

ONE-TO-MANY MULTILINGUAL END-TO-END SPEECH TRANSLATION

Mattia A. Di Gangi^{1,2}, Matteo Negri¹, Marco Turchi¹

¹Fondazione Bruno Kessler, MT unit, Italy

²University of Trento, DiSI, Italy

ABSTRACT

Nowadays, training end-to-end neural models for spoken language translation (SLT) still has to confront with extreme data scarcity conditions. The existing SLT parallel corpora are indeed orders of magnitude smaller than those available for the closely related tasks of automatic speech recognition (ASR) and machine translation (MT), which usually comprise tens of millions of instances. To cope with data paucity, in this paper we explore the effectiveness of transfer learning in end-to-end SLT by presenting a multilingual approach to the task. Multilingual solutions are widely studied in MT and usually rely on “*target forcing*”, in which multilingual parallel data are combined to train a single model by prepending to the input sequences a language token that specifies the target language. However, when tested in speech translation, our experiments show that MT-like *target forcing*, used as is, is not effective in discriminating among the target languages. Thus, we propose a variant that uses target-language embeddings to shift the input representations in different portions of the space according to the language, so to better support the production of output in the desired target language. Our experiments on end-to-end SLT from English into six languages show important improvements when translating into similar languages, especially when these are supported by scarce data. Further improvements are obtained when using English ASR data as an additional language (up to +2.5 BLEU points).

Index Terms—deep learning, speech translation, multilinguality, direct speech-to-text translation.

1. INTRODUCTION

The state-of-the-art results obtained by encoder-decoder models [1] with sequence-to-sequence learning in fields like ASR [2, 3, 4, 5] and, most importantly, machine translation [6, 7], have led to the recent proposal of sequence-to-sequence learning for direct speech-to-text translation [8, 9], that is translating from audio without an intermediate output representation. Unfortunately, end-to-end models require large amounts of parallel training data [10] that are not yet available for the SLT task. Indeed, while state-of-the-art sequence-to-sequence systems for ASR and MT are respectively trained on thousands

of hours of transcribed speech [11] and tens of millions of parallel sentences [12], the largest publicly available SLT corpus comprises about 500 hours of translated speech and few others amount to less than 300 hours each [13].

To overcome this limitation, several works have proposed approaches that exploit in different ways the wealth of ASR or MT data available (respectively, the audio-transcription and the parallel text corpora used to train the two types of systems). Multitask learning or transfer learning are generally used to exploit ASR data [14, 15, 16, 17] with positive results, and the improvements are more evident when less training data for SLT are available. In [18], the authors report good results on a small dataset by an approach that combines transfer learning from both ASR and MT using task-based curriculum learning. Other approaches to overcome the low-resource condition are data augmentation [19] and knowledge distillation [20]. However, [21] showed that current direct models are not data-efficient in leveraging non-SLT data, and that the classic “cascade” approach (i.e. a pipelined architecture integrating ASR and MT¹) still performs better.

In this paper, we take advantage of the recent release of MuST-C [13], which provides parallel SLT data for eight languages, in order to study whether multilingual data can be used to train systems with better translation quality than unidirectional (i.e. *one-to-one*) systems. Multilinguality has been widely explored in neural MT [22, 23, 24, 25, 26, 27], where it is now commonly performed using the *target forcing* mechanism [28, 29], which enables translation to many languages (*{one,many}-to-many*) without changing the underlying NMT architecture. The idea is to prepend the source sentence with a token representing the target language, and all the sentences are processed using the same shared encoder-decoder architecture. Although this approach has been proposed for RNN-based NMT, it works even better [30] when using the Transformer [6] architecture. Target forcing has also been applied to multilingual speech recognition [31, 32] showing to improve the transcription quality, although multilingual ASR shows to be better than its monolingual counterparts even when the language token is not provided.

¹Though effective, cascade SLT solutions have some drawbacks that end-to-end SLT aims to overcome in the long run. Besides the higher architectural complexity, these include: error propagation (ASR errors are hard to recover by the MT component), and larger inference latency.

A single model with shared parameters is particularly appealing in low-resource scenarios [33, 29] as it performs a sort of transfer learning across languages. However, compared to *one-to-one* models, the results of a multilingual model usually degrade in the language directions supported by more training data. Taking advantage of the MuST-C corpus, in this study we focus on the *one-to-many* scenario and investigate what groups of target languages favor transfer learning in SLT. To the best of our knowledge, this represents the first study on the effectiveness of the multilingual approach to SLT.

Along this direction, we proceed incrementally by first showing the limitations of MT-like target forcing and then by proposing and evaluating our SLT-oriented enhancements. First, our initial experiments show that the target forcing approach as proposed in [29] compares poorly with the unidirectional baselines. By looking at the output, we observe that the system produces sentences whose words are coherently in one language, but in many cases the chosen language is wrong. As the system is not able to learn the co-occurrence between the embedding of the language token and the words in the target language, we then propose to give a stronger learning signal by modifying the input content using the language embedding. This, in practice, is repeated along the time dimension so to be propagated through the whole input sequence (rather than being one single vector among thousands of others). Our experiments show that, by using this variant, translating into similar languages, i.e. Germanic (German and Dutch) and western European Romance (French, Italian, Spanish and Portuguese) leads to better average results than those obtained by unidirectional systems. However, and to our view unsurprisingly due to the difficulty to transfer knowledge across distant languages, the same improvements are not observed when merging more languages. Indeed, using the six languages together as a target yields improvements only for the lesser-resourced language direction, while combining all the eight languages covered by MuST-C leads to performance degradations in all the cases. In our final experiments, we also added English to the target languages of all the multilingual systems, which are then trained for translation and ASR. This provides a slight but consistent improvement to all results. Overall, in all but the two target languages with more training data, our best multilingual models outperform *one-to-one* models of comparable size by at least 0.4 BLEU points. In particular, on the least represented language direction in MuST-C (i.e. English→Portuguese, for which the corpus includes 385 hours of translated speech), the observed performance improvement over the *one-to-one* competitor (up to +2.5 BLEU points) indicates the feasibility of the proposed approach to operate in low-resource conditions.

2. DIRECT SPEECH TRANSLATION

Direct speech translation is defined as the problem of generating a sequence \mathbf{Y} representing a text in a target language,

given an input signal \mathbf{X} representing a speech in the source language. A model for this task can be trained by optimizing the likelihood $L = \log P(\mathbf{Y}|\mathbf{X}; \theta)$, where θ is the vector of model’s parameters. The loss is usually computed in an autoregressive manner such as $L = \sum_{t=0}^T \log P(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{X}; \theta)$. Recent works on end-to-end SLT used deep learning recurrent models based on LSTMs [34] that differ mainly in the number of LSTM layers and the use of convolution in the encoder [14, 16, 19]. However, recurrent models are characterized by slow training [35, 36] and they have been replaced in MT by Transformer [6]. In our experiments we used Speech-Transformer [37], an adaptation of Transformer to ASR, which in recent previous work has shown to perform really well with little modification also on monolingual end-to-end SLT [38].

Speech-Transformer adapts Transformer to work with audio input provided as sequences of MEL filterbanks [39], which are characterized by joint dependencies in the two dimensions of time and frequency [40] and that are orders of magnitude longer than the text representations handled by MT. Because of those characteristics, the input is first processed and reduced with 2 layers of strided 2D CNNs [41], each followed by ReLU activation and batch normalization [42]. 2D CNNs reduce the input dimension while capturing short-range bidimensional dependencies. The output of the second CNN layer is processed by two stacked 2D self-attention layers [37], which are meant to model long-range bidimensional dependencies, then it is reshaped and processed by a feed-forward layer with ReLU activation. Then, the output of ReLU is summed to position vectors obtained through a trigonometric positional encoding [6] processed by a stack of Transformer encoder layers (darker box on the left in Figure 1, which provides a simplified schema of the architecture). Each layer consists of a multi-head attention network for computing self attention of its input, followed by a 2-layered feed-forward network. Each of the two sublayers is followed by layer normalization [43] and residual connections [44]. The decoder consists of a stack of Transformer layers, similar to the encoder layers, except for the presence of an attention between encoder and decoder before the feed-forward network. The decoder receives in input a sequence of character embeddings summed with trigonometric positional encoding (right-hand side of the figure). **2D Self-attention** (2DSAN) starts with three parallel 2D CNN layers that compute three different representation of their input $\mathbf{Q}, \mathbf{K}, \mathbf{V}$. Attention is computed as:

$$d_i = \text{softmax}(\mathbf{Q}_i \mathbf{K}_i^T / \sqrt{d_{\text{model}}}) \cdot \mathbf{V}_i \quad (1)$$

with $i = \{1, \dots, c\}$, where c is the number of CNN filters, and the attention is computed in parallel for each i . Equation 1 shows how attention along the time dimension is computed, but the three tensors are also transposed to compute attention along the frequency domain. All the attention outputs of the two parallel attention layers are then concatenated

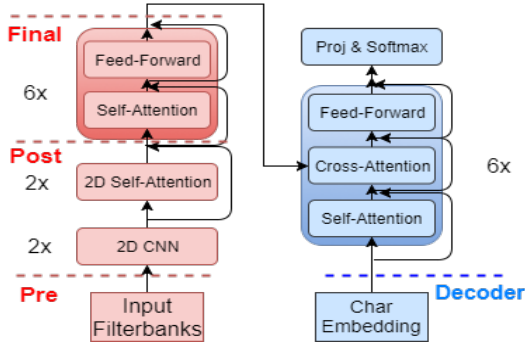


Fig. 1. Proposed encoder-decoder architecture. The dashed lines represent the points where *target forcing* can be applied.

and processed by a final 2D CNN layer. All convolutions are followed by ReLU activation and batch normalization. This mechanism is analogous to the multi-head attention, except for the use of CNNs instead of affine transformations, and the use of the additional attention along the frequency domain.

The authors of [37] mention the introduction of a distance penalty mechanism in their encoder self-attention layers, but they do not provide additional details. Building on the same idea, we use a distance penalty that is subtracted to the input of softmax in Equation 1. Let $d = |i - j|$ be the distance between positions i and j , we then define a logarithmic penalty [38] as follows:

$$\pi_{\log}(d) = \begin{cases} 0, & \text{if } d = 0 \\ \log_e(d), & \text{else} \end{cases} \quad (2)$$

The use of logarithm is motivated by the fact that we want to bias the self attention towards the local context of each position, but we do not want to entirely prevent it from capturing the influence of global context. In preliminary experiments, we observed that using this logarithmic distance penalty mechanism leads to better results than an unbiased attention.

3. MULTILINGUAL TRANSLATION

In this section we first introduce the target forcing mechanism used in multilingual NMT and propose two variants to apply it to the SLT task. Then, we discuss two different but complementary ways to exploit ASR data in order to improve final translation performance.

3.1. Target forcing

We perform one-to-many multilingual SLT with a single model by using the *target forcing* mechanism [29], which tags every source sentence with a language token indicating the target language. In NMT, the language token is simply prepended to the input text sequence, and used identically to

all the other tokens to retrieve an embedding that is jointly learned with the rest of the network. However, this method has to be adapted to a speech encoder that does not have an embedding layer. Thus, we propose two different approaches, namely: *i) concat*, which prepends a language embedding to the input sequence, and *ii) merge*, which sums the embedding to all the elements in the sequence. In both cases, the language embeddings are learnable parameters of the network.

Concat. The first approach is a straightforward adaptation of the *target forcing* mechanism [29] to speech translation. Let $\mathbf{X} \in R^{T \times F}$ be a matrix containing a sequence of audio feature vectors, where T is the number of time steps and F is the number of features, and let $\mathbf{l} \in R^{1 \times F}$ be a language embedding of the same size F . Then, analogously to the text input case, where the language token is prepended to the input string, we produce the new target-language dependent representation $\mathbf{X}' \in R^{(T+1) \times F}$ by concatenating \mathbf{l} to \mathbf{X} . In this case, only the language embedding is a learnable parameter, while \mathbf{X} is a fixed sequence of MEL filterbanks.

Merge. While the *concat* method modifies the input representation by concatenating an additional vector, the *merge* approach alters the content of the input representation. Given the two tensors \mathbf{X} and \mathbf{l} as in the previous case, now we define $\mathbf{X}' \in R^{T \times F}$ as the sum $\mathbf{X}' = \mathbf{X} + \mathbf{l}$ where \mathbf{l} is repeated along the time dimension. Considering the length of the audio input (recall that, differently from the textual representations handled by MT, the sequences of MEL filterbanks input to SLT are orders of magnitude longer), the intuition is that the *concat* approach can fail in propagating the language token towards the whole input sequence. In contrast, with the *merge* approach the input tensor is translated (in a geometric sense) to different portions of the space for each different language so to have representations that are clearly distinct between one language and another starting by the input. In this way, the same input sentence has clearly different representations when it has to be translated to different languages, and the learning task becomes easier.

3.2. ASR data

It is widely demonstrated in literature that exploiting ASR data is useful for improving the performance of direct SLT systems, with the main approaches being transfer learning [16, 15] and multitask learning [14, 17]. Indeed, the ASR task is easier than translation as it is monolingual and does not involve reordering and, as such, it is used to obtain encoder representations that are better also for SLT. In our experiments, we exploit ASR data in two ways. The first way is transfer learning, and consists in training another Speech Transformer for the ASR task, then using its encoder weights to initialize the parameters of the SLT model (pre-training). The second way is to use English ASR data as if it was an additional language for the multilingual system (+ASR). This approach is analogous to multi-task training but, unlike the

Tgt	#Hours	#Sent	#Talk	src w	tgt w
De	408	234K	2,093	4.3M	4.0M
Es	504	270K	2,564	5.3M	5.1M
Fr	492	280K	2,510	5.2M	5.4M
It	465	258K	2,374	4.9M	4.6M
Nl	442	253K	2,267	4.7M	4.3M
Pt	385	211K	2,050	4.0M	3.8M
Ro	432	240K	2,216	4.6M	4.3M
Ru	489	270K	2,498	5.1M	4.3M

Table 1. Statistics for each section of MuST-C.

approach used in [14] of using two different decoders, we can take advantage of the multilingual model and use ASR data to train also the decoder [21]. Because of its documented effectiveness, we use pre-training in all our experiments, while we evaluate the effectiveness of the +ASR approach with ablation experiments.

4. EXPERIMENTS

Dataset. We use the MuST-C corpus [13], which represents the largest publicly available multilingual corpus (one-to-many) for SLT. It covers eight language directions, from English to German, Spanish, French, Italian, Dutch, Portuguese, Romanian and Russian. The corpus consists of audio, transcriptions and translations of English TED talks, and it comes with a predefined training, validation and test split. In terms of hours of transcribed/translated speech (see Table 1 for complete statistics), the size of the different sections of the corpus ranges from 385 (Portuguese) to 504 hours (Spanish). **Model settings.** The first two CNNs in the encoder have 16 output channels, 3×3 kernel and stride (2, 2). The CNNs inside the 2D self-attention have 3×3 kernels, 4 output channels and stride 1. The output CNN of the 2D self-attention has 16 output channels. The following feed-forward layer has 512 output features, which is the same size as the Transformer layers. The hidden feed-forward layer size of Transformer is 1024. The decoder layers have also size 512 and hidden size 1024. Dropout is set to 0.1 in each layer. Each minibatch includes up to 8 sentences for each language and we update the gradient every 16 iterations. All the models are trained with the Adam [45] optimizer with an initial learning rate of 0.0003, then 4000 warmup steps during which it increases linearly up to a max value, and then decreases with the inverse square root of the number of steps [6]. As the batch size depends on the number of languages, the maximum learning rate is increased with this number. We searched the best learning rate testing on a held-out set, and we selected 0.01 for the experiments with the Germanic and Romance languages, and 0.02 for the experiments with 6 or all languages. Our baselines are based on the same architecture and replicate the results obtained in [38], which are reportedly stronger than the ones in [13]. As additional stronger baselines, we also

	De	Nl	Es	Fr	It	Pt
Baseline	17.3	18.8	20.8	26.9	16.8	20.1
C-Pre	14.0	11.6	13.0	16.3	10.7	14.5
C-Post	12.0	13.8	12.3	18.0	9.3	14.6
C-Final	14.5	12.1	13.6	16.7	10.2	16.2
M-Pre	17.6	19.5	20.5	26.2	17.2	22.3
M-Post	17.1	19.2	20.5	26.2	17.4	22.3
M-Final	17.4	18.8	20.4	26.7	17.2	22.2
M-Dec.	17.3	19.1	20.6	26.2	17.2	22.0
M-Pre + ASR	17.7	20.0	20.9	26.5	18.0	22.6

Table 2. BLEU score results with *concat* (C-*) and *merge* (M-*) *target forcing* on 6 languages. The baselines are *one-to-one* systems. The other results are computed with one multilingual system for En→De,NL and one for En→Es,Fr,It,Pt.

train cascade models that concatenate an ASR and an MT system, where the ASR consists of an attention model using the same architecture of our direct SLT systems and the MT is a Transformer Base architecture [6]. All the models are implemented in Pytorch [46] with FBK-fairseq-ST², derived from the fairseq toolkit [35].

Experimental Settings. In our first experiments we train two multilingual models, one for Germanic languages (German, Dutch), and one for western European Romance languages (Spanish, French, Italian, Portuguese). Although Romanian is also a Romance language, we keep it out from this experiment because its slavic influences make it quite different from the other 4 languages. Romanian and Russian are finally used in one last experiment.

Data preprocessing. From each audio segment we compute the MEL filter-banks [39] with 40 filters, using overlapping windows of 25 ms and step of 10 ms. The resulting spectrograms are normalized by subtracting the mean and dividing by the standard deviation. All the texts are tokenized and the punctuation is normalized. In our cascade models, ASR systems are trained without punctuation in output and with lowercased text split into characters, while MT systems receive lowercased English text without punctuation in input, and produce text that preserves the casing and with punctuation in the target language as output. Both source and target are split in subwords with BPE segmentation [47] using 32,000 joint merge rules in MT. In the target side of the SLT systems, texts are split into characters and punctuation is kept in the target texts. We do not use BPE for SLT because in our preliminary experiments it performed poorly.

5. RESULTS

Concat vs Merge. Our first experiment consists in comparing the baselines with the multilingual models based on the *target forcing* mechanism. The results presented in Table 2 show that *concat* target forcing (C-Pre) is much worse than the

²<https://github.com/mattiadg/FBK-Fairseq-ST>

baselines. However, note that our baselines are stronger than the ones reported in [13]. By looking at the translations, we found that the cause of the degradation is that many sentences are acceptable translations, but in a wrong language. We first hypothesize that the processing performed in the layer preceding the encoder self-attentional stack loses the language information. Thus, we concatenate the language embedding to the representations of the *Post* and *Final* positions (see Fig. 1), both after the 2DSANs. The new results, listed in Table 2, show small and non consistent variations, and are still worse than the baselines in all languages. Our second hypothesis is that the networks are not able to learn the joint probabilities of language embeddings and character sequences because of a combination of factors: character-level translations (instead of sub-words), very long source-side sequences and the source sides in the corpus are highly overlapping between languages. Thus, we assume that our networks can learn to discriminate better among target languages by giving a stronger language signal. For this reason, we introduce the *merge* target forcing that forces the network to generate target-language dependent encoder representations by translating them in different portions of the space according to the target language. The results in Table 2 show that *merge* target forcing (M-*) is definitely better than *concat* for all the target languages and also obtains performance that is on par with, or better than, the baselines. M-Pre is the system that shows high results more consistently in all languages, and the largest improvement is observed in En-Pt, with over 2.0 BLEU points of gain, followed by +0.7 in En-Nl. The BLEU score slightly degrades for Spanish by 0.2 ~ 0.4, and for French by 0.2 ~ 0.7. Besides the three different language embedding positions in the encoder, we also performed experiments by applying target forcing on the decoder, but they show slightly worse performance. Then, for the following experiments we will continue only with the *Pre* position that results in the best average performance on all the language directions.

When adding also ASR data to our training set (M-Pre + ASR) we observe small but consistent improvements in all languages. In this case, there are improvements over the baselines larger than 1 BLEU point in 3 out of 6 target languages: Dutch, Italian and Portuguese. Moreover, the system does not degrade in En-Es, the direction with the largest dataset available, and it is only 0.4 BLEU points below the baseline in En-Fr. These last experiments show that the advantage of training ASR data are visible even when the SLT models have been pre-trained on the ASR task.

Number of languages. When training a system with all the 6 target languages (De, Nl, Es, Fr, It, Pt) together, in Table 3 we observe results on par with the baseline for German and Italian, slightly worse results on Dutch and Spanish (respectively, -0.4 and -0.8), a larger degradation for French (-1.5), but also a large improvement on Portuguese (+1.7), although smaller than using 4 languages plus ASR. When adding ASR data to the 6 languages, we observe improvements in most languages,

	De	Nl	Es	Fr	It	Pt	Ro	Ru
Baseline								
	17.3	18.8	20.8	26.9	16.8	20.1	16.5	10.5
Multilingual								
6	17.3	18.4	20.0	25.4	16.9	21.8	-	-
8	16.5	17.8	18.9	24.5	16.2	20.8	15.9	9.8
+ ASR								
6	17.4	19.2	19.7	26.0	17.2	21.8	-	-
8	15.9	17.2	18.3	23.7	15.1	19.9	15.5	9.7

Table 3. BLEU score results for multilingual direct SLT systems with 6 and 8 target languages.

	De	Nl	Es	Fr	It	Pt
Baseline	17.3	18.8	20.8	26.9	16.8	20.1
M-Pre + ASR	17.7	20.0	20.9	26.5	18.0	22.6
BL-Cascade	18.5	22.2	22.5	27.9	18.9	21.5
M-Cascade	18.6	22.0	22.1	27.3	18.5	22.8
MT	25.3	30.3	29.9	35.5	25.8	31.1

Table 4. Comparison (BLEU scores) of the Baseline and the best multilingual system with the single language cascade (BL-Cascade) and the multilingual cascade (M-Cascade).

and the new system is worse than the baseline only for Spanish and French, although the gap for French has been reduced to -0.9. However, using all the 8 target languages leads to worse results and adding ASR data contributes to worsen the performance in this case. We think that the reason is related to the relatively low number of parameters of our models (~ 33 millions), which reduce their capability to learn and discriminate between a larger number of languages.

Comparison with cascade. In Table 4, we compare the single and our best multilingual systems with two different cascade models. One is BL-Cascade, in which the ASR system is concatenated with MT systems individually trained for each target language. The other is M-Cascade, where ASR is concatenated with a multilingual NMT model trained on all the 6 language pairs. As expected, our direct SLT baselines are significantly worse than the BL-cascade systems with differences that range from -1.0 for French to -3.4 for Dutch. Comparing the BL-Cascade with the M-Cascade systems, we do not observe significant variations for the Germanic languages, but lower results in 3 out of 4 Romance languages (-0.6 for French), with the single improvement of +1.3 for Portuguese. Thus, multilingual MT generally affects negatively the performance, being beneficial only for the lesser resourced languages.

A comparison between our best multilingual SLT (M-Pre + ASR) and the cascade systems shows that our system is able to reduce the gap between them. On Portuguese, it is on par with M-Cascade and 1.1 BLEU points above BL-Cascade. For Dutch, the target language with the largest initial gap, the difference is reduced to 2 BLEU points, and for Italian it is only 0.9 lower than the best cascade system (BL-Cascade).

	De	Nl	Es	Fr	It	Pt
M-Pre	95.7	98.5	97.2	94.6	95.3	96.6
M-Pre + ASR	96.1	98.7	97.9	95.3	95.4	95.2

Table 5. Percentage of sentences translated in the correct language.

These results show that the multilingual SLT system is a valid solution to enhance the performance of the end-to-end SLT system and is able to reduce the gap with cascade architectures. The last row of table 4 shows the results of the MT components of the cascade, while the ASR WER is about 25% for all languages.

Language analysis. To better understand the behaviour of the multilingual system, we used langdetect³ to automatically detect the language of each translated sentence. The results listed in Table 5, show that our M-Pre systems do not translate in the correct language in all the cases, although the percentage is higher than 95%. When using also ASR data, the percentage of outputs in the correct language increases slightly in all languages except for Portuguese. However, these improvements do not correlate with the performance variations measured in terms of BLEU score. This suggests that BLEU gains observed with M-Pre + ASR come from better translations rather than a higher number of sentences translated in the correct target language.

6. DISCUSSION

The classic *target forcing* mechanism, which prepends the language embedding in the input sequence and is widely used in MT, resulted to be not effective for the SLT task. Our variation, which sums the language embedding to the entire input representation, shows to be effective by imposing a sharper distinction between encoder representations for different target languages. This allows us to reduce the gap in performance with respect to a stronger cascade model. The actual reasons of this behaviour will be further investigated in future work, but here it is important to remark the differences with multilingual works in MT and ASR to understand the complexity of the SLT scenario. In MT, translations are performed at BPE-level, which results in a shorter input sequence than character-level, but also limits the vocabulary selection for each target language. Moreover, character-level representations mean that most of the vocabulary is shared between the languages, which can be a source of confusion. Multilingual ASR shares with our task the mapping from audio to text, with long sequences in the source side. However, the language is the same in the source and target sides and, as such, it is more difficult for the model to confuse the target language. Toshniwal et al. [31] have shown that a Transformer model trained on 9 Indian languages with different scripts

³<https://github.com/shuyo/language-detection>

outperforms monolingual baselines without any training trick, and further improvements are obtained with the *target forcing* mechanism. This case is significantly different from our task where, without target forcing, the system has a probability of $1/N$ to translate into the correct language, with N being the number of languages. Although these differences make multilingual SLT a challenging scenario, the results achieved by our systems indicate that the lack of data can be overcome by training a single multilingual model on more languages.

Our experiments have also shown, consistently with the results in MT [29], that multilingual translation is particularly beneficial for the less resourced language pairs. However, in order to obtain the best results, it is needed to keep the number of target languages limited and train on similar target languages. We have also found beneficial to use ASR data together with the other translation data, and we are not aware of works showing similar results on an MT task. The reason of the significant performance drops when the number of languages increases is probably related to the model capacity. Indeed, our models have ~ 30 million parameters, while large MT multilingual models can have one order of magnitude more parameters. Unfortunately, due to the large GPU memory occupation of SLT models, in particular with Transformer, we could not perform experiments with larger models.

7. CONCLUSIONS

We explored one-to-many multilingual speech translation as a method to increase the training data size for direct speech translation. Since in our experimental conditions MT-like target forcing often results in translation into a wrong language, we proposed a variation that overcomes the problem. Our results show that this approach can produce important improvements in the target languages with less data. Adding ASR data to the training set allows the multilingual SLT system to outperform baseline unidirectional (i.e. one-to-one) systems in 3 out of 6 target languages (the lesser-resourced ones) with an improvement larger than 1 BLEU score point, and to achieve comparable performance in the languages with more data. When adding all the MuST-C target languages in a single system, the performance degrades showing the difficulties of the multilingual SLT model to manage an increasing number of unrelated target languages. In future work, we plan to extend our method to many-to-many multilingual SLT, and to investigate strategies that take into account the peculiarities of this task, as well as evaluating the impact of model capacity to manage more languages.

Acknowledgements

This work is part of a project financially supported by an Amazon AWS ML Grant.

8. REFERENCES

- [1] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, “Sequence to Sequence Learning with Neural Networks,” in *Proc. of NIPS*, 2014.
- [2] Dario Amodei et al., “Deep Speech 2: End-to-end Speech Recognition in English and Mandarin,” in *Proc. of ICML*, 2016.
- [3] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, “Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition,” in *Proc. of ICASSP*, 2016.
- [4] Chung-Cheng Chiu et al., “State-of-the-art Speech Recognition with Sequence-to-sequence Models,” in *Proc. of ICASSP*, 2018.
- [5] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney, “Improved Training of End-to-end Attention Models for Speech Recognition,” in *Proc. of Interspeech*, 2018.
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is All You Need,” in *Proc. of NIPS 2017*, 2017.
- [7] Ondrej Bojar et al., “Findings of the 2018 Conference on Machine Translation (WMT18),” in *Proc. of WMT 2018*, 2018, pp. 272–307.
- [8] Alexandre Bérard, Olivier Pietquin, Laurent Besacier, and Christophe Servan, “Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation,” in *Proc. of NIPS Workshop on end-to-end learning for speech and audio processing*, 2016.
- [9] Sameer Bansal, Herman Kamper, Adam Lopez, and Sharon Goldwater, “Towards Speech-to-text Translation without Speech Recognition,” in *Proc. of EACL*, 2017.
- [10] Philipp Koehn and Rebecca Knowles, “Six Challenges for Neural Machine Translation,” in *Proc. of Workshop on Neural Machine Translation*, 2017.
- [11] Chiu Chung-Cheng et al., “State-of-the-art Speech Recognition With Sequence-to-Sequence Models,” in *Proc. of ICASSP*, 2018.
- [12] Hany Hassan et al., “Achieving Human Parity on Automatic Chinese to English News Translation,” *ArXiv e-prints arXiv:1803.05567v2*, 2018.
- [13] Mattia Antonino Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi, “MuST-C: a Multilingual Speech Translation Corpus,” in *Proc. of NAACL*, 2019, pp. 2012–2017.
- [14] Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen, “Sequence-to-Sequence Models Can Directly Translate Foreign Speech,” in *Proc. of Interspeech 2017*, Stockholm, Sweden, 2017.
- [15] Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater, “Pre-training on High-resource Speech Recognition Improves Low-resource Speech-to-text Translation,” *Proc. of NAACL*, 2019.
- [16] Alexandre Bérard, Laurent Besacier, Ali Can Kobayikoglu, and Olivier Pietquin, “End-to-End Automatic Speech Translation of Audiobooks,” in *Proc. of ICASSP 2018*, Calgary, Alberta, Canada, April 2018.
- [17] Antonios Anastasopoulos and David Chiang, “Tied Multitask Learning for Neural Speech Translation,” in *Proc. of NAACL*, 2018.
- [18] Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura, “Structured-based curriculum learning for end-to-end english-japanese speech translation,” *Proc. Interspeech 2017*, pp. 2630–2634, 2017.
- [19] Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron-J. Weiss, Yuan Cao, Chung-Cheng Chiu, Stella-Laurenzo Ari, and Yonghui Wu, “Leveraging Weakly Supervised Data to Improve End-to-End Speech-to-Text Translation,” *ArXiv e-prints arXiv:1811.02050*, 2018.
- [20] Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing Zong, “End-to-End Speech Translation with Knowledge Distillation,” *arXiv preprint arXiv:1904.08075*, 2019.
- [21] 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.
- [22] Barret Zoph and Kevin Knight, “Multi-Source Neural Translation,” in *Proc. of NAACL*, 2016, pp. 30–34.
- [23] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang, “Multi-task Learning for Multiple Language Translation,” in *Proc. 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)*, 2015, vol. 1, pp. 1723–1732.
- [24] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser, “Multi-task Sequence to Sequence Learning,” *Proc. of ICLR*, 2016.

- [25] Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T Yarman Vural, and Kyunghyun Cho, “Zero-Resource Translation with Multi-Lingual Neural Machine Translation,” in *Proc. of EMNLP*, 2016.
- [26] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio, “Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism,” in *Proc. of NAACL*, 2016, pp. 866–875.
- [27] Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun, “A Neural Interlingua for Multilingual Machine Translation,” in *Proc. of the Third Conference on Machine Translation: Research Papers*, 2018, pp. 84–92.
- [28] Thanh-Le Ha, Jan Niehues, and Alex Waibel, “Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder,” in *Proc. of the 13th International Workshop on Spoken Language Translation (IWSLT)*, 2016.
- [29] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al., “Googles multilingual neural machine translation system: Enabling zero-shot translation,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.
- [30] Surafel M Lakew, Mauro Cettolo, and Marcello Federico, “A Comparison of Transformer and Recurrent Neural Networks on Multilingual Neural Machine Translation,” *arXiv preprint arXiv:1806.06957*, 2018.
- [31] Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao, “Multilingual Speech Recognition With a Single End-to-End Model,” in *Proc. of ICASSP*, 2018.
- [32] Shiyu Zhou, Shuang Xu, and Bo Xu, “Multilingual End-to-End Speech Recognition with a Single Transformer on Low-Resource Languages,” *arXiv preprint arXiv:1806.05059*, 2018.
- [33] Surafel M Lakew, Mattia A Di Gangi, and Marcello Federico, “Multilingual Neural Machine Translation for Low Resource Languages,” in *Proc. of CLiC-it*, 2017.
- [34] Sepp Hochreiter and Jürgen Schmidhuber, “Long Short-Term Memory,” *Neural computation*, vol. 9, no. 8, 1997.
- [35] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin, “Convolutional Sequence to Sequence Learning,” in *Proc. of ICML*, August 2017.
- [36] Mattia Antonino Di Gangi and Marcello Federico, “Deep Neural Machine Translation with Weakly-Recurrent Units,” in *Proc. of EAMT*, 2018.
- [37] Linhao Dong, Shuang Xu, and Bo Xu, “Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition,” in *Proc. of ICASSP*, 2018.
- [38] Mattia Antonino Di Gangi, Matteo Negri, and Marco Turchi, “Adapting Transformer to End-to-end Spoken Language Translation,” in *INTERSPEECH*, 2019.
- [39] Steven Davis and Paul Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.
- [40] Jinyu Li, Abdelrahman Mohamed, Geoffrey Zweig, and Yifan Gong, “Exploring Multidimensional LSTMs for Large Vocabulary ASR,” in *Proc. of ICASSP*, 2016.
- [41] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-based Learning Applied to Document Recognition,” *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [42] Sergey Ioffe and Christian Szegedy, “Batch normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proc. of ICML*, 2015, pp. 448–456.
- [43] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep Residual Learning for Image Recognition,” in *Proc. of CVPR*, 2016, pp. 770–778.
- [45] Diederik Kingma and Jimmy Ba, “Adam: A Method for Stochastic Optimization,” in *Proc. of ICLR*, 2015.
- [46] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, “Automatic differentiation in PyTorch,” in *NIPS 2017 Workshop Autodiff*.
- [47] Rico Sennrich, Barry Haddow, and Alexandra Birch, “Neural Machine Translation of Rare Words with Subword Units,” in *Proc. of ACL*, 2016, pp. 1715–1725.