LEVERAGING TIMESTAMP INFORMATION FOR SERIALIZED JOINT STREAMING RECOGNITION AND TRANSLATION

Sara Papi[‡*], *Peidong Wang*[†], *Junkun Chen*[†], *Jian Xue*[†], *Naoyuki Kanda*[†], *Jinyu Li*[†], *Yashesh Gaur*[†]

[†]Microsoft, USA
[‡]Fondazione Bruno Kessler and University of Trento, Italy
spapi@fbk.eu,{peidongwang,junkunchen,jian.xue,nakanda,jinyli,yashesh.gaur}@microsoft.com

ABSTRACT

The growing need for instant spoken language transcription and translation is driven by increased global communication and cross-lingual interactions. This has made offering translations in multiple languages essential for user applications. Traditional approaches to automatic speech recognition (ASR) and speech translation (ST) have often relied on separate systems, leading to inefficiencies in computational resources, and increased synchronization complexity in real time. In this paper, we propose a streaming Transformer-Transducer (T-T) model able to jointly produce many-to-one and one-to-many transcription and translation using a single decoder. We introduce a novel method for joint token-level serialized output training based on timestamp information to effectively produce ASR and ST outputs in the streaming setting. Experiments on {it,es,de}↔en prove the effectiveness of our approach, enabling the generation of one-to-many joint outputs with a single decoder for the first time.

***Index Terms*—** speech recognition, speech translation, streaming, joint, timestamp

1. INTRODUCTION

With the expansion of global communication and cross-lingual interactions, the demand for real-time spoken language transcription and translation in multiple languages is rapidly increasing [1]. Conventionally, this task is addressed by separate automatic speech recognition (ASR) and translation (ST) models, leading to the necessity of running several models in parallel to obtain the required outputs. This leads to a huge demand for computational resources, in contrast with Green AI [2], and also increases the complexity of coordinating several systems in real-time. Moreover, in some applications like news reports, maintaining consistency between on-screen transcriptions and translations is crucial to deliver to the user similar content [3, 4, 5].

Previous works [6, 7] have shown improvements in quality and consistency when the model is trained to jointly generate both ASR and ST outputs. This approach was later adapted to the streaming scenario by Weller et al. [8] using an attention-based encoder-decoder architecture [9] with re-translation [10]. More recently, Papi et al. [11] proposed the adoption of a Transformer-Transducer (T-T) architecture [12, 13], which is more suitable for the simultaneous scenario [14], with the joint token-level serialized output training (joint t-SOT). This method employs an off-the-shelf textual aligner to determine how to effectively produce transcription and translation words in real-time. However, their study only focused on the many-to-one language setting and can be applied to one translation direction at a time, since finding the alignment between one source language and multiple target languages is a very complex task [15, 16, 17].

To overcome this limitation, in this paper, we propose a streaming T-T model that is able to jointly produce both many-to-one and one-to-many outputs using a single decoder. We introduce a novel interleaving method based on timestamp information that enables the model to learn how to produce multiple target languages while maintaining a low latency. Comparative experiments on {it,es,de}↔en with separate and multilingual state-of-the-art T-T architectures show the effectiveness of our interleaving approach, yielding significant improvements in terms of transcription quality while being competitive when producing multiple translation languages.

2. METHOD

2.1. Joint t-SOT

Inspired by t-SOT [18] originally proposed for multi-talker ASR, in the joint t-SOT, two output modalities are produced by the model: ASR and ST. Therefore, two special symbols are introduced $\langle asr \rangle$ and $\langle st \rangle$ to interleave a word or a set of words. Specifically, given the reference transcription $\mathbf{r_{asr}} = [r_{asr_1}, ..., r_{asr_m}]$ with $m \leq \text{len}(\mathbf{r_{asr}})$ and translation $\mathbf{r_{st}} = [r_{st_1}, ..., r_{st_n}]$ with $n \leq \text{len}(\mathbf{r_{st}})$, the joint t-SOT reference is:

$$\mathbf{r}_{\text{t-SOT}} = [\langle asr \rangle, r_{asr_1}, r_{asr_2}, ..., r_{asr_m}, \langle st \rangle, r_{st_1}, r_{st_2}, ..., r_{st_n}]$$

To obtain the final $\mathbf{r}_{\text{t-SOT}}$, the concatenation process is repeated until all the ASR and ST words have been consumed, i.e. $m = \text{len}(\mathbf{r_{asr}})$ and $n = \text{len}(\mathbf{r_{st}})$.

[*]Work done during an internship at Microsoft.

2.2. Timestamp-based joint t-SOT

In Weller et al. [8], several methods have been proposed to interleave ASR and ST modalities. In INTER 0.0, the entire ASR output is emitted before the ST one, whereas in INTER 1.0, the entire ST output is emitted first and followed by ASR. In INTER 0.5, ASR and ST words are alternated. These methods were then integrated into the joint t-SOT [11], and extended to another technique, INTER ALIGN, where the words are interleaved based on the source-target text alignments obtained by an external off-the-shelf tool. However, all these methods are not trivial to adapt to the multilingual target scenario, especially the INTER ALIGN method since finding a one-to-many alignment is a complex task.

To overcome this limitation, we propose **INTER TIME**, a novel interleaving method based on word-level timestamps, which not only is able to build more effective joint t-SOT outputs but also enables the one-to-many multilingual scenario by interleaving more than one translation language at a time. Specifically, for each reference word, we extract the corresponding timestamp information (for simplicity, we consider the end time) by applying the Viterbi algorithm to pretrained streaming models. This process is executed for each modality (ASR and ST) and language direction.

In the one-to-many setting, let $\mathbf{r_{asr}}$ be the utterance transcription, $\mathbf{r_{st_1}}, ..., \mathbf{r_{st_n}}$ the corresponding translations in n different languages, and $\langle asr \rangle$, $\langle st_1 \rangle$,..., $\langle st_n \rangle$ the language tags.[1] Each element of $\mathbf{r_{asr}}$, $\mathbf{r_{st_1}}, ..., \mathbf{r_{st_n}}$ is composed of three elements $(time, tag, word)$, where $time$ is the timestamp (integer number, in milliseconds), tag is the corresponding language tag, and $word$ is the word that has been emitted with timestamp $time$. For instance, if the word "***I***" has been uttered at 200ms, the word "***am***" at 300ms, and the word "***happy.***" at 500ms, the corresponding $\mathbf{r_{asr}}$ extracted from the ASR model is:

$$\mathbf{r_{asr}} = [(200, \langle asr \rangle, \text{“}\textbf{\textit{I}}\text{”}), (300, \langle asr \rangle, \text{“}\textbf{\textit{am}}\text{”}) \\ (500, \langle asr \rangle, \text{“}\textbf{\textit{happy.}}\text{”})]$$

if the Spanish translation is "***Estoy feliz.***" with emission timestamps $[250, 550]$ and the German translation is "***Ich bin froh.***" with timestamps $[350, 550, 650]$, the corresponding $\mathbf{r_{st_1}}$ and $\mathbf{r_{st_2}}$ extracted from the ST models are:

$$\mathbf{r_{st_1}} = [(250, \langle st_1 \rangle, \text{“}\textbf{\textit{Estoy}}\text{”}), (550, \langle st_1 \rangle, \text{“}\textbf{\textit{feliz.}}\text{”})]$$

$$\mathbf{r_{st_2}} = [(350, \langle st_2 \rangle, \text{“}\textbf{\textit{Ich}}\text{”}), (550, \langle st_2 \rangle, \text{“}\textbf{\textit{bin}}\text{”}) \\ (650, \langle st_2 \rangle, \text{“}\textbf{\textit{froh.}}\text{”})]$$

The INTER TIME output is built by applying Algorithm 1 to $\mathbf{r_{asr}}$, $\mathbf{r_{st_1}}$, $\mathbf{r_{st_2}}$ to obtain the final $\mathbf{r}_{\text{t-SOT}}$. In particular, the reference words for each modality and language are concatenated, sorted by timestamp (increasing order) and then interleaved following the temporal order. The language tag is inserted only if the previous interleaved word was of a different language. Following the previous example, the output is:

$$\mathbf{r}_{\text{t-SOT}} = [(200, \langle asr \rangle, \text{“}\textbf{\textit{I}}\text{”}), (250, \langle st_1 \rangle, \text{“}\textbf{\textit{Estoy}}\text{”}), \\ (300, \langle asr \rangle, \text{“}\textbf{\textit{am}}\text{”}), (350, \langle st_2 \rangle, \text{“}\textbf{\textit{Ich}}\text{”}), \\ (500, \langle asr \rangle, \text{“}\textbf{\textit{happy.}}\text{”}), (550, \langle st_1 \rangle, \text{“}\textbf{\textit{feliz.}}\text{”}), \\ (550, \langle st_2 \rangle, \text{“}\textbf{\textit{bin}}\text{”}), (650, \langle st_2 \rangle, \text{“}\textbf{\textit{froh.}}\text{”})]$$

with $\mathbf{r_{asr}}$ = "#ASR#", $\mathbf{r_{st_1}}$ = "#ES#", $\mathbf{r_{st_2}}$ = "#DE#", the corresponding textual output used during training is:

> "**#ASR# *I* #ES# *Estoy* #ASR# *am* #DE# *Ich* #ASR# *happy.* #ES# *feliz.* #DE# *bin froh.***"

Note that Algorithm 1 can be easily applied to the many-to-one scenario by using different ASR models to obtain multilingual $\mathbf{r_{asr}}$ and a unique $\mathbf{r_{st}}$.

Algorithm 1 INTER TIME

Require: $\mathbf{r_{asr}}, \mathbf{r_{st_1}}, ..., \mathbf{r_{st_n}}$ ▷ ASR and multi ST references
  $\mathbf{r} \leftarrow [\ \ ]$
  for r_i **in** $[\mathbf{r_{asr}}, \mathbf{r_{st_1}}, ..., \mathbf{r_{st_n}}]$ **do**
    $\mathbf{r} \leftarrow \mathbf{r} + r_i$ ▷ Concatenate all the reference words
  end for
  $\mathbf{r} \leftarrow sort_{time}\big(\mathbf{r}(time, tag, word)\big)$ ▷ Sort by timestamp
  $\mathbf{r}_{\text{t-SOT}} \leftarrow [\ \ ]$
  $prev\_tag \leftarrow$ None
  for $(time, tag, word)$ **in** $\mathbf{r}$ **do**
    if $tag \neq prev\_tag$ **then** ▷ Language switch
      $\mathbf{r}_{\text{t-SOT}} \leftarrow \mathbf{r}_{\text{t-SOT}} + tag$
      $prev\_tag \leftarrow tag$
    end if
    $\mathbf{r}_{\text{t-SOT}} \leftarrow \mathbf{r}_{\text{t-SOT}} + word$
  end for

2.3. Time Step Grouping

With the aim of limiting the frequency of the switch between languages, we propose the adoption of a grouping mechanism in the data construction process. The grouping mechanism is guided by the size of the time step T (e.g., 500ms, 1000ms, ...). It groups the $(time, tag, word)$ tuple of each reference word r_i of the sorted reference $sort_{time}\big(\mathbf{r}(time, tag, word)\big)$ in Algorithm 1 by looking at the $time$ attribute. Then, the words that belong to the current same time step group t_s, i.e. $t_s - T \leq time < t_s$, are interleaved together. The final sequence can be obtained by substituting the $time$ attribute of each word with its corresponding t_s in Algorithm 1.

For instance, if we look at the example in Section 2.2 and set the step size T to 300ms, we have three groups $[time <$

[1] $\langle asr \rangle$, $\langle st_1 \rangle$,..., $\langle st_n \rangle$ are not considered as special tokens during training: they are added directly to the vocabulary and considered as all the other tokens in the loss computation.

$300, 300 \leq time < 600, 600 \leq time < 900]$. If we substitute $time$ with the corresponding t_s in the sorted $\mathbf{r}$, we obtain:

$$\begin{aligned}\mathbf{r} = [&(300, \langle asr \rangle, \text{“}\textbf{\textit{I}}\text{”}), (300, \langle st_1 \rangle, \text{“}\textbf{\textit{Estoy}}\text{”}),\\ &(600, \langle asr \rangle, \text{“}\textbf{\textit{am}}\text{”}), (600, \langle asr \rangle, \text{“}\textbf{\textit{happy.}}\text{”}),\\ &(600, \langle st_1 \rangle, \text{“}\textbf{\textit{feliz.}}\text{”}), (600, \langle st_2 \rangle, \text{“}\textbf{\textit{Ich}}\text{”}),\\ &(600, \langle st_2 \rangle, \text{“}\textbf{\textit{bin}}\text{”}), (900, \langle st_2 \rangle, \text{“}\textbf{\textit{froh.}}\text{”})]\end{aligned}$$

that corresponds to the output:

“**#ASR#** ***I*** **#ES#** ***Estoy*** **#ASR#** ***am happy.*** **#ES#** ***feliz.*** **#DE#** ***Ich bin froh.***”

The overall language switch reduction depends on the time step T and, on our training data, is estimated as -34% for 500ms, and -54% for 1000ms.

3. EXPERIMENTAL SETTINGS

For all our experiments, we use a streaming T-T architecture [19] with 24 Transformer layers for the encoder with 8 attention heads, 6 LSTM layers for the predictor and 2 feed-forward layers for the joiner. The embedding dimension of the encoder is 512 and the feed-forward units are 4096. We use a chunk size of 1 second with 18 left chunks. The LSTM predictor and feed-forward layers of the joiner have 1024 hidden units. We use 80-dimensional log-mel fiterbanks (fbanks) as features, sampled every 10 milliseconds. Before feeding them to the Transformer encoders, we apply 2 layers of CNN with stride 2 and kernel size of (3, 3), with an overall input compression of 4. The total number of parameters is 188.5M.

Our Many-to-English experiments follow the settings of previous work [11]: all models are trained for 6.4M steps on 1k hours of proprietary data for each source language (Italian, Spanish, German) and tested on the CoVoST2 dataset [20]. 8k-sized SentencePiece vocabulary [21] was trained with coverage 1.0 and shared between languages.

For the English-to-Many experiments, we used 1k hours of English audio with the corresponding translation into Italian, Spanish, and German. The models are tested on the FLEURS dataset [22]. The multitask multilingual ASR & ST model is realized by pre-pending the language ID (LID) tag [23], i.e. by replacing the `<SOS>` with `<LID>`. Pre-prended LID is used also to train the single-translation version of the joint t-SOT. All but separate models are trained for 6.4M steps starting from the multitask multilingual ASR & ST model weights pretrained for 3.2M steps, including the multitask multilingual model itself. Timestamps for INTER TIME are estimated using monolingual ASR and ST models trained on the same data. Time step grouping is applied at 500ms and 1000ms, since preliminary experiments with higher values (e.g., 2000ms) showed quality degradation.

AdamW [24] is used as optimizer with the RNN-T loss [25]. Checkpoints are saved every 320k steps. The learning rate is set to 3e-4 with Noam scheduler, 800k warm-up steps and linear decay. We use 16 NVIDIA V100 GPUs with 32GB of RAM for all the training and a batch size of 350k. We select the last checkpoint for inference, which is then converted to open neural network exchange (ONNX) format and compressed. The beam size of the beam search is set to 7.

We report WER for the ASR quality and BLEU[2] for the ST quality. Latency is measured in milliseconds (ms) with the length-adaptive average lagging (LAAL) [27].

4. RESULTS

4.1. Many-to-English

Table 1 shows the results for the {it,es,de}-en language directions. For comparison, we report the results for the joint t-SOT INTER 0.5 and INTER ALIGN approaches while not including the INTER 0.0 and 1.0 since they are not streaming for one of the two modalities (either ASR or ST).

We observe that INTER TIME achieves higher or similar BLEU scores, except for de-es, and obtains the best WER on all the source languages (Italian, Spanish, and German). Compared to all models, it yields improvements from 0.93 to 3.19 WER on average among languages while showing a comparable latency with INTER ALIGN, the model with lower latency. When time step grouping is applied, we notice an overall translation quality improvement, especially using 500ms time step, with average gains of 0.54 BLEU compared to the multilingual ASR & ST and, respectively, 0.12 and 0.64 BLEU compared to INTER ALIGN and TIME without time step grouping. With 500ms time step, INTER TIME also obtains higher quality transcriptions, with an average improvement of -2.00 WER compared to the multilingual ASR & ST, and -0.91 WER compared to INTER ALIGN while being comparable with INTER TIME without time step grouping.

All in all, the best quality-latency trade-off is achieved by INTER TIME with time step grouping of 500ms, yielding the best results in most languages and modalities.

4.2. English-to-Many

In Table 2, we compare our joint t-SOT INTER TIME using both multiple translation languages (rows 4-6), and only one translation language (rows 7-9) in the target. Therefore, the latter produces the transcription and the corresponding translation in a single language, similar to Section 4.1.

First, comparing ASR and ST models (rows 1-3), we observe that the multitask multilingual ASR & ST model achieves the best results, with an improved WER and similar or better BLEU scores compared to using separate models for modalities (ASR and ST) and languages. For the joint t-SOT INTER TIME trained on multiple translation languages, we notice that the time step grouping helps with the performance, both in terms of quality and latency, yielding improvements

[2] sacreBLEU [26] version 2.3.1

Model	# inf. steps	it-en				es-en				de-en			
		WER	LAAL	BLEU	LAAL	WER	LAAL	BLEU	LAAL	WER	LAAL	BLEU	LAAL
separate ASR & ST*	6	25.83	1191	16.41	1844	22.69	1149	19.24	1682	23.11	1071	19.11	1613
multilingual ASR & ST*		23.48	1181	21.06	1663	22.84	1147	22.76	1622	21.82	1133	**21.51**	1642
joint t-SOT INTER 0.5*	3	22.35	1110	20.22	1515	21.19	1126	22.25	1468	21.35	1051	20.19	1547
joint t-SOT INTER ALIGN*		21.74	**1092**	21.80	**1355**	21.04	**1094**	23.42	**1341**	22.07	**1043**	21.36	**1335**
joint t-SOT INTER TIME	3	**21.11**	1141	21.70	1442	19.79	1143	23.38	1452	**21.16**	1112	19.96	1791
+ 500ms step grouping		21.22	1142	**22.05**	1493	**19.74**	1139	**24.09**	1489	21.17	1103	20.81	1664
+ 1000ms step grouping		21.64	1115	21.75	1457	20.26	1052	23.75	1467	21.49	1076	20.58	1651

Table 1. WER↓ and BLEU↑ on CoVoST 2 for the Many-to-English setting with their latency LAAL↓. **Bold** represents overall best result, underline represents best result balancing both quality and latency (there can be multiple combinations for each language). * Results reported in [11].

Model	# inf. steps	en		en-it		es-en		de-en	
		WER	LAAL	BLEU	LAAL	BLEU	LAAL	BLEU	LAAL
separate ASR & ST	4	29.02	1089	8.76	1932	9.50	1853	9.87	2156
+ multilingual ST				11.17	1612	11.34	1618	**13.14**	1799
multitask multilingual ASR & ST		27.53	917	**11.56**	1607	11.38	1608	13.11	1844
joint t-SOT INTER TIME	1	36.23	1544	7.52	2313	8.28	2331	8.54	2497
+ 500ms step grouping		31.51	1118	9.68	1668	9.86	1852	11.30	1993
+ 1000ms step grouping		29.34	913	10.85	**1395**	10.90	**1509**	12.74	1918
- single translation	3	**26.33**	959	10.38	1564	11.24	1520	12.39	**1733**
+ 500ms step grouping		27.00	918	11.45	1580	**11.89**	1610	12.79	1830
+ 1000ms step grouping		26.81	**892**	11.25	1776	11.52	1797	12.85	1999

Table 2. WER↓ and BLEU↑ on FLEURS for the English-to-Many setting with their latency LAAL↓. **Bold** represents overall best result, underline represents best result balancing both quality and latency (there can be multiple combinations for each language). ASR results of joint t-SOT INTER TIME with single translation are averaged among the three languages.

of -6.89 WER and an average of +3.38 BLEU with 983ms latency reduction obtained by the 1000ms step grouping. However, compared with the strongest ASR & ST system (multitask multilingual), our model shows a quality degradation of +1.81 WER and -0.52 BLEU although with a slight latency reduction. Even demonstrating quality degradation, especially on the recognition quality, it is important to notice that this model is the only one producing all the outputs within a single inference step. Therefore, quality drops are expected but, nevertheless, the results obtained by this model are competitive with those obtained by training separate ASR & ST models, showing to be a promising direction.

If we constrain the join t-SOT INTER TIME strategy to deal with only one translation language, we observe significant improvements compared to its multiple translation languages version, especially with the time step grouping. WER improvements range from -2.34 to -3.01 and average BLEU gains are up to 0.55 compared to multiple translations and 1000ms time step grouping. In accordance with the results obtained in Section 4.1, the 500ms time step grouping is the best-performing model and, compared with the multitask multilingual ASR & ST model, it yields a transcription quality improvement of -0.53 WER while maintaining comparable or slightly better translation quality and latency.

To conclude, we show the effectiveness of the joint t-SOT INTER TIME, especially when time step grouping is applied. Results on both ASR and ST tasks show that our method achieves the best overall results compared to the strongest multitask multilingual model and, if extended to deal with multiple translation languages all at once, maintains comparable results to models specifically tailored for ASR or ST.

5. CONCLUSIONS

In this paper, we proposed a streaming Transformer-Transducer able to jointly produce both many-to-one and one-to-many transcriptions and translations. To effectively train the model to maximize ASR and ST quality while minimizing latency, we introduced INTER TIME, a novel method for the joint token-level serialized output training based on timestamp information. We also proposed a variant of this method based on grouping the timestamp according to a fixed time step.

Comparative studies on {it,es,de}-en and en-{it,es,de} prove the effectiveness of our approach, especially when the time step grouping is adopted, achieving the best performance in both many-to-one and one-to-many scenarios while being competitive when the joint generation of multiple target languages with a single decoder is enabled.

6. REFERENCES

[1] E. Steigerwald, V. Ramírez-Castañeda, D. Y. C. Brandt, A. Báldi, J. T. Shapiro, L. Bowker, and R. D. Tarvin, "Overcoming language barriers in academia: Machine translation tools and a vision for a multilingual future," *Bioscience*, vol. 72, pp. 988 – 998, 2022.

[2] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Commun. ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[3] C. Fügen, *A System for Simultaneous Translation of Lectures and Speeches*, Ph.D. thesis, 2009.

[4] A. Karakanta, M. Gaido, M. Negri, and M. Turchi, "Between flexibility and consistency: Joint generation of captions and subtitles," in *Proc. 18th IWSLT*, 2021, pp. 215–225.

[5] J. Xu, F. Buet, J. Crego, E. Bertin-Lemée, and F. Yvon, "Joint generation of captions and subtitles with dual decoding," in *Proc. 19th IWSLT*, 2022, pp. 74–82.

[6] M. Sperber, H. Setiawan, C. Gollan, U. Nallasamy, and M. Paulik, "Consistent transcription and translation of speech," *TACL*, vol. 8, pp. 695–709, 2020.

[7] H. Le, J. Pino, C. Wang, J. Gu, D. Schwab, and L. Besacier, "Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation," in *Proc. 28th COLING*, Dec. 2020, pp. 3520–3533.

[8] O. Weller, M. Sperber, C. Gollan, and J. Kluivers, "Streaming models for joint speech recognition and translation," in *Proc. 16th EACL*, 2021, pp. 2533–2539.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st NeurIPS*, 2017, p. 6000–6010.

[10] J. Niehues, N. Pham, T. Ha, M. Sperber, and A. Waibel, "Low-Latency Neural Speech Translation," in *Proc. Interspeech*, 2018, pp. 1293–1297.

[11] S. Papi, P. Wang, J. Chen, J. Xue, J. Li, and Y. Gaur, "Token-level serialized output training for joint streaming asr and st leveraging textual alignments," 2023.

[12] C. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. L Seltzer, "Transformer-transducer: End-to-end speech recognition with self-attention," 2019.

[13] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. ICASSP*, 2020, pp. 7829–7833.

[14] J. Li, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1.

[15] E. Grave, A. Joulin, and Q. Berthet, "Unsupervised alignment of embeddings with wasserstein procrustes," in *Proc. 22nd AISTATS*, Apr. 2019, pp. 1880–1890.

[16] A. Kalinowski and Y. An, "A survey of embedding space alignment methods for language and knowledge graphs," *arXiv preprint arXiv:2010.13688*, 2020.

[17] A. Imani, L. K. Senel, M. Jalili Sabet, F. Yvon, and H. Schuetze, "Graph neural networks for multiparallel word alignment," in *Findings of ACL 2022*, Dublin, Ireland, May 2022, pp. 1384–1396.

[18] N. Kanda, J. Wu, Y. Wu, X. Xiao, Z. Meng, X. Wang, Y. Gaur, Z. Chen, J. Li, and T. Yoshioka, "Streaming multi-talker ASR with token-level serialized output training," in *Proc. Interspeech*, 2022, pp. 3774–3778.

[19] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *Proc. ICASSP*, 2021, pp. 5904–5908.

[20] C. Wang, A. Wu, J. Gu, and J. Pino, "CoVoST 2 and Massively Multilingual Speech Translation," in *Proc. Interspeech 2021*, 2021, pp. 2247–2251.

[21] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. EMNLP: System Demonstrations*, 2018, pp. 66–71.

[22] A. Conneau, M. Ma, S. Khanuja, Y. Zhang, V. Axelrod, S. Dalmia, J. Riesa, C. Rivera, and A. Bapna, "Fleurs: Few-shot learning evaluation of universal representations of speech," in *Proc. SLT*, 2023, pp. 798–805.

[23] T. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," in *Proc. 13th IWSLT*, 2016.

[24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

[25] A. Graves, "Sequence transduction with recurrent neural networks," 2012.

[26] M. Post, "A call for clarity in reporting BLEU scores," in *Proc. 3rd WMT*, 2018, pp. 186–191.

[27] S. Papi, M. Gaido, M. Negri, and M. Turchi, "Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation," in *Proc. 3rd AutoSimTrans*, 2022, pp. 12–17.