EFFICIENT USE OF END-TO-END DATA IN SPOKEN LANGUAGE PROCESSING

Yiting Lu, Yu Wang, Mark J.F. Gales

ALTA Institute / Engineering Department, University of Cambridge, UK
{ytl28,yw396,mjfg}@cam.ac.uk

ABSTRACT

For many challenging tasks there is often limited data to train the systems in an end-to-end fashion, which has become increasingly popular for deep-learning. However, these tasks can normally be split into multiple separate modules, with significant quantities of data associated with each module. Spoken language processing applications fit into this scenario, as they usually start with a speech recognition module, followed by multiple task specific modules to achieve the end goal. This work examines how the best use can be made of limited end-to-end training for sequence-to-sequence tasks. The key to improving the use of the data is to more tightly integrate the modules via embeddings, rather than simply propagating words between modules. In this work speech translation is considered as the spoken language application. When significant quantities of in-domain, end-to-end data is available, cascade approaches operate well. When the in-domain data is limited, however, tighter integration between modules enables better use of the data to be made. One of the challenges with tighter integration is how to ensure embedding consistency between the modules. A novel form of embedding-passing between modules is proposed that shows improved performance over both cascade and standard embedding-passing approaches for limited in-domain data.

Index Terms— spoken language processing, speech translation, embedding-passing, end-to-end training

1. INTRODUCTION

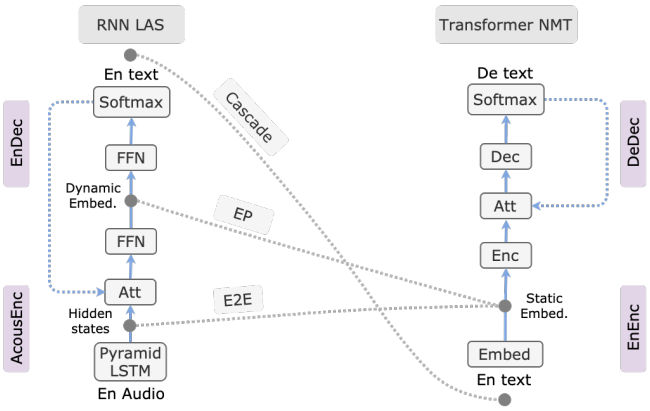
Modular systems are widely used for complex tasks when there is limited data for end-to-end training. Having multiple separate modules allows each sub-task to be trained individually, with significantly larger amount of data associated with each module. However, there are draw-backs to this approach. Modular systems operate in a sequential fashion which requires early decisions to be made, and any error made in early stages will have a ripple effect on downstream modules. The challenge, therefore, lies in mitigating error propagation through tightly integrated training and efficient use of limited end-to-end data. In this work, speech translation (ST) is considered as an example task, and a novel embedding-passing approach is proposed to allow tighter integration. This work is not intended to compete against the state-of-the-art ST systems [1, 2], which adopt ensembles, augmentation and much larger data sets. Instead, the focus is laid on exploring tighter integration in end-to-end training of modular systems, especially under a restrictive data scenario.

Traditional cascaded approach brings together an automatic speech recognition (ASR) module and a machine translation (MT) module. Past work explored connections through 1-best words [3, 4], n-best lists [5, 6] and lattices [7, 8]. Error propagation can be mitigated to some extent with an increasing level of complexity involved in the connection point. However, maintaining a rich search space for transcriptions is computationally expensive. Moreover, word-based discrete connection cannot carry over prosody information, and potentially incurs ambiguity in downstream MT. With recent advances in attention-based encoder decoder models [9, 10], more work has been done on integrating ASR and MT into a single model. Direct end-to-end model [11, 12, 13] does not rely on intermediate speech recognition, but it requires significant quantities of in-domain, end-to-end data to reach good performance. Another line of approaches still relies on explicit speech recognition, and adopts end-to-end trainable triangle structures [14, 15, 16]. Attention-passing [15] uses attention-generated context vectors to pass information between RNN-based sequence models, yet it is not compatible with the state-of-the-art transformer models where multiple layers of self-attention are used. Multi-task end-to-end model [16, 17] learns hidden representations of words from the ASR module, yet they cannot operate under zero-data scenario.

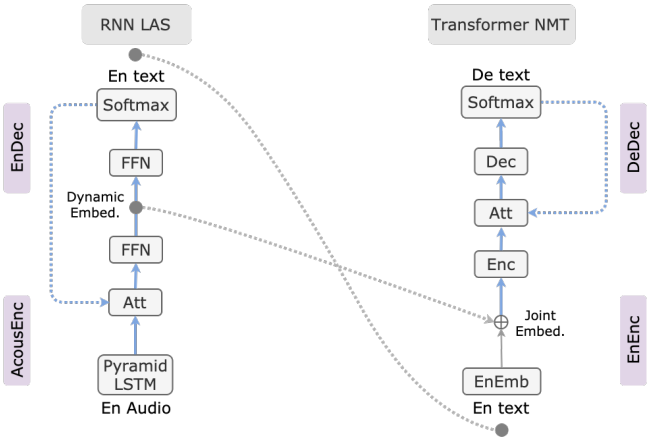
There is a notable trade-off between modeling power and data efficiency [18]. Models that are flexible to train on auxiliary MT corpora tend to be less sensitive to prosodies, whereas models that make full use of acoustic information are less adaptive to diverse corpora. Despite previous effort on analysing data efficiency [15, 16, 19], little has been done to contrast end-to-end trainable systems against cascaded structure under limited end-to-end data, neither do they take into account hybrid speech recognition, which potentially provides better ASR transcriptions compared to some end-to-end ASR models. This work studies integrated training of modular structures, contrasting models with different connections between ASR and MT modules. Baseline cascade connects through discrete words, while vanilla end-to-end model connects through acoustic hidden states. The former has restricted modeling power whereas the latter is low on data efficiency. Aiming to combine the best of both worlds, an embedding-passing model is proposed. It matches acoustically derived & word level embeddings and initialises the translation model with auxiliary data, thus achieving speech translation without in-domain training. Pure embedding-passing suffers from poor speech recognition, and consequently sabotages translation performance. To strike a balance between rich information flow and regularisation through words, another joint embedding-passing model is proposed to use both acoustic and word embeddings as module connection. Models are compared in the following aspects: impact of different levels of in-domain data availability; data efficiency; translation performance when high quality ASR transcriptions are used.

2. MODELS

Define speech translation task with audio sequences $\mathbf{v}_{1:T}$, speech transcriptions $x_{1:N}$ and target translations $y_{1:L}$. Due to the lack of end-to-end ST corpora, where all three components are available, additional auxiliary ASR $\{\mathbf{v}_{1:T}, x_{1:N}\}$ and MT $\{x_{1:N}, y_{1:L}\}$ corpora are used for pre-training. The focus of this work is to design models



(a) Cascade (Casc), End2End (E2E), Embedding-Passing (EP)



(b) Joint Embedding-Passing (EP-J)

Fig. 1: Various models with different connection points between ASR and MT

such that they can be well initialised with ASR, MT corpora, and more importantly, quickly adapt to the target domain with limited quantities of fine-tuning ST data. Various models shown in Figure 1 are discussed below.

Cascade (Casc) The vanilla Casc model consists of an RNN-based Listen-attend-spell (LAS) [20] and a transformer-based NMT [10]. In the LAS module, audio sequences $\mathbf{v}_{1:T}$ are first mapped into time-reduced hidden states through pyramidal LSTM:

$$\mathbf{h}_{1:\tau} = \text{pBLSTM}(\mathbf{v}_{1:T}) \tag{1}$$

Acoustic-level $\mathbf{h}_{1:\tau}$ are converted into word-level dynamic embeddings $\mathbf{e}^d_{1:N}$ through an attention mechanism, and then mapped into words (FFN: feed-forward network):

$$\mathbf{c}_n = \text{Att}(\mathbf{s}_n, \mathbf{h}_{1:\tau}) \quad \mathbf{s}_n = \text{RNN}(\mathbf{s}_{n-1}, x_{n-1}, \mathbf{c}_{n-1}) \tag{2}$$

$$\mathbf{e}^d_n = \text{FFN}([\mathbf{c}_n, \mathbf{s}_n]) \tag{3}$$

$$x_{1:N} = \text{argmax}(\sigma(\text{FFN}(\mathbf{e}^d_{1:N}))) \tag{4}$$

The transcribed word tokens $x_{1:N}$ are passed onto the NMT module, mapped into static embeddings $\mathbf{e}^s_{1:N}$, and then translated to be $y_{1:L}$:

$$\mathbf{e}^s_{1:N} = \text{Emb}(x_{1:N}) \tag{5}$$

$$y_{1:L} = \text{Transformer}(\mathbf{e}^s_{1:N}) \tag{6}$$

The cascade structure connects speech recognition and machine translation through words. Such discrete connection allows hybrid ASR transcripts to be used as an alternative to LAS. It is convenient to pre-train each module on ASR and MT corpora, and yet fine-tuning on ST data has to be done in a modular fashion as well. As are all cascade style models, flexibility is achieved through compromising modeling power. Casc model optimises two modules separately and thus tends to propagate ASR errors to the NMT module. To mitigate this issue, several other approaches are proposed seeking to allow softer and tighter connections.

End-to-end (E2E) As shown in Figure 1(a), the E2E model omits the intermediate speech recognition stage. Acoustic-level hidden states $\mathbf{h}_{1:\tau}$ are directly fed into the translation model:

$$y_{1:L} = \text{Transformer}(\mathbf{h}_{1:\tau}) \tag{7}$$

Compared to Casc, connection through hidden states provides a richer feature space, which potentially encapsulates both acoustic and textual information. On the other hand, the end-to-end nature prohibits the model from operating without ST data, and it requires large quantities of end-to-end data for training.

Embedding-Passing (EP) To obtain a higher level of abstraction, the EP model passes word-level embeddings rather than acoustic-level hidden states. It enforces the acoustically derived dynamic embeddings $\mathbf{e}^d_{1:N}$ to match with the static embeddings $\mathbf{e}^s_{1:N}$, and uses this embedding as the connection between ASR and MT. It uses $\mathbf{e}^s_{1:N}$ to initialise the MT module, and uses $\mathbf{e}^d_{1:N}$ for speech translation:

$$y_{1:L} = \text{Transformer}(\mathbf{e}^d_{1:N}) \tag{8}$$

Static embeddings are derived using one-to-one mapping from words, thus called static; whereas dynamic embeddings follow many-to-one mapping since the same word can be pronounced very differently. The matched embedding connection allows auxiliary data to be used, and maintains a rich information flow that is particularly useful during fine-tuning.

Joint Embedding-Passing (EP-J) In EP, embedding matching reaches a compromise between richness of acoustics and robustness of text. However, the many-to-one nature of the dynamic embeddings causes a systematic mismatch from the static embeddings. Therefore in EP-J, to loosen the constraints posed by embedding matching, the model decouples the static and dynamic embeddings, and simply concatenates the two to yield a joint embedding for translation:

$$y_{1:L} = \text{Transformer}(\mathbf{W}[\mathbf{e}^s_{1:N}, \mathbf{e}^d_{1:N}]) \tag{9}$$

where $\mathbf{W}$ is a transformation matrix setting the dimension of the joint embedding. In MT pre-training, there is no dynamic embedding from the acoustics, and an average dynamic embedding $\bar{\mathbf{e}}^d$, obtained from ASR pre-training, is used to initialise the transformer.

3. EXPERIMENTAL SETUP

This work focuses on En-De speech translation. The three main corpora used in the experiments are summarised in Table 1. WMT17 En-De has over 4M sentences in total, but only 10% was used for pre-training purposes to save computational time. All results reported are evaluated on MuST-C tst-COMMON with case-sensitive BLEU. On the audio side, 40 dimensional filter bank features are extracted at 10ms frame rate. English transcriptions are lower-cased, punctuation-normalised using Moses toolkit [21], and further tokenised following byte-pair encoding [22] with a 40k vocabulary trained on MuST-C. German translations are kept true-cased.

Corpus	Task	#Sentences
TED-LIUM3 [23]	ASR	268k
WMT17-P En-De [24]	MT	400k
MuST-C En-De [25]	ST	229k

Table 1: Corpora

The LAS model has an encoder of 1x256D BLSTM and 3x256D pLSTM layers, reducing acoustic sequence lengths by 8. The decoder then uses bilinear attention, followed by 3x512D uniLSTM layers. Speaker level normalisation and SpecAug [26] were enabled in LAS training. The NMT transformer is a standard base-sized model with 512D hidden states, 6 encoder, and 6 decoder layers. LAS uses BPE tokens as targets, while NMT uses character-level targets. Both static and dynamic embeddings are 512D. Casc models can be trivially trained module-by-module. In E2E, the pLSTMs are trained with LAS target, which are then fixed during the transformer training. In EP, the matching between static and dynamic embeddings was achieved as such: (1) initialise dynamic embeddings with LAS target (2) freeze LAS and train the static embedding-mapping function (3) free up all parameters associated with the two embeddings and enforce an L2 loss with ASR corpora. Once the embeddings are matched and fixed, the transformer can be initialised on MT data, and later all parameters are fine-tuned on ST data. Training of EP-J is simpler since dynamic and static embeddings are no longer coupled. An averaged dynamic embedding is used to initialise the transformer on MT data, and the fine-tuning stage is similar to EP. When manual transcriptions are available, fine-tuning always uses both LAS and NMT objectives. When they are not, LAS-related parameters are fixed during fine-tuning, and only the NMT objective is imposed. All models are trained using Adam optimiser [27] with a batch size of 256, dropout 0.2, and a learning rate of 0.001 with gradient clipping. A pytorch implementation is available for download [1]. Translations were generated using beam search with a beam width of 5, and models are averaged over the 5 best checkpoints under each setup.

	ASR			ST (BLEU↑)	
Name	Models	Data	WER↓	Base	Tune
ASRT	Hybrid	TED	10.58	13.58	23.85
ASRM	Hybrid	MuSTC	7.32	14.41	25.45
LAST	LAS	TED	35.20	9.60	15.97
LASM	LAS	MuSTC	20.99	12.25	20.54

Table 2: Casc baselines on different ASR transcriptions

Table 2 shows 4 ASR systems with their corresponding Casc performance before and after fine-tuning on ST data. On the ASR side, both LAS and hybrid models were trained on TED-LIUM3 corpus (out-of-domain) and MuST-C corpus (in-domain). Hybrid ASR systems adopted lattice-free maximum mutual information (LF-MMI) factorised time-delay neural network (TDNN-F) acoustic model [28] followed by a 3-gram decoding. On the NMT side, the base model was trained on WMT17-P, and the fine-tuned model was further trained on MuST-C. Compared to LAS models, hybrid ASR systems obtained 25% and 13% lower WER when trained on TED and MuST-C respectively. They consequently led to higher ST BLEU in both base and fine-tuned cases. As expected, the highest BLEU score was achieved by combining ASRM transcriptions with the fine-tuned NMT.

[1] https://github.com/EdieLu

4. RESULTS

4.1. In-domain Data Availability

ST Data	None	Audio-De	Audio-En-De
E2E	-	-	19.29
Casc	9.60	16.56	20.54
EP	7.97	**18.84**	22.56
EP-J	9.60	16.24	**23.25**

Table 3: ST BLEU under 3 levels of in-domain data availability

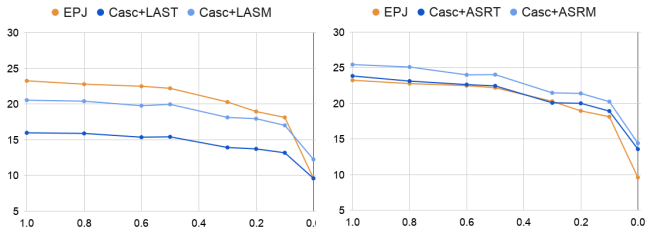
This section investigates the impact of in-domain data availability on different models. The key difference that distinguishes various models is the information flow passed from the acoustic side to the translation module. Casc uses discrete words, E2E uses acoustic-level hidden states, EP and EP-J use matched embeddings and joint embeddings respectively. To directly contrast the impact of different connection points, LAS style ASR is considered across all models, and hybrid ASR will be discussed in later sections.

Three conditions of data availability are considered here: (1) zero in-domain ST data; (2) ST data without manual En transcriptions (Audio-De); (3) ST data with both En transcriptions and De translations (Audio-En-De). All models were initialised with ASR, MT data, and were then fine-tuned towards the ST domain, except for E2E which was directly trained on ST data. During fine-tuning, when manual transcriptions were not available, the NMT component in Casc was trained with LAST transcriptions. EP and EP-J models were trained with LAS-related parameters being fixed. Due to restricted data availability, Casc was decoded using LAST transcriptions under condition (1)(2); and LASM under (3).

Table 3 shows that Casc and EP-J are similar under zero in-domain data, EP-J performs the best under full data, whereas EP is the best when manual transcriptions aren't available. In the base case, EP falls short mainly because of the imperfect matching between static and dynamic embeddings. The dynamic embedding can be seen as a noisy, perturbed version of the static embedding, since different pronunciations can point to the same word. Embedding mismatch is a systematic issue before any in-domain training takes place. In comparison, EP-J loosens the restriction and makes use of both dynamic and static embeddings. Before fine-tuning, the transformer NMT in EP-J is initialised using averaged dynamic embeddings, which explains why the model is not better than Casc since it's yet to benefit from richer acoustic context. After fine-tuning, EP-J is able to gain robustness through regularisation provided by static embeddings, meanwhile retaining richness of the acoustic features from dynamic embeddings, and thus achieve the best performance among all. However, when the in-domain speech transcriptions are not available, which means LAS errors are passed down to the NMT module during fine-tuning, EP outperforms cascade and EP-J by over 2 BLEU points. This confirms that having a soft dynamic connection in a modular structure helps to mitigate error propagation.

Most ST corpora provides audio, transcriptions and translations as it is natural to produce speech transcripts first, then generate translations in the annotation process. Therefore in the following sections, we focus on comparing Casc and EP-J models as they are most competitive under realistic settings.

4.2. Data Efficiency



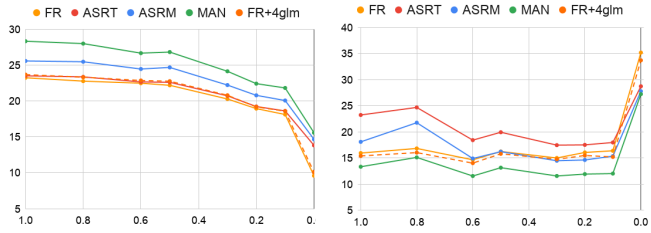
(a) Casc+LAS vs EP-J (b) Casc+Hybrid ASR vs EP-J

Fig. 2: Data efficiency: BLEU - data ratio
(All models fine-tuned with MAN transcriptions)

Last section discussed two ends of the data spectrum, this part further explores how Casc and EP-J behave under different quantities of in-domain data. To investigate data efficiency, Casc and EP-J are both initialised with the same ASR, MT corpora, and then fine-tuned with a sweep through ST data (from 100% to 0%). Their respective performances are recorded at each data ratio. Figure 2(a) compares Casc and EP-J models when they both adopt the same LAS-style ASR module. For Casc, both LAST and LASM transcriptions are used, setting the lower (out-of-domain LAS) and upper (in-domain LAS) bounds respectively. EP-J and Casc behave similarly in the zero data region, and yet EP-J adapts to the target domain much more quickly with an increasing amount of in-domain ST data, starting to outperform the Casc upper bound at only 10% data level. The main difference between Casc and EP-J is the additional dynamic embeddings incorporated into the source side of the MT module. This result confirms that dynamic embeddings do provide downstream tasks with a richer acoustic context, allowing more efficient domain adaptation with as few as 23k sentence pairs.

However, it is not fair to only compare EP-J with Casc models that are using LAS produced transcriptions. One of the benefits of cascaded structures is the flexibility in improving each individual module. To construct a stronger cascaded baseline, two hybrid ASR systems are used to provide better speech transcriptions, with ASRT (trained out-of-domain) and ASRM (trained in-domain) setting the lower and upper bounds. Figure 2(b) shows that when Casc model adopts hybrid ASR transcriptions, its lower bound performs at the similar level as EP-J, and the upper-bound outperforms EP-J. This can be accounted for by the huge performance gap between LAS (over 15% WER) and hybrid ASR (7.32% WER).

4.3. Improved LAS back history



(a) ST BLEU - data ratio (b) LAS WER - data ratio

Fig. 3: EP-J decoded with various LAS back history
(FR: back history generated under free running)

As mentioned above, EP-J suffers from poor LAS performance, and consequently leads to inferior speech translation quality. One of the advantages of having an explicit speech recognition stage in EP-J, compared with pure E2E, is the flexibility of incorporating external sources of information. In EP-J, it is possible to amend LAS generated hypotheses and propagate its impact through LAS attention with the modified back history. One way of improving LAS transcriptions is to decode with external language model. As shown in Figure 3 (dotted orange line), by adding an explicit 4-gram LM trained on TED, LAS WER dropped slightly and led to some increase in ST BLEU. Another more effective approach is to directly replace LAS hypotheses with external ASR transcriptions. In general, better back history leads to lower LAS WER, and thus higher ST BLEU score. However, as is shown in Figure 3(b), although the transcription quality used for back history follows FR $<$ ASRT $<$ ASRM $<$ MAN, the generated LAS hypotheses gives ASRT $<$ ASRM $<$ FR $<$ MAN, ordered from the highest to the loweset WER. The inversion between FR and ASRT/M is mainly because of the mismatch between LAS predicted and hybrid ASR generated transcripts. Despite the worse WER, with the in-domain ASRM transcriptions being used, EP-J improved by around 2 BLEU points throughout the sweep. It is also worth notice that, even when manual transcriptions are fed in as LAS back history, WER of LAS hypotheses are still above 10% (3 points higher than ASRM), and such suboptimal LAS has already led to 28.34 BLEU (with data ratio = 1.0), which is way higher than the Casc highest 25.45 BLEU. Therefore, it is safe to speculate that when LAS reaches a comparable WER as hybrid ASRs, EP-J will achieve much better ST performance.

Figure 4 compares EP-J with Casc, both making use of hybrid ASR transcriptions in their own ways. Under zero data, true performance is closer to the lower bound (ASRT), whereas full data leads to the upper bound (ASRM). It is expected that, compared to the vanilla Casc, it is more challenging for EP-J to indirectly propagate ASR transcriptions through LAS back history. In the low data region, Casc is better, suggesting that regularisation via words still is an effective way of handling out-of-domain corpus. After training on 40% of in-domain data or more, EP-J starts to outperform Casc model even though the LAS performance (over 15% WER) is nowhere near ASRM. This shows that joint embedding does help make the model more robust against poor LAS performance, and supports efficient adaptation towards the target domain.

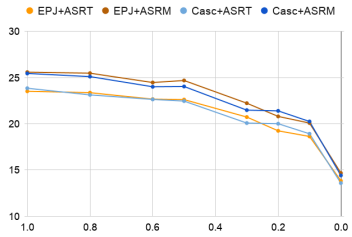


Fig. 4: Casc vs EP-J with Hybrid ASR Transcriptions

5. CONCLUSIONS

This work takes a close look at different connections in modular structures, and their impact on data efficiency in end-to-end training. Hard, discrete connection poses strong regularisation on complex systems and performs well under low data scenario. Soft, embedding-like connection provides richer context and adapts well under fine-tuning. Combination of the two seeks compromise between regularisation and richness, and is proved useful when competing with a challenging baseline.

6. REFERENCES

[1] Marco Gaido, Mattia Antonino Di Gangi, Matteo Negri, and Marco Turchi, "End-to-end speech-translation with knowledge distillation: Fbk@ iwslt2020," *arXiv preprint arXiv:2006.02965*, 2020.

[2] Nikhil Kumar Lakumarapu, Beomseok Lee, Sathish Reddy Indurthi, Hou Jeung Han, Mohd Abbas Zaidi, and Sangha Kim, "End-to-end offline speech translation system for iwslt 2020 using modality agnostic meta-learning," in *Proceedings of the 17th International Conference on Spoken Language Translation*, 2020, pp. 73–79.

[3] Fred WM Stentiford and Martin G Steer, "Machine translation of speech," *Speech and language processing*, pp. 183–196, 1988.

[4] Ngoc-Quan Pham, Thai-Son Nguyen, Thanh-Le Ha, Juan Hussain, Felix Schneider, Jan Niehues, Sebastian Stüker, and Alexander Waibel, "The iwslt 2019 kit speech translation system," in *Proceedings of the 16th International Workshop on Spoken Language Translation*, 2019.

[5] M Woszczyna, N Coccaro, A Eisele, A Lavie, A McNair, T Polzin, I Rogina, CP Rosé, T Sloboda, M Tomita, et al., "Recent advances in janus: a speech translation system," 1993.

[6] Alon Lavie, Donna Gates, Marsal Gavalda, Laura Mayfield Tomokiyo, Alex Waibel, and Lori Levin, "Multi-lingual translation of spontaneously spoken language in a limited domain," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

[7] Tanja Schultz, Szu-Chen Jou, Stephan Vogel, and Shirin Saleem, "Using word latice information for a tighter coupling in speech translation systems," in *Eighth International Conference on Spoken Language Processing*, 2004.

[8] Pei Zhang, Boxing Chen, Niyu Ge, and Kai Fan, "Lattice transformer for speech translation," *arXiv preprint arXiv:1906.05551*, 2019.

[9] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[11] Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," *arXiv preprint arXiv:1612.01744*, 2016.

[12] Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater, "Low-resource speech-to-text translation," *arXiv preprint arXiv:1803.09164*, 2018.

[13] Mihaela C Stoian, Sameer Bansal, and Sharon Goldwater, "Analyzing asr pretraining for low-resource speech-to-text translation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7909–7913.

[14] Antonios Anastasopoulos and David Chiang, "Tied multitask learning for neural speech translation," *arXiv preprint arXiv:1802.06655*, 2018.

[15] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel, "Attention-passing models for robust and data-efficient end-to-end speech translation," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 313–325, 2019.

[16] Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou, "Bridging the gap between pre-training and fine-tuning for end-to-end speech translation.," .

[17] Shun-Po Chuang, Tzu-Wei Sung, Alexander H Liu, and Hung-yi Lee, "Worse wer, but better bleu? leveraging word embedding as intermediate in multitask end-to-end speech translation," *arXiv preprint arXiv:2005.10678*, 2020.

[18] Matthias Sperber and Matthias Paulik, "Speech translation and the end-to-end promise: Taking stock of where we are," *arXiv preprint arXiv:2004.06358*, 2020.

[19] Ye Jia and Johnson et al., "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7180–7184.

[20] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[21] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al., "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

[22] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[23] François Hernandez and Nguyen et al., "TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *International Conference on Speech and Computer*. Springer, 2018, pp. 198–208.

[24] Bojar Ondrej, Rajen Chatterjee, Federmann Christian, Graham Yvette, Haddow Barry, Huck Matthias, Koehn Philipp, Liu Qun, Logacheva Varvara, Monz Christof, et al., "Findings of the 2017 conference on machine translation (wmt17)," in *Second Conference onMachine Translation*. The Association for Computational Linguistics, 2017, pp. 169–214.

[25] Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi, "Must-C: a multilingual speech translation corpus," in *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019, pp. 2012–2017.

[26] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[27] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohamadi, and Sanjeev Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," 2018, pp. 3743–3747.