INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Joint Speech Translation and Named Entity Recognition

Marco Gaido[†], *Sara Papi*[†‡], *Matteo Negri*[†], *Marco Turchi*[⋆*]

[†]Fondazione Bruno Kessler, Italy [‡]University of Trento, Italy [⋆]Independent Researcher
{mgaido, spapi, negri}@fbk.eu,marco.turchi@gmail.com

Abstract

Modern automatic translation systems aim at supporting the users by providing contextual knowledge. In this framework, a critical task is the output enrichment with information regarding the mentioned entities. This is currently achieved by processing the generated translations with named entity recognition (NER) tools and retrieving their description from knowledge bases. In light of the recent promising results shown by direct speech translation (ST) models and the known weaknesses of cascades (error propagation and additional latency), in this paper we propose multitask models that jointly perform ST and NER, and compare them with a cascade baseline. Experimental results on three language pairs (en-es/fr/it) show that our models significantly outperform the cascade on the NER task (by 0.4-1.0 F1), without degradation in terms of translation quality, and with the same computational efficiency of a plain direct ST model.

Index Terms: augmented translation, speech translation, named entity recognition, direct, multi-task

1. Introduction

Drawing inspiration from augmented reality, where real-world vision is complemented with overlaid relevant information, "augmented translation" [1] is an emerging research line aimed to enrich automatically-generated translations with semantic information by highlighting named entities (NEs) and key concepts (the focus of this work) and eventually linking them to external knowledge bases (an aspect we do not cover here). On one side, this can ease, speed up, and improve the generation of fluent and high-quality translations by professional translators and post-editors; on the other, it provides end users with additional information that may be needed to fully understand a sentence, especially in highly specialized domains.[1]

Current solutions rely on a cascade architecture comprising a text-to-text machine translation (MT) system whose output is fed to a NE recognition (NER) model [2]. No work has instead explored its application to speech-to-text translation (ST), and the possibility of jointly performing the ST and NER tasks with a single model, despite positive signals from related fields. Indeed, in the task of NER from speech, the traditional cascade approach – composed of an automatic speech recognition (ASR) system followed by an NER model – has been recently challenged by the competitiveness of direct models that perform the two tasks jointly [3, 4, 5, 6]. Similarly, in MT, multitask models that jointly perform MT and NER have been shown to improve NE accuracy without degrading translation quality [7].

In light of this and the competitive results of direct ST models [8, 9] compared to the conventional ASR+MT pipeline [10, 11], in this paper we address two research questions: **(1)** *Is the current cascade of an ST system (either direct ST or ASR+MT) followed by an NER tool better than performing the two tasks with a single model?* **(2)** *What are the effects on NE accuracy and translation quality of using a single multi-task model?*

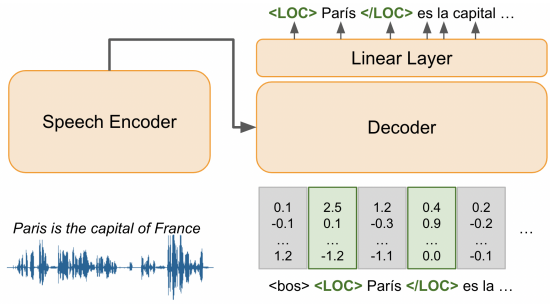

Figure 1: *Architecture of the inline solution. The additional tokens generated in the output are highlighted in green, and are passed to the decoder as all the other previous output tokens.*

To answer these questions, we explore different methods to jointly perform ST and NER.[2] Our experiments on three language pairs show that joint models significantly outperform the ST+NER cascade by 0.4-1.0 F1 in the NER task while being on par in terms of translation quality. Such improvement is achieved without introducing any computational overhead with respect to a plain ST model, making our solution remarkably more efficient than the cascade approach. This is directly reflected in the computational-aware latency in simultaneous ST scenarios, where our best model jointly performs ST and NER with the same latency (and quality) as an ST-only model.

2. Joint NER and ST

The easiest way to extract the NEs from a translation consists in applying an NER model on the output of the ST model. Henceforth, we refer to this approach as *cascade*, and we consider it as a baseline for comparison against our systems that jointly perform the two tasks with a single model. Our solutions – *inline*, and *parallel* – are described below:

Inline (Fig. 1). The vocabulary of the direct ST model is extended with tags that represent the start (e.g., `<LOC>`) and end (e.g., `</LOC>`) of the NE categories to be recognized which, in our case, are 18,[3] for a total of 36 entries. These tags are treated as all other tokens (subwords): they are predicted in the output sequence, and – together with the other tokens – fed to the

[1] https://intelligent-information.blog/en/augmented-translation-puts-translators-back-in-the-center/

[2] Code available at: https://github.com/hlt-mt/FBK-fairseq.

[3] The categories are those defined in the OntoNotes annotation [12].

10.21437/Interspeech.2023-1767

decoder as previous output tokens, informing it about the NE categories. This solution does not require architectural changes to the ST model but introduces additional overhead, especially at inference time, as the higher number of tokens to generate (due to the additional start/end NE tags) leads to an increase in the number of forward passes on the autoregressive decoder.

Parallel (Fig. 2). At each time step, two linear layers process in parallel the output of the last decoder layer: one maps the vectors to the vocabulary space to predict the next token as in standard ST models; the other maps the same vectors to the NE-category space to predict the NE category to which the token belongs, if any, or *O* (i.e. *OTHER*), if the token is not part of a NE. Although the second linear layer introduces additional parameters to train, its computational cost is negligible compared to that of the whole decoder. Moreover, this solution avoids the supplementary decoder forward passes required by the inline method. However, the potential drawback in comparison with the inline solution is that it cannot exploit information about the NE categories predicted for the previously generated tokens during translation. As we posit that this lack of information may cause performance degradation, we propose a variant of this method in which the embeddings of the previous output tokens are summed with learned embeddings of their corresponding NE categories.[4] This change requires only 19 additional embeddings to learn (one for each NE category, plus *O*) – a negligible number compared to the target vocabulary size – and a sum, hence producing no significant computational overhead. We refer to this variant as **Parallel + NE emb.**

3. Experimental Settings

Models. All our ST models are fed with 80 features extracted from the audio every 10ms with sample windows of 25ms. These sequences of features are processed by two 1D convolutional layers that reduce the sequence length by a factor of 4, before passing them to a 12-layers Conformer encoder [13], and a 6-layer autoregressive Transformer decoder [14]. We use 512 features with 1024 hidden neurons in the FFN for both the encoder and decoder. The target vocabulary is created with 8,000 BPE [15] merge rules. As a result, our models have 116M parameters that we optimize with label-smoothed cross-entropy loss [16] (0.1 smoothing factor) and an auxiliary CTC [17] loss on the output of 8th encoder layer with the transcript as the target to improve model convergence [18]. Moreover, we adopt CTC compression [19] to reduce the input dimension and speed up both training and inference. As optimizer, we use Adam [20], and the learning rate is initially increased for 20k steps up to 0.005 and then it decreases with the inverse squared root policy. We train on 4 K80 GPUs with 10k tokens per mini-batch and 8 as update frequency. The training stops after 5 epochs without loss decrease on the validation set, and average 5 checkpoints around the best. At inference time, we decode using beam search with 5 as beam size. The ASR model of our ASR+MT+NER pipelines is trained on the same data and with the same method described for the ST models. We rely on a multilingual BERT-based model,[5] openly-available in DeepPavlov [21], as NER system and, on the 1.3B-parameters distilled NLLB [22] as MT model.

Data and Evaluation Metrics. All models are trained on MuST-C [23] and Europarl-ST [24]. To train the joint NER and ST models, we automatically annotated the NEs on the target translations with the same NER tool used in our cascade approach, obtaining parallel training data with speech and the corresponding annotated translations without any manual intervention. Translation quality is evaluated with SacreBLEU[6] [25] on the Europarl-ST test set. Regarding NEs, instead, we measure three aspects on NEuRoparl-ST benchmark [26]: *1)* the generation of the correct translation, *2)* the recognition (or identification) of the NEs in the generated text, and *3)* the classification with the correct NE category. First, we use NE accuracy (case-insensitive, for the sake of comparison with previous work) to assess the ability to translate NEs. Second, we compute F1 to measure the ability in recognizing NEs, although F1 is also influenced by the NE translation quality, as it is computed by considering as correct only those NEs that are accurately translated and identified, but disregarding their category classification. As such, NEs that are poorly translated and recognized by a model penalize both recall and precision. The strict F1 definition mirrors the users' perception: in augmented ST, while unrecognized NEs are only a lack of help to the users, recognized but incorrect NEs are more harmful as they would distract them with unrelated and potentially misleading content. Lastly, we use classification accuracy to measure the percentage of NEs assigned to the correct category.

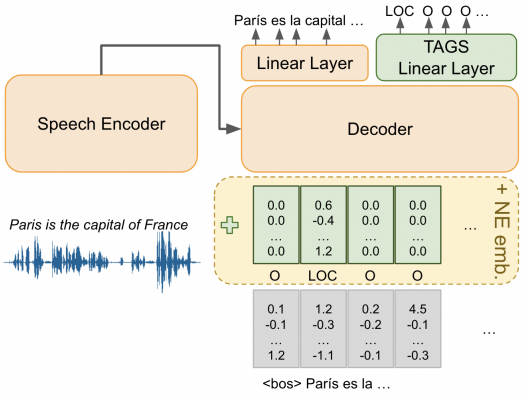

Figure 2: *Architecture of the parallel solution. The introduced linear layer (in green) is processed token-by-token in parallel with the other linear layer. In the + NE emb. variant (yellow dotted area), the previous tags are converted into embeddings that are summed to those of the corresponding previous tokens.*

4. Results

Translation Quality (Overall and at NE Level). First, to ensure the soundness of our experimental settings and, in turn, of our analysis, we compare our base direct ST model with recent works on Europarl-ST (Table 1). As shown by the results, our systems outperform, to the best of our knowledge, all recently published scores on the same benchmark. This confirms the strength of our models and the reliability of our results.

In Table 2, instead, we compare our cascade ST+NER and ASR+MT+NER baselines, the joint ST&NER inline and parallel methods, and the only previous work [26] that reports scores (NE accuracy) on the NEuRoparl-ST benchmark (using a direct ST system trained on a large amount of data). We can notice that, even though trained on fewer data, the direct ST models of our cascade ST+NER baselines compare favourably with the

[4]The beginning-of-sentence (*bos*) token is considered of *O* category.

[5]http://docs.deeppavlov.ai/en/master/features/models/bert.html

[6]case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0

Model	en-es				en-fr				en-it			
	BLEU	NE Acc	F1	Cat. Acc.	BLEU	NE Acc	F1	Cat. Acc.	BLEU	NE Acc	F1	Cat. Acc.
Prev. work [26]	37.7	71.4	-	-	30.1	67.3	-	-	26.0	67.3	-	-
Cascade (ASR+MT+NER)	37.3	71.0	47.4	89.9	**37.6**	68.8	44.6	90.2	26.2	64.8	42.0	87.5
Cascade (ST+NER)	37.9	71.9	49.1	89.8	36.2	69.2	44.8	90.2	28.3	66.5	44.5	88.8
Inline	37.9	**72.2**	**49.5**†‡	**90.1**	36.3	**69.6**	**45.6**†‡	90.2	28.3	66.9	**45.5**†‡	**89.4**
Parallel	**38.1**	71.9	48.1	89.5	36.1	69.0	44.5	**90.6**	**28.4**	**67.5**	43.9	89.1
+ NE emb.	38.0	72.1	**49.5**†‡	89.9	36.1	69.3	45.5†‡	90.4	28.2	67.3	45.4†‡	89.1

Table 2: *BLEU (↑), case-insensitive NE accuracy (↑), F1 (↑), and category classification accuracy (Cat. Acc., ↑) of previous work, our cascades (ASR+MT+NER and ST+NER) and the proposed joint ST&NER models. All results are the average of three runs.* † *indicates statistically significant improvements over ST+NER, and* ‡ *over parallel. A result is considered statistically significant if we can reject with 95% confidence the null hypothesis that the considered mean is not higher than the mean of the baseline with Student's t-test [27].*

Model	en-es	en-fr	en-it
ASR+MT [24]	28.0	23.4	-
NPDA-kNN-ST [28]	29.0	27.7	20.5
STR+KD [29]	-	29.3	-
NEuRoparl-ST [26]	37.7	30.1	26.0
Triangle Multi [30]	37.4	35.4	28.2
Ours	**37.9**	**36.2**	**28.3**

Table 1: *BLEU (↑) of our direct ST system in comparison with previous ST works on Europarl-ST.*

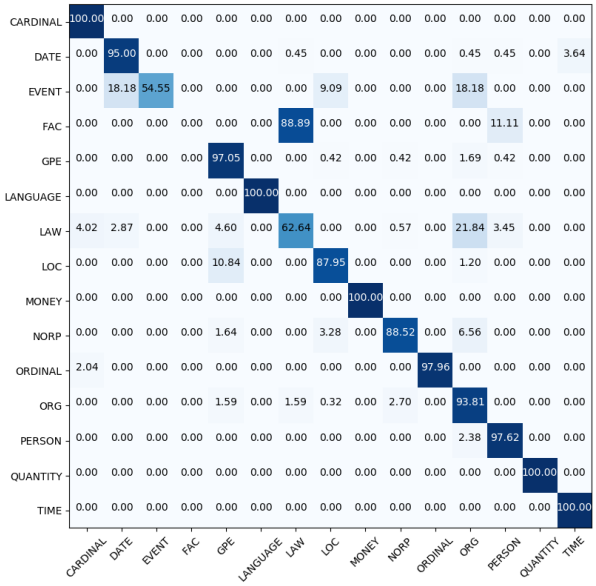

Figure 3: *Confusion matrix over the 15 NE categories with at least one NE correctly translated and recognized for the parallel + NE emb. system on en-es. On the y-axis, there are the true labels, while on the x-axis the predicted labels. The numbers are percentages computed on the y-axis.*

previous work not only in terms of translation quality (BLEU), but also in NE accuracy. In particular, the NE accuracy of our cascade ST+NER baseline is superior on average on the three language pairs, as the gains in en-es (+0.5) and en-fr (+1.9) are only partially balanced by the small drop in en-it (-0.8). In addition, the full cascade model (ASR+MT+NER) – despite leveraging NLLB, which is trained on a large amount of data, and the good ASR performance (12.5 WER, slightly better than the ASR trained on thousands of hours presented in [26]) – is inferior to the ST+NER baseline on all metrics, with the only exception of the en-fr BLEU. This further demonstrates the strength of our cascade ST+NER baseline.

Focusing on the comparison of the cascade and joint methods in performing the ST and NER tasks, we notice that the performance of both *inline* and *parallel* models are close in terms of translation quality, both generic (BLEU) and specific to NEs (NE accuracy), compared to the ST+NER baseline. The small differences among the scores of the various methods (up to 0.2 BLEU and up to 0.6 NE accuracy) are not consistent across language directions and are never statistically significant, thus being ascribable to fluctuations due to the inherent randomness of neural methods. We can conclude that the additional NER task does not bring any improvement to ST in terms of NE translation (in contrast with previous findings for MT [7]) but also does not degrade translation quality, as it could have happened since part of the model is dedicated to the additional task.

NE Recognition. When we consider the F1 metric, instead, the results highlight the differences between the various approaches. Our joint NER and ST approaches beat the cascade by a statistically significant margin on all language pairs (0.4-1.0 F1). This is surprising if we consider that the training data of the joint methods was generated with the NER system of the cascade approach, and highlights the strength of direct multitask systems. Among the joint solutions, the *inline* and *parallel + NE emb.* significantly outperform the *parallel* method, proving the importance of feeding the decoder with information about the NE category predicted for the previously generated tokens. The difference between *inline* and *parallel + NE emb.*, however, is very small (0.1, if any) and not statistically significant. These two methods can therefore be considered on par.

NE Classification. Lastly, all systems (joint and cascade) show a good ability in NE category classification. The accuracy differences range between 0.6 and 0.3, are not coherent across language pairs and are never statistically significant. Not only their overall performance is on par, but also their confusion matrices over the NE categories are basically the same on all language pairs. As an example, Fig. 3 reports the confusion matrix of the *parallel + NE emb.* model for en-es. The classification accuracy is high (87.95-100%) for all categories but three: facilities (*FAC*), events (*EVENT*), and names of laws (*LAW*). *FAC* and *EVENT* are very rare (19 and 9 occurrences in the test set), while *LAW* is more frequent (141 occurrences), thus representing the main source of classification errors. The root of this difficulty may lay in the nature of law names, which have high variability, are long, and frequent only in specific domains. At

last, another common source of errors is classifying *GPE* as location names, which is unsurprising as their categorization highly depends on the context in which they occur (e.g. *Europe* as a continent is a *LOC*, but in politics it can be a *GPE*).

5. Efficiency in Simultaneous ST

One known advantage of direct systems over cascade ones is their lower overall computational cost since they need a forward pass on only one model instead of two. For this reason, in applications where the computational cost is particularly critical, such as in simultaneous ST (SimulST), where it directly affects the output latency, direct ST systems obtain a significantly better latency-quality trade-off than ASR+MT solutions [11, 31].

However, in our case of ST and NER, the computational cost is not only determined by the choice of a cascade or a full direct system, but also by which of the two joint solutions is used. Indeed, the number of decoding steps (i.e. forward passes on the autoregressive decoder) required by the *inline* and *parallel* systems is different: the former method has to predict the start and end NE tags, requiring on average 7% more decoding steps on the Europarl-ST test set compared to a plain ST model and to the *parallel* systems, which do not introduce additional decoding steps. For this reason, we conclude our work by comparing the two best models (*inline* and *parallel + NE emb*) in the simultaneous setting using the popular wait-k [32] policy.

The wait-k policy consists in initially waiting for a predefined number of words (k) before starting to alternate between WRITE (emit a word) and READ (wait for more input audio) actions. Since the source is speech, the information about the number of words is not already present in the input, therefore a word detection strategy is applied to determine how many words have been pronounced at each time step. Here, we use an adaptive word detection strategy [33, 34] that estimates the number of words in an audio segment by counting them in the transcripts predicted by the CTC module trained on the encoded audio. The choice of this method is motivated by its favorable performance compared to other word detection strategies [35]. The wait-k policy is directly applied to offline-trained models without the need for any adaptation for the task, as this approach has been demonstrated to be competitive with the one adopting models specifically trained to work in simultaneous [36].

Evaluating the performance in simultaneous allows us to estimate the overhead introduced by the additional decoding steps of the *inline* model compared to the *parallel + NE emb.* one. In Fig. 4, we report the BLEU- and F1-latency curves computed on the outputs obtained by running the SimulEval tool [37] on the two joint NER&ST models for en-es (for the sake of brevity, we do not report the curves for en-fr and en-it that show the same trends). We also report the BLEU-latency curve of the direct ST-only model as a reference, while we do not show the cascade ST+NER as the computational cost (and hence, latency) is significantly higher. Latency is measured through computational-aware length-adaptive average lagging (LAAL) [38]. The k value of the wait-k policy is varied from 1 to 3 ($k = \{1, 2, 3\}$), in order to reach different latency regimes.

The curves show that the *parallel + NE emb* model has the same latency and quality of an ST-only model, despite the additional NER task to perform. The *inline* solution, instead, has similar quality but features a (slightly) increased latency, because of its higher computational cost. However, since the computational cost only accounts for a fraction of the latency (∼53% of the computational-aware LAAL is due to the wait time of the wait-k policy), and the computational difference is not large (∼5%), the gap between the two models is limited.

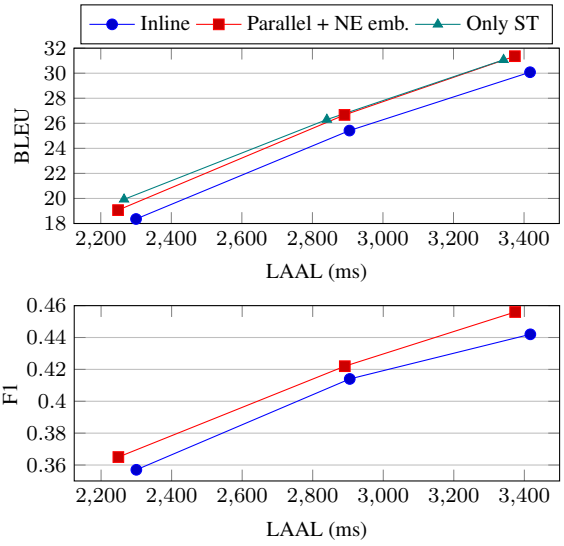

Figure 4: *BLEU- and F1-LAAL curves for en-es of the inline and parallel + NE emb solutions (we also include an ST-only system as reference). Each point corresponds to a different value of* $k = \{1, 2, 3\}$ *and is the average over three models.*

All in all, we can conclude that the *inline* model introduces a computational overhead that depends on the number of NEs detected in an utterance. On our test set, with 1,267 sentences, 30.6K words, and 1,638 NEs, we estimated as 5% its computational overhead in time compared to a base direct ST model and to our *parallel + NE emb.* solution. In light of the similar quality of *inline* and *parallel + NE emb.* systems, this difference – which may be larger in domains where NEs are more frequent, as news or molecular biology [39] – makes the *parallel + NE emb.* method our best solution overall.

6. Conclusions

We presented the first multitask models jointly performing speech translation and named entity recognition. First, we showed the importance of properly feeding information about the previously predicted NE tags, as done in the *inline* and *parallel + NE emb.* models. Second, and most importantly, we showed that our joint solutions consistently outperform a cascade system on the NER task (by 0.4-1.0 F1), while being on par in terms of translation quality. Lastly, we evaluated the computational efficiency of our methods and demonstrated that the *parallel + NE emb.* system, which does not introduce noticeable overhead with respect to a plain ST model, is more efficient than the *inline* method, besides being on par in terms of translation and NER quality. As such, it represents the most attractive solution to jointly perform ST and NER, especially in the simultaneous scenario where its computational-aware latency is the same as a single model performing the ST task only.

7. Acknowledgment

This work is part of the project Smarter Interpreting (https://smarter-interpreting.eu/) financed by CDTI Neotec funds. We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

8. References

[1] A. Lommel, "Augmented translation: A new approach to combining human and machine capabilities," in *Proc. of the 13th AMTA*, Boston, MA, 2018, pp. 5–12.

[2] A. D. De Palma, "Augmented translation powers up language services," *Common Sense Advisory Research*, 2017.

[3] S. Ghannay, A. Caubrière, Y. Estève, N. Camelin, E. Simonnet, A. Laurent, and E. Morin, "End-to-end named entity and semantic concept extraction from speech," in *Proc. of 2018 IEEE SLT Workshop*, 2018, pp. 692–699.

[4] H. Yadav, S. Ghosh, Y. Yu, and R. R. Shah, "End-to-End Named Entity Recognition from English Speech," in *Proc. of Interspeech 2020*, 2020, pp. 4268–4272.

[5] A. Caubrière, S. Rosset, Y. Estève, A. Laurent, and E. Morin, "Where are we in Named Entity Recognition from Speech?" in *Proc. of the 12th LREC*, Marseille, France, 2020, pp. 4514–4520.

[6] B. Chen, G. Xu, X. Wang, P. Xie, M. Zhang, and F. Huang, "Aishell-ner: Named entity recognition from chinese speech," in *Proc. of ICASSP 2022*, 2022, pp. 8352–8356.

[7] S. Xie, Y. Xia, L. Wu, Y. Huang, Y. Fan, and T. Qin, "End-to-end entity-aware neural machine translation," *Mach. Learn.*, vol. 111, no. 3, p. 1181–1203, mar 2022.

[8] A. Bérard, O. Pietquin, C. Servan, and L. Besacier, "Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation," in *NIPS Workshop on end-to-end learning for speech and audio processing*, Barcelona, Spain, 2016.

[9] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-Sequence Models Can Directly Translate Foreign Speech," in *Proc. of Interspeech 2017*, Stockholm, Sweden, Aug. 2017, pp. 2625–2629.

[10] L. Bentivogli, M. Cettolo, M. Gaido, A. Karakanta, A. Martinelli, M. Negri, and M. Turchi, "Cascade versus direct speech translation: Do the differences still make a difference?" in *Proc. of the 59th ACL*, Online, 2021, pp. 2873–2887.

[11] A. Anastasopoulos *et al.*, "FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN," in *Proc. of the 18th IWSLT 2021*, Bangkok, Thailand (online), 2021, pp. 1–29.

[12] R. Weischedel *et al.*, "OntoNotes Release 5.0," 2012. [Online]. Available: https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf

[13] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. of Interspeech 2020*, 2020, pp. 5036–5040.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proc. of NIPS 30*, Long Beach, California, 2017, pp. 5998–6008.

[15] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proc. of the 54th ACL*, Berlin, Germany, Aug. 2016, pp. 1715–1725.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proc. of 2016 IEEE CVPR*, Las Vegas, Nevada, United States, 2016, pp. 2818–2826.

[17] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proc. of the 23rd ICML*, Pittsburgh, Pennsylvania, 2006, pp. 369–376.

[18] P. Bahar, T. Bieschke, and H. Ney, "A Comparative Study on End-to-End Speech to Text Translation," in *Proc. of 2019 IEEE ASRU*, 2019, pp. 792–799.

[19] M. Gaido, M. Cettolo, M. Negri, and M. Turchi, "CTC-based compression for direct speech translation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online, Apr. 2021, pp. 690–696.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the 3rd ICLR*, San Diego, USA, 2015.

[21] M. Burtsev *et al.*, "DeepPavlov: Open-Source Library for Dialogue Systems," in *Proc. of ACL 2018, System Demonstrations*, Melbourne, Australia, 2018, pp. 122–127.

[22] M. R. Costa-jussà, J. Cross, O. Çelebi *et al.*, "No language left behind: Scaling human-centered machine translation," *arXiv preprint arXiv:2207.04672*, 2022.

[23] R. Cattoni, M. A. Di Gangi, L. Bentivogli, M. Negri, and M. Turchi, "Must-c: A multilingual corpus for end-to-end speech translation," *Computer Speech & Language*, vol. 66, pp. 101–155, 2021.

[24] J. Iranzo-Sánchez, J. A. Silvestre-Cerdà, J. Jorge, N. Roselló, A. Giménez, A. Sanchis, J. Civera, and A. Juan, "Europarl-st: A multilingual corpus for speech translation of parliamentary debates," in *Proc. of ICASSP 2020*, 2020, pp. 8229–8233.

[25] M. Post, "A Call for Clarity in Reporting BLEU Scores," in *Proc. of the 3rd WMT*, Belgium, Brussels, 2018, pp. 186–191.

[26] M. Gaido, S. Rodríguez, M. Negri, L. Bentivogli, and M. Turchi, "Is "moby dick" a Whale or a Bird? Named Entities and Terminology in Speech Translation," in *Proc. of the 2021 EMNLP*, Punta Cana, Dominican Republic, 2021, pp. 1707–1716.

[27] Student, "The probable error of a mean," *Biometrika*, vol. 6, no. 1, pp. 1–25, 1908.

[28] Y. Du, W. Wang, Z. Zhang, B. Chen, T. Xu, J. Xie, and E. Chen, "Non-parametric domain adaptation for end-to-end speech translation," in *Proceedings of the 2022 EMNLP*, Abu Dhabi, United Arab Emirates, Dec. 2022, pp. 306–320.

[29] T. K. Lam, S. Schamoni, and S. Riezler, "Sample, translate, recombine: Leveraging audio alignments for data augmentation in end-to-end speech translation," in *Proc. of the 60th ACL*, Dublin, Ireland, May 2022, pp. 245–254.

[30] M. Gaido, M. Negri, and M. Turchi, "Who Are We Talking About? Handling Person Names in Speech Translation," in *Proc. of the 19th IWSLT*, Dublin, Ireland, May 2022, pp. 62–73.

[31] A. Anastasopoulos *et al.*, "Findings of the IWSLT 2022 evaluation campaign," in *Proc. of IWSLT 2022*, Dublin, Ireland, May 2022, pp. 98–157.

[32] X. Ma, J. Pino, and P. Koehn, "SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation," in *Proc. of the 1st AACL-IJCNLP*, Suzhou, China, 2020, pp. 582–587.

[33] Y. Ren, J. Liu, X. Tan, C. Zhang, T. Qin, Z. Zhao, and T.-Y. Liu, "SimulSpeech: End-to-end simultaneous speech to text translation," in *Proc. of the 58th ACL*, Online, Jul. 2020, pp. 3787–3796.

[34] X. Zeng, L. Li, and Q. Liu, "RealTranS: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer," in *Findings of ACL-IJCNLP 2021*, Online, Aug. 2021, pp. 2461–2474.

[35] B. Zheng, K. Liu, R. Zheng, M. Ma, H. Liu, and L. Huang, "Simultaneous translation policies: From fixed to adaptive," in *Proc. of the 58th ACL*, Online, Jul. 2020, pp. 2847–2853.

[36] S. Papi, M. Gaido, M. Negri, and M. Turchi, "Does simultaneous speech translation need simultaneous models?" in *Findings of EMNLP 2022*, Abu Dhabi, United Arab Emirates, Dec. 2022.

[37] X. Ma, M. J. Dousti, C. Wang, J. Gu, and J. Pino, "SIMULEVAL: An evaluation toolkit for simultaneous translation," in *Proc. of EMNLP 2020*, Online, Oct. 2020.

[38] S. Papi, M. Gaido, M. Negri, and M. Turchi, "Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation," in *Proc. of the 3rd Workshop on Automatic Simultaneous Translation*, Online, 2022, pp. 12–17.

[39] C. Nobata, N. Collier, and J. Tsujii, "Comparison between tagged corpora for the named entity task," in *The Workshop on Comparing Corpora*, Hong Kong, China, 2000, pp. 20–27.