained on the LibriVoxDeEn 16kHz ground truth spectrograms and 22kHz original audios (up-sampled to 24kHz for simplicity) with a batch size of 64 distributed along 4 GPUs for 1M steps. Then, the text-to-spectrogram model is used to compute ground truth aligned spectrograms using reference phoneme durations, pitch and energy values, and the vocoder model is fine-tuned on the predicted spectrograms for an additional 100K steps.

4 Conclusions

The MLLP-VRAIN research group has participated in the Simultaneous Speech Translation and Speech-to-Speech Translation tasks using our state-of-the art streaming-ready cascade systems. Under the cascade approach, each individual component has been described and evaluated, as well as the joint cascade system.

The results show that the cascade approach remains a flexible and powerful solution for ST tasks, yet at the same time there is a great deal of hyperparameter optimization that needs to be carried out in order to properly integrate the different components. The use of unconsolidated ASR hypothesis has enabled very low-latency translation in exchange for a small decrease in quality. In terms of future work, we would like to further study the use of partial hypothesis by the MT system and other downstream components, as a means of improving the quality-latency tradeoff.

[1] http://espeak.sourceforge.net

Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements no. 761758 (X5Gon) and 952215 (TAILOR), and Erasmus+ Education programme under grant agreement no. 20-226-093604-SCH (EXPERT); the Government of Spain's grant RTI2018-094879-B-I00 (Multisub) funded by MCIN/AEI/10.13039/501100011033 & "ERDF A way of making Europe", and FPU scholarships FPU18/04135; and the Generalitat Valenciana's research project Classroom Activity Recognition (ref. PROMETEO/2019/111).

References

News Crawl corpus (WMT workshop) 2015. http://www.statmt.org/wmt15/translation-task.html.

Martín Abadi et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondrej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussà, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proc of WMT*, pages 1–88.

Pau Baquero-Arnal, Javier Jorge, Adrià Giménez, Joan Albert Silvestre-Cerdà, Javier Iranzo-Sánchez, Alberto Sanchís, Jorge Civera Saiz, and Alfons Juan-Císcar. 2020. Improved Hybrid Streaming ASR with Transformer Language Models. In *Proc. of Interspeech*, pages 2127–2131.

Loïc Barrault, Ondrej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Feder-

mann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors. 2020. *Proceedings of the Fifth Conference on Machine Translation*.

Benjamin Beilharz, Xin Sun, Sariya Karimova, and Stefan Riezler. 2020. Librivoxdeen: A corpus for german-to-english speech translation and speech recognition. In *Proc. of LREC*.

Edresson Casanova, Christopher Shulby, Eren Gölge, Nicolas Michael Müller, Frederico Santos de Oliveira, Arnaldo Candido Jr., Anderson da Silva Soares, Sandra Maria Aluisio, and Moacir Antonelli Ponti. 2021. SC-GlowTTS: An Efficient Zero-Shot Multi-Speaker Text-To-Speech Model. In *Proc. of Interspeech*, pages 3645–3649.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012a. WIT^3: Web Inventory of Transcribed and Translated Talks. In *Proc. of EAMT*, pages 261–268.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012b. Wit3: Web inventory of transcribed and translated talks. In *Proc. of EAMT*, pages 261–268.

Joon Son Chung, Jaesung Huh, and Seongkyu Mun. 2020a. Delving into VoxCeleb: Environment Invariant Speaker Recognition. In *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, pages 349–356.

Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee-Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han. 2020b. In Defence of Metric Learning for Speaker Recognition. In *Proc. of Interspeech*, pages 2977–2981.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. VoxCeleb2: Deep Speaker Recognition. In *Proc. of Interspeech*, pages 1086–1090.

Erica Cooper, Jeff Lai, Yusuke Yasuda, Fuming Fang, Xin Wang, Nanxin Chen, and Junichi Yamagishi. 2020. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In *Proc. of ICASSP*, pages 6184–6188.

Alejandro Pérez-González de Martos, Albert Sanchis, and Alfons Juan. 2021. Vrain-upv mllp's system for the blizzard challenge 2021. *arXiv preprint arXiv:2110.15792*.

M.A. del Agua et al. 2014. The translectures-UPV toolkit. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 269–278.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proc. of NAACL-HLT*, pages 2012–2017.

Rama Doddipatla, Norbert Braunschweiler, and Ranniery Maia. 2017. Speaker Adaptation in DNN-Based Speech Synthesis Using d-Vectors. In *Proc. of Interspeech*, pages 3404–3408.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient Wait-k Models for Simultaneous Machine Translation. In *Proc. of Interspeech*, pages 1461–1465.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. of CVPR*, pages 770–778.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proc. of WMT*, page 187–197.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. 2018. Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer*, pages 198–208.

Javier Iranzo-Sánchez, Jorge Civera, and Alfons Juan. 2022. From simultaneous to streaming machine translation by leveraging streaming history. *arXiv preprint arXiv:2203.02459*.

Javier Iranzo-Sánchez, Adrià Giménez, Joan Albert Silvestre-Cerdà, Pau Baquero, Jorge Civera, and Alfons Juan. 2020a. Direct Segmentation Models for Streaming Speech Translation. In *Proc. of EMNLP*, pages 2599–2611.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020b. Europarl-ST: A Multilingual Corpus For Speech Translation Of Parliamentary Debates. In *Proc. of ICASSP*, pages 8229–8233.

Won Jang, Dan Lim, Jaesam Yoon, Bongwan Kim, and Juntae Kim. 2021. UnivNet: A Neural Vocoder with Multi-Resolution Spectrogram Discriminators for High-Fidelity Waveform Generation. In *Proc. of Interspeech*, pages 2207–2211.

Ye Jia et al. 2018. Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. In *Proc. of NIPS*, pages 4485–4495.

Javier Jorge, Adrià Giménez, Joan Albert Silvestre-Cerdà, Jorge Civera, Albert Sanchis, and Alfons Juan. 2022. Live streaming speech recognition using deep bidirectional lstm acoustic models and interpolated language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:148–161.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of MT Summit*, pages 79–86.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proc. of EMNLP: System Demonstrations*, pages 66–71.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proc. of LREC*, pages 923–929.

Yanqing Liu, Zhihang Xu, Gang Wang, Kuan Chen, Bohan Li, Xu Tan, Jinzhu Li, Lei He, and Sheng Zhao. 2021. Delightfultts: The microsoft speech synthesis system for blizzard challenge 2021. *arXiv preprint arXiv:2110.12612*.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proc. of ACL*, pages 3025–3036. ACL.

Masanori Morise, Hideki Kawahara, and Haruhiro Katayose. 2009. Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech. In *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. Audio Engineering Society.

Masanori MORISE, Fumiya YOKOMORI, and Kenji OZAWA. 2016. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, E99.D(7):1877–1884.

Mozilla. 2022. Commonvoice 6.1.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL-HLT*, pages 48–53.

V. Panayotov et al. 2015. Librispeech: an ASR corpus based on public domain audio books. In *Proc. of ICASSP*, pages 5206–5210.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, pages 311–318.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proc. of EAMT*, pages 291–298.

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 *Proc. of ICLR*.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: a large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proc. of EACL*, pages 1351–1361.

Jiaqi Su, Zeyu Jin, and A. Finkelstein. 2020. Hifi-gan: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks. In *Proc. of Interspeech*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proc. of LREC*.

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 *Proc. of NIPS*, pages 5998–6008.

Zhou Wang, Alan Bovik, Hamid Sheikh, and Eero Simoncelli. 2004. Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13:600 – 612.

Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2019. Utterance-level aggregation for speaker recognition in the wild. In *Proc. of ICASSP*, pages 5791–5795.

Geng Yang et al. 2020. Multi-band MelGAN: Faster Waveform Generation for High-Quality Text-to-Speech. *arXiv preprint arXiv:2005.05106*.

Chengqi Zhao, Zhicheng Liu, Jian Tong, Tao Wang, Mingxuan Wang, Rong Ye, Qianqian Dong, Jun Cao, and Lei Li. 2021. The volctrans neural speech translation system for IWSLT 2021. In *Proc. of IWSLT*, pages 64–74.

Renjie Zheng, Mingbo Ma, Baigong Zheng, and Liang Huang. 2019. Speculative beam search for simultaneous translation. In *Proc. of EMNLP-IJCNLP*, pages 1395–1402.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proc. of LREC*, pages 3530–3534.

Adrian Łańcucki. 2020. Fastpitch: Parallel text-to-speech with pitch prediction. *arXiv preprint arXiv:2006.06873*.

A Appendix: ASR resources

Table 4: Transcribed speech resources, with the sets used and total hours per set and globally. (tr=train, d=dev, t=test, v=val, do/to=dev-other/test-other)

Set	Hours
CommonVoice 6.1 (Mozilla, 2022) (v)	1668.0
Librispeech(tr+do+to) (Panayotov et al., 2015)	970.1
MuST-C v2.0(tr en-{de,ja,zh}) (Di Gangi et al., 2019)	608.2
How2 (Sanabria et al., 2018)(tr+v+d)	304.5
Europarl-ST v1.1 (tr+d+t) (Iranzo-Sánchez et al., 2020b)	98.7
Total	3649.6

Table 5: Text resources used to train the ngram LM.

Set	Sent (K)	Words (M)
News discussions	635117.8	8317.1
News crawl (new)	274930.0	6029.9
Open Subs 18 (Lison and Tiedemann, 2016)	439507.3	2429.2
WikiMatrix v1 (Schwenk et al., 2021)	19422.8	2107.5
UN Parallel Corpus V1.0 (Ziemski et al., 2016)	14517.5	308.4
Europarl v10 (Koehn, 2005)	2317.3	56.3
News Commentary (Tiedemann, 2012) v1	646.8	14.1
LibriSpeech	287.0	9.5
CommonVoice 6.1	613.5	6.3
MuST-C v2.0	389.3	6.3
How2	191.6	3.4
Europarl-ST v1.1	36.0	0.9
WIT3 (Cettolo et al., 2012b)	14.6	0.2
Total	1387991.6	17522.1