The University of Helsinki submission to the IWSLT2020 Offline Speech Translation Task

Raúl Vázquez, Mikko Aulamo, Umut Sulubacak, Jörg Tiedemann

University of Helsinki
{name.surname}@helsinki.fi

Abstract

This paper describes the University of Helsinki Language Technology group's participation in the IWSLT 2020 offline speech translation task, addressing the translation of English audio into German text. In line with this year's task objective, we train both cascade and end-to-end systems for spoken language translation. We opt for an end-to-end multitasking architecture with shared internal representations and a cascade approach that follows a standard procedure consisting of ASR, correction, and MT stages. We also describe the experiments that served as a basis for the submitted systems. Our experiments reveal that multitasking training with shared internal representations is not only possible but allows for knowledge-transfer across modalities.

1 Introduction

An effective solution for performing spoken language translation (SLT) must deal with the evident challenge of transferring the implicit semantics between audio and text modalities. An end-to-end SLT system must hence appropriately address this problem while simultaneously performing accurate machine translation (MT) (Sulubacak et al., 2018).

In last year's IWSLT challenge, both end-to-end and cascade systems yielded similar results (Niehues et al., 2019). For this reason, this year's IWSLT offline speech translation challenge focuses on whether *"the cascaded solution is still the dominant technology in spoken language translation"*. For our participation on this task, we train both cascade and end-to-end systems for SLT. For the end-to-end system, we use a multimodal approach trained in a multitask fashion, which maps the internal representations of different encoders into a shared space before decoding. For the cascade approach, we use a pipeline of three stages: (i) automatic speech recognition (ASR), (ii) punctuation and letter-case restoration, and (iii) MT.

We focus on exploiting the knowledge-transfer capabilities of a multitasking architecture based on language-specific encoders-decoders (Lu et al., 2018; Schwenk and Douze, 2017; Luong et al., 2016). This idea has been proposed and studied in the multilingual scenario (Vázquez et al., 2020; Subramanian et al., 2018; Firat et al., 2017), however, we adapt it to be used in a multimodal scenario. Regarding different modalities (in this case, audio and text) as different languages when training the model, allows us to employ a cross-modal intermediate shared layer for performing SLT in an end-to-end fashion. By jointly training this layer, we aim for the the model to combine the semantic information provided in the text-to-text MT tasks with the ability to generate text from audio in the ASR tasks.

2 Proposed Systems

End-to-end SLT

We use an inner-attention based architecture proposed by Vázquez et al. (2020). In a nutshell, it follows the conventional structure of an encoder-decoder model of MT (Bahdanau et al., 2015; Luong et al., 2016) enabled with multilingual training by incorporating language-specific encoders and decoders trainable with a language-rotating scheduler (Dong et al., 2015; Schwenk and Douze, 2017), and an intermediate shared inner-attention layer (Cífka and Bojar, 2018; Lu et al., 2018). We implement our model on top of an OpenNMT-py (Klein et al., 2017) fork, which we make available for reproducibility purposes.[1]

The text encoders and the decoders (always text output) are transformers (Vaswani et al., 2017). We implement the transformer-based audio encoders

[1] https://github.com/Helsinki-NLP/OpenNMT-py/tree/att-brg

inspired by the SLT with tied layer structure architecture from Tu et al. (2019) and the R-Transformer from Di Gangi et al. (2019b). It consists of n CNN layers; the first one taking k stacked Mel filterbank features as input channels, and the following ones 32 input channels. Afterwards, a linear layer corrects the shape of the embeddings and is concatenated with the positional embeddings to be fed as input to m transformer layers.

Given the multimodal nature of the task, we modified the source-target rotating scheduler. Instead of a uniform distribution over the language pairs, we propose using a weighted sampling scheme based on the inverse of the batch size of the modalities. This modification allows us to have a more balanced training because audio inputs tend to be considerably longer than text inputs, and a transformer-based encoder could not possibly handle the 4096 tokens conventionally used as the ad-hoc choice of batch size for a text-based transformer.

Cascade approach

The ASR stage of our pipeline is trained with an S-Transformer (Di Gangi et al., 2019b); an adaptation of the transformer architecture to end-to-end SLT. The encoder in this architecture makes it possible to process audio features. It consists of two 2-dim CNN-blocks meant to downsample the input, followed by two 2-dim self-attention layers to model the long-range context, an attention layer that concatenates its output with the positional encodings of the input, and six transformer-based layers.

The output of the ASR stage is followed by the **restoration stage** for repunctuation and letter case restoration. Since the training data for the ASR model mixes different training sets with different formatting, the raw output from the ASR block can have stylistic differences from the input seen during the training of the translation stage. The restoration stage involves the use of an auxiliary transformer-based MT model to perform "intralingual translation" from lowercased text without punctuation into fully-cased and punctuated text. Stripping punctuation on the ASR output, converting the text to lowercase, and processing the result through this stage ensures that the output conforms to the same format that the translation stage was optimized for.

As the last step, **the translation stage** uses another transformer to translate the processed ASR output to German. Both this transformer model and the one used in the restoration stage are based on the freely available Marian NMT implementation (Junczys-Dowmunt et al., 2018). Our configuration uses a learning rate of 0.0003 with linear warmup through the first 16 000 batches, decaying afterwards. The decoder normalizes scores by translation length (normalization exponent of 1.0) during beam search. All other options use the default values.

3 Data Preprocessing

The MT, ASR and end-to-end SLT systems have been trained on different subsets of the allowed training corpora.

Corpora	# utterances	Length
Europarl-ST	40,141	89 hrs
IWSLT2018	166,214	271 hrs
How2	189,366	297 hrs
MuST-C	264,036	400 hrs
Mozilla Common Voice	854,430	1,118 hrs

Table 1: Size of audio data used.

Data for the end-to-end SLT system. We use Europarl-ST (Iranzo-Sánchez et al.), IWSLT2018 (Niehues et al., 2019) and MuST-C (Di Gangi et al., 2019a), a total of 433k utterances after cleaning some corrupt files or with other problems in the sampling. We extracted 80-dimensional Mel filterbank features for each sentence-like segment using our own implementation.

Text data for the end-to-end SLT system. For the text data of the multimodal end-to-end SLT system, we use a total of ∼51M sentence pairs from corpora specified in Table 2. Instead of using all of this data, we first filter out noisy translations. OpenSubtitles2018, which consists of subtitle translations, and corpora gathered by crawling the internet, Common Crawl and ParaCrawl, are especially likely to contain noisy data. For filtering the corpora, we utilize OpusFilter (Aulamo et al., 2020), a toolbox for creating clean parallel corpora.

First, we extract six feature values for each of the sentence pairs. In particular, we apply the following features: CharacterScore, CrossEntropy, LanguageID, NonZeroNumeral, TerminalPunctuation and WordAlign, each of which is defined in Aulamo et al. (2020). Secondly, we train a logistic regression classifier based on those features.

The classifier is trained only on WIT^3, MuST-C, Europarl-ST and IWSLT18, which are multimodal datasets with speech-to-text and text-to-text data. This allows the system to adapt to text translations that are associated with speech translations. Finally, we use the classifier to assign a cleanness score ranging from 0 to 1 for all sentence pairs in all corpora. The data is then ranked based on the cleanness score, after which a portion of noisy pairs is removed from the tail. Our preliminary translation experiments showed that removing up to 40% of the data improves the translation quality, leaving us ∼30.5M sentence pairs of training data, which are then used in all our end-to-end experiments.

Corpora	**# sentences**
WIT^3	196,112
MuST-C train	229,703
Rapid 2019	1,480,789
Europarl v9	1,817,763
OpenSubtitles2018	11,621,073
News Commentary v14	365,340
Common Crawl	2,399,123
Europarl-ST	32,628
WikiTitles	1,305,078
IWSLT2018	171,025
ParaCrawl v3	31,360,203
Total	50,978,837
Filtered	30,540,267

Table 2: Text training data used for end-to-end systems.

Audio for the cascade system. We have extracted 40-dimensional Filterbank features with speaker normalization for each sentence-like segment of the MuST-C, How2 (Sanabria et al., 2018) and Mozilla Common Voice (Ardila et al., 2019) corpora using XNMT (Neubig et al., 2018). After getting rid of audio files that were too short (less than 0.4 seconds), corrupted or no longer available for download form Youtube, some 1.2M clean utterances remained for training the ASR system, and 30k for validation.

On the target side, we use two contrastive preprocessing pipelines:

i) the same subword segmentation used for the MT system

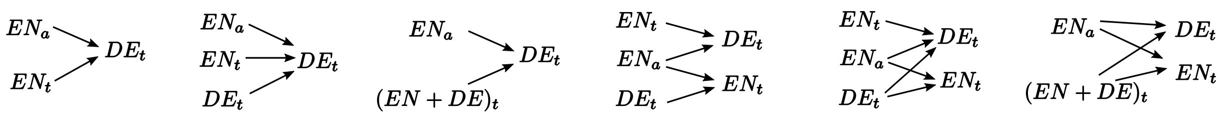
```
_it _& apos ; s _a _lobster _made
_of _play d ough _that _& apos ; s
_afraid _of _the _dark _.
```

ii) character level segmentation

```
I t <space> ' s <space> a <space>
l o b s t e r <space> m a d e <space>
o f <space> p l a y d o u g h <space>
t h a t <space> ' s <space> a f r a i d
<space> o f <space> t h e <space>
d a r k <space> .
```

Text data for the cascade system. In our SLT pipeline, the data we applied for our restoration and translation models have some overlap and some differences. For training, both models use the text data from the IWSLT 2018 speech translation corpus, the MuST-C training set, News Commentary v14, Europarl v9, and Rapid 2019. The translation model also uses data from the OpenSubtitles2018 dataset, which the restoration model does not since this dataset is particularly noisy in terms of punctuation and letter cases. Conversely, the restoration model also uses data from the How2 and Mozilla Common Voice datasets, which the translation model does not use as they do not contain German text. The translation model uses the IWSLT development set from 2010 and test sets from 2011–2015 as validation data, while the restoration model uses them as supplementary training data in order to reinforce domain bias, using only the MuST-C development set for validation.

Initially, we "clean" the output of our ASR model to remove segments containing musical note characters (♫♪), and repeating phrases that were consistently hallucinated during silence, applause, laughter or noise in the audio (e.g. in our case, "Shake. Fold."), as well as parts of segments that designate the speaker (e.g. "Audience: ..."). Subsequently, we use the same preprocessing pipeline for the cleaned ASR output as we do for all of our text data. For this, we start by removing non-printing characters, normalizing punctuation, and retokenizing the text using the corresponding utilities from the Moses toolkit (Koehn et al., 2007). Afterwards, we apply subword segmentation via SentencePiece (Kudo and Richardson, 2018), using a joint English–German BPE model with a vocabulary size of 32 000 for all of our translation models, and an English unigram model with a vocabulary size of 24 000 for the restoration stage of our cascade SLT, both trained on all of the data used for the translation and restoration models combined.



(a) opt1 (b) opt2 (c) opt2ShareEnc (d) opt3 (e) opt4 (f) opt5ShareEnc

Figure 1: Configurations tested for multitask training.

Before the training of the restoration model, the training data was run through a Moses truecaser model (trained on the same selection of training data as the restoration model) as an additional step before segmentation. This step removes sentence-initial capitalization for words that would not be capitalized otherwise, ensuring that differences in distributions of words appearing in sentence-initial positions does not influence case restoration for the model. Once truecased and segmented, we assign the processed data as the target for the restoration model, and continue to strip punctuation and lowercase the target to generate the source. This configuration comes with the useful side effect of the model learning to generate truecased output, which may be beneficial for MT.

4 Experiments

In this section we report on the experiments that lead up to our final submissions. The experiments on this section have been trained, validated and tested on the respective splits of the MuST-C.

As a first stage, we focused on selecting the multitask training strategy that performed better. Having the three modalities ENAUDIO, ENTEXT and DETEXT as possible inputs, and both text modalities as possible outputs, the number of combinations where audio is an input scales up to 64[2], without taking into account the cases where the text encoder is shared between German and English. We considered the 5 scenarios depicted in Figure 1 and present its results in Table 3 together with the number of steps it took for them to converge.

All the models were trained using the same set of hyperparameters. At the time we ran these experiments, the final version of the audio encoder was not ready for deployment, so we used a 4-layered pyramidal CNN+RNN encoder adaptation from Amodei et al. (2016) with 512 hidden units and pooling factors of (1,1,2,2) after each layer, respectively. For the text encoders, we applied embedding layers of 512 dimensions, four stacked bidirectional LSTM layers with 512 hidden units (256 per direction). We use attentive text decoders composed of two unidirectional LSTM layers with 512 units. Regarding the shared attention bridge layer, we used 100 *attention heads* with 1024 hidden units each. For multilingual models, we apply a uniform language-rotating scheduler. Training is performed using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0002 and batch size 32 for all source-target pairs, for at most 100,000 steps per language pair[3].

Our preliminary BLEU scores for these models are low. We, however, justify our choice to include them given the low performance of other experiments in similar scenarios reported in the literature. Namely, Tu et al. (2019) reported 9.55 BLEU training on the same set with a transformer based architecture, the only paper that trains and tests on the same set, and thus the only truly comparable results. In addition, Di Gangi et al. (2019a) reported 12.25 BLEU training MuST-C together with IWSLT18 and initialized their system with the ASR system.

Configuration	BLEU	Steps
opt3	5.00	330K
opt5shareEnc	4.94	250K
opt2shareEnc	4.84	250K
opt4	4.50	300K
opt1	4.30	220K
opt2	3.62	190K

Table 3: Training steps and best BLEU scores obtained with end-to-end systems on the German part fo the MuST-C test set.

The well-known sensitivity to hyperperparameter choice of the transformer architecture is also visible in our transformer-based audio encoders. We performed hyperparameter tuning on opt3 multitask training configuration (Figure 1 (d)). This resulted in a performance of a 9.53 BLEU score on German translations and 47.63 on the English,

[2] 64 is the total number of bipartite graphs that can be defined on sets of three and two vertices.

[3] Model configuration 3, for instance, has 4 language pairs was trained for at most 400K steps

System	status	de BLEU	en BLEU	WER	Steps
end-to-end opt6	submission time	12.90	56.65	36	172K
	converged	14.38	59.22	33	294K
end-to-end opt3	submission time	9.47	44.12	48	32K
	converged	11.71	52.91	40	72K
cascade bpe37k		22.20	60.87	29	-
cascade char-level		20.90	54.49	55	-

Table 4: Scores of our primary and contrastive submissions on on the MuST-C test set.

a clear increase from the untuned models that got at most 1 BLEU point in any of them. The final hyperparameter setup consists of:

- text encoders and decoders using 3 layered transformer architecture with 8 heads, 512 dimensional embeddings, 2048 feedforward hidden dimensions, and a batch size of 4096 tokens;
- audio encoders as described in Section 2 with 2 CNN layers with stride of 2 and kernel width of, the first of which takes a single input channel, three 8-headed transformer layers, positional embeddings of size 512 concatenated to the output of a linear layer for being passed to the transformer layers, a batch size of 32 utterances; and
- an attention bridge of size 100 with a hidden dimension of 1024.

Training was done with 8,000 warmup steps, using an Adam optimizer with learning rate 2 and Noam decay method, accumulation count of 8 to have an approximate effective batch size of 256 for the audio utterances, dropping utterances above the length of 5500, and a language rotating scheduler that uses the inverse of the batch size as weights [4].

We also tried other strategies such as (i) using 3, 4 and 6 stacked filterbanks as different channel inputs for the CNNs to reduce the input size instead of dropping utterances, (ii) using SpecAugment (Park et al., 2019) layers (2 frequency masks of width 20 and 2 time masks of width 50) to produce a data augmentation effect while training, (iii) including layer normalization after the attention bridge, (iv) using the positional embeddings of our transformer-based audio encoder in other places of the encoder or not using them at all. Unfortunately, none of them produced as effective improvements as what we describe above. We note that it is probable that using milder hyperparameters for SpecAugment could be beneficial.

5 Results

From the insights gained out of our experiments on the MuST-C dataset, for our submission, we train a system using the data as described in section 3 with the training configuration opt3 (see Figure 1 (d)) and the hyperparameters that yielded the best results. Further, we decided to try out an additional training configuration we had not previously tested: ENAUDIO as input and DETEXT and ENTEXT as output, which we refer to as opt6. The former outperformed the latter when tested on the MuST-C test set. One of our main aims in participating in this task is to test our multitask architecture; for this reason we submit our best SLT system as primary system and the cascade approach with subword segmentation as contrastive baseline. We would like to note that, unfortunately, at the time of submission, our end-to-end systems had not converged yet.

For the sake of consistency, these have been benchmarked with the MuST-C test set as well. The results are reported in Table 4.

6 Conclusion

In this paper we present our work for the IWSLT2020 offline speech translation task, along with the set of experiments that led to our final systems. Our submission includes both a cascaded baseline and a multimodal system trainable in a multitask fashion. Our work shows that it is possible to train a system that shares internal representations for transferring the implicit semantics between audio and text modalities. The nature of the architecture enables end-to-end SLT, while at

[4] In case of training opt3, the weights assigned to ENAUDIO → {DETEXT,ENTEXT} are 0.42 each and both text-to-text pairs get 0.08 because the average sentence length of MuST-C is around 24, which implies that 4096 tokens are about 170 sentences.

the same time providing a system capable of performing ASR and MT. Although this represents an important step in multimodal MT, there is still a lot of room for improvement in the proposed systems. In future work, we would like to implement more sophisticated audio encoders, such as the S-Transfomer. This, along with using the same amount of data during training, will allow us to draw a truly fair comparison between both end-to-end and cascade approaches.

Acknowledgments

This work is part of the FoTran project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 771113), as well as the MeMAD project, funded by the European Union's Horizon 2020 Research and Innovation Programme (grant № 780069).

References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. 2016. Deep Speech 2: End-to-end speech recognition in English and Mandarin. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 173–182, New York, New York, USA. PMLR.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2019. Common Voice: A massively-multilingual speech corpus.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. Accepted to ACL 2020, System Demonstrations.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, California, USA. Conference Track.

Ondřej Cífka and Ondřej Bojar. 2018. Are BLEU and meaning representation in opposition? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1362–1371, Melbourne, Australia. Association for Computational Linguistics.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019a. MuST-C: a multilingual speech translation corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, "Minneapolis, MN, USA".

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2019b. Adapting transformer to end-to-end spoken language translation. In *Proc. Interspeech 2019*, pages 1133–1137.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1723–1732, Beijing, China.

Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T Yarman Vural, and Yoshua Bengio. 2017. Multi-way, multilingual neural machine translation. *Computer Speech & Language*, 45:236–252.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. Europarl-ST: A multilingual corpus for speech translation of parliamentary debates. Accepted to ICASSP2020.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, Vancouver, Canada.

Philipp Koehn, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, Evan Herbst, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, and Christine Moran. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions - ACL '07*, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92, Belgium, Brussels. Association for Computational Linguistics.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016*, San Juan, Puerto Rico. Conference Track (Poster).

Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Singh Sachan, Philip Arthur, Pierre Godard, John Hewitt, Rachid Riad, and Liming Wang. 2018. XNMT: The extensible neural machine translation toolkit. In *Conference of the Association for Machine Translation in the Americas (AMTA) Open Source Software Showcase*, Boston.

J. Niehues, R. Cattoni, S. Stüker, M. Negri, M. Turchi, E. Salesky, R. Sanabria, L. Barrault, L. Specia, and M. Federico. 2019. The IWSLT 2019 evaluation campaign. In *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT 2019)*, Hong Kong, China.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Dogus Cubuk, and Quoc V. Le. 2019. SpecAugment: A simple augmentation method for automatic speech recognition. In *INTERSPEECH*.

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 and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *ACL workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In *6th International Conference on Learning Representations, ICLR 2018*, Vancouver, Canada. Conference Track (Poster).

Umut Sulubacak, Jörg Tiedemann, Aku Rouhe, Stig-Arne Grönroos, and Mikko Kurimo. 2018. The memad submission to the iwslt 2018 speech translation task. In *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT 2019)*, pages 89–94, Brussels, Belgium.

Mei Tu, Wei Liu, Lijie Wang, Xiao Chen, and Xue Wen. 2019. End-to-end speech translation system description of LIT for IWSLT 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, Long Beach, California, USA.

Raúl Vázquez, Alessandro Raganato, Mathias Creutz, and Jörg Tiedemann. 2020. A systematic study of inner-attention-based sentence representations in multilingual neural machine translation. *Computational Linguistics*, 0(ja):1–53.