PRE-TRAINING ON HIGH-RESOURCE SPEECH RECOGNITION IMPROVES LOW-RESOURCE SPEECH-TO-TEXT TRANSLATION

Sameer Bansal[1]*, Herman Kamper*[2]*, Karen Livescu*[3]*, Adam Lopez*[1]*, Sharon Goldwater*[1]

[1]School of Informatics, University of Edinburgh, UK
[2]E&E Engineering, Stellenbosch University, South Africa
[3]Toyota Technological Institute at Chicago, USA

{sameer.bansal, sgwater, alopez}@inf.ed.ac.uk, kamperh@sun.ac.za, klivescu@ttic.edu

ABSTRACT

We present a simple approach to improve direct speech-to-text translation (ST) when the source language is low-resource: we pre-train the model on a high-resource automatic speech recognition (ASR) task, and then fine-tune its parameters for ST. We demonstrate that our approach is effective by pre-training on 300 hours of English ASR data to improve Spanish-English ST from 10.8 to 20.2 BLEU when only 20 hours of Spanish-English ST training data is available. Through an ablation study, we find that the pre-trained encoder (acoustic model) accounts for most of the improvement, which is surprising since the shared language in these tasks is the target language (text), and not the source language (audio). Applying this insight, we show that pre-training on ASR helps ST even when the ASR language differs from both source and target ST languages: pre-training on French ASR also improves Spanish-English ST. Finally, we show that the approach improves a true low-resource task: pre-training on a combination of English ASR and French ASR improves Mboshi-French ST, where only 4 hours of data are available, from 3.5 to 7.1 BLEU.

Index Terms— low-resource speech processing, speech translation, transfer learning, sequence-to-sequence model

1. INTRODUCTION

Speech-to-text Translation (ST) has many potential applications for low-resource languages: for example in language documentation, where the source language is often unwritten or endangered [1–5]; or in crisis relief, where emergency workers might need to respond to calls or requests in a foreign language [6]. Traditional ST is a pipeline of automatic speech recognition (ASR) and machine translation (MT), and thus requires requires transcribed source audio to train ASR and parallel text to train MT. These resources are typically unavailable for low-resource languages, but for our potential applications, there may be some source language audio paired with target language text translations. In these scenarios, end-to-end ST is appealing.

Recently, Weiss et al. [7] showed that end-to-end ST can be very effective, achieving an impressive BLEU score of 47.3 on Spanish-English ST. But this result required over 150 hours of translated audio for training, still a substantial resource requirement. By comparison, a similar system trained on only 20 hours of data for the same task achieved a BLEU score of 5.3 [8]. Other low-resource systems have similarly low accuracies [9, 10].

To improve end-to-end ST in low-resource settings, we can try to leverage other data resources. For example, if we have transcribed audio in the source language, we can use multi-task learning to improve ST [7, 9, 10]. But source language transcriptions are unlikely to be available in our scenarios of interest. Could we improve low-resource ST by leveraging resources from a high-resource language? For ASR, training a single model on multiple languages can be effective for all of them [11, 12]. For MT, *transfer learning* [13] has been very effective: pre-training a model for a high-resource language pair and transferring its parameters to a low-resource language pair when the target language is shared [14, 15]. Inspired by these successes, we show that low-resource ST can leverage transcribed audio in a high-resource target language, or even a different language altogether, simply by pre-training a model for the high-resource ASR task, and then transferring and fine-tuning some or all of the model's parameters for low-resource ST.

We first test our approach using Spanish as the source language and English as the target. After training an ASR system on 300 hours of English, fine-tuning on 20 hours of Spanish-English yields a BLEU score of 20.2, compared to only 10.8 for an ST model without ASR pre-training. Analyzing this result, we discover that the main benefit of pre-training arises from the transfer of the *encoder* parameters, which model the input acoustic signal. In fact, this effect is so strong that we also obtain improvements by pre-training on a language that differs from either the source or target: pre-training on French and then fine-tuning on Spanish-English. We hypothesize that pre-training the encoder parameters, even on a different language, allows the model to better normalize over acoustic variability (such as speaker and channel differences), and conclude that this variability, rather than translation itself, is one of the main difficulties in low-resource ST. A final set of experiments confirm that ASR pre-training also helps on another language pair where the input is truly low-resource: Mboshi-French.

2. METHOD

For both ASR and ST, we use an encoder-decoder model with attention adapted from [7, 8, 10], as shown in Figure 1. We use the same set of hyper-parameters for all our models, allowing us to conveniently transfer parameters between them. We also constrain the hyper-parameter search to fit a model into a single Titan X GPU, allowing us to maximize available compute resources.

We use a pre-trained English ASR model to initialize training of Spanish-English ST models, and a pre-trained French ASR model to initialize training of Mboshi-French ST models. In these configurations, the decoder shares the same vocabulary across the ASR and ST tasks. This is practical for settings where the target text language is high-resource with ASR data available.

In settings where both ST languages are low-resource, ASR data may only be available in a third language. To test whether

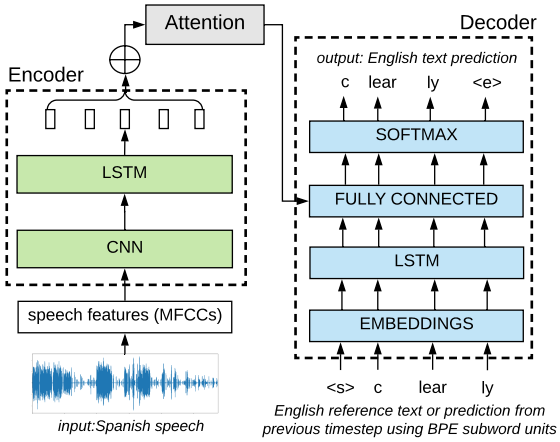


Fig. 1. Encoder-decoder with attention model architecture for both ASR and ST. Encoder input is the Spanish speech utterance *claro*, translated as *clearly*, preprocessed as BPE (subword) units.

transfer learning will help in this setting, we use a pre-trained French ASR model to train Spanish-English ST models; and English ASR for Mboshi-French models. In these cases, the ST languages are different from the ASR language, so we can only transfer the encoder parameters of the ASR model, since the dimensions of the decoder's output softmax layer are indexed by the vocabulary, which is not shared.[1] Sharing only the speech encoder parameters is much easier, since the speech input can be preprocessed in the same manner for all languages. This form of transfer learning is more flexible, as there are no constraints on the ASR language used.

3. EXPERIMENTAL SETUP

3.1. Data sets

English ASR. We use the Switchboard Telephone speech corpus [16], which consists of around 300 hours of English speech and transcripts, split into 260k utterances. The development set consists of 5 hours that we removed from the training set, split into 4k utterances.

French ASR. We use the French speech corpus from the GlobalPhone collection [17], which consists of around 20 hours of high quality read speech and transcripts, split into 9k utterances. The development set consists of 2 hours, split into 800 utterances.

Spanish-English ST. We use the Fisher Spanish speech corpus [18], which consists of 160 hours of telephone speech in a variety of Spanish dialects, split into 140K utterances. To simulate low-resource conditions, we construct smaller training corpora consisting of 50, 20, 10, 5, or 2.5 hours of data, selected at random from the full training data. The development and test sets each consist of around 4.5 hours of speech, split into 4K utterances. We do not use the corresponding Spanish transcripts; our target text consists of English translations that were collected through crowdsourcing [19, 20].

Mboshi-French ST. Mboshi is a Bantu language spoken in the Republic of Congo, with around 160,000 speakers.[2] We use the Mboshi-French parallel corpus [21], which consists of around 4 hours of Mboshi speech, split into a training set of 5K utterances and a development set of 500 utterances. Since this corpus does not include a designated test set, we randomly sampled and removed 200 utterances from training to use as a development set, and use the designated development data as a test set.

3.2. Preprocessing

Speech. We convert raw speech input to 13-dimensional MFCCs using Kaldi [22].[3] We also perform speaker-level mean and variance normalization. We do not stack the 13-dimensional features with delta and delta-deltas.

Text. The target text of the Spanish-English dataset contains 1.5M word tokens and 17K word types. If we model text as sequences of words, our model cannot produce any of the unseen word types in the test data and is penalized for this, but it can be trained very quickly [8]. If we instead model text as sequences of characters as in [7], we would have 7M tokens and 100 types, resulting in a model that is open-vocabulary, but very slow to train [8]. As an effective middle ground, we use byte pair encoding (BPE) [23] to segment each word into subwords, each of which is a character or a high-frequency sequence of characters—we use 1000 of these high-frequency sequences. Since the set of subwords includes the full set of characters, the model is still open vocabulary; but it results in a text with only 1.9M tokens and just over 1K types, which can be trained almost as fast as the word-level model.

The vocabulary for BPE depends on the frequency of character sequences, so it must be computed with respect to a specific corpus. For English, we use the Spanish-English ST target training text; and for French, the Mboshi-French ST target training text. Using only one vocabulary per target language simplifies our experimental setup, but is slightly unnatural since the Spanish-English vocabulary depends on the full 160-hour training set. However we have not observed substantial differences in experimental results due to this.

3.3. Model architecture for ASR and ST

Speech encoder. MFCC feature vectors are fed into a stack of two CNN layers, with 128 and 512 filters, a filter width of 9 frames each, and rectified linear unit (ReLU) activations [24]. We stride with a factor of 2 along time, in each of the CNN layers, reducing the sequence length by a factor of 4, which is important for reducing computation in the subsequent RNN layers. We apply batch normalization [25] after each CNN layer.

The output of the CNN layers is fed into a three-layer bidirectional long short-term memory (LSTM) [26]; each hidden layer has 256 dimensions.

Text decoder. At each time step, the decoder chooses the most probable token from the output of a softmax layer produced by a fully-connected layer that receives the current state of a recurrent layer computed from previous time steps, and an attention vector computed over the input. Attention is computed using the *global attentional model* with *general* score function and *input-feeding*, as described in [27]. The predicted token is then fed into a 128-dimensional embedding layer followed by a three-layer LSTM to update the recurrent state; each hidden state has 256 dimensions.

[1] Using a shared vocabulary of characters or subwords is an interesting direction for future work, but one we don't explore here.

[2] `https://www.ethnologue.com/language/mdw`

[3] In preliminary experiments, we did not find much difference between Mel filterbank features and MFCCs.

While training, we use the predicted token 20% of the time as input to the next decoder step; and the training token for the remaining 80% of the time [28]. At test time we use beam decoding with a beam size of 5 and length normalization [29] with a weight of 0.6.

Training and implementation. Parameters for CNN and RNN layers are initialized by sampling from Gaussian distribution with mean 0, and standard deviation scaled by the number of input units [30]. For embedding and fully-connected layers, we use Chainer [31] default initializer settings.

We regularize using dropout [32], with a ratio of 0.3 over the embedding and the LSTM layers [33], and a weight decay rate of 0.0001. The parameters are optimized using Adam [34], with a starting alpha of 0.001.

Following some preliminary experimentation on our development set, we add Gaussian noise with standard deviation of 0.25 to the MFCC features during training, and drop frames with a probability of 0.10. After 20 epochs, we corrupt the true decoder labels by sampling a random output label with a probability of 0.3.

Our code is implemented in Chainer [31] and freely available.[4]

3.4. Evaluation

Metrics. We report BLEU [35] for all our models.[5] In low-resource settings, BLEU scores tend to be low, difficult to interpret, and poorly correlated with model performance. This is because BLEU requires exact four-gram matches only, but low four-gram accuracy may obscure a high unigram accuracy and inexact translations that partially capture the semantics of an utterance, and these can still be very useful in situations like language documentation and crisis response. Therefore, we also report word-level unigram precision and recall, taking into account *stem*, *synonym*, and *paraphrase* matches. To compute these scores, we use METEOR [37] with default settings for English and French.[6] For example, METEOR assigns "eat" a recall of 1 against reference "eat" and a recall of 0.8 against reference "feed", which it considers a synonym match.

Naive baselines. We also include evaluation scores for a naive baseline model that predicts the *K* most frequent words of the training set as a bag of words for each test utterance. In each experimental condition, we set *K* to be the value at which precision/recall are most similar, which is always between 5 and 20 words. The purpose of this baseline is to provide an empirical lower bound on precision and recall, since we would expect any usable model to outperform a system that does not even depend on the input utterance. We do not compute BLEU for these baselines, since they do not predict sequences, only bags of words.

4. ASR RESULTS

Using the experimental setup of Section 3, we pre-trained ASR models in English and French. To put our subsequent results in context, we report their word error rates (WER) on development data in Table 1.[7] We denote each ASR model by *L-Nh*, where *L* is a language code and *N* is the size of the training set in hours. For example, *en-300h* denotes an English ASR model trained on 300 hours of data; *fr-20h*, a French ASR model trained on 20 hours of data.

Training ASR models for state-of-the-art performance requires substantial hyper-parameter tuning and long training times. For example, it took 3 days for our *en-300h* model to reach a WER of 27.3% on development data, though we observed that it continued to improve for three more days. Since our goal is simply to see whether pre-training is useful, and not to produce state-of-the-art ASR systems, we stopped pre-training our models after around 30 epochs to focus on transfer experiments. As a consequence, our ASR results are far from state-of-the-art: current end-to-end Kaldi systems obtain 16% WER on Switchboard *train-dev*, and 22.7% WER on the French Globalphone dev set.[8] We believe that better ASR pre-training may produce better ST results.

	en-100h	**en-300h**	**fr-20h**
WER	35.4	27.3	29.6

Table 1. Word Error Rate (WER). Computed on Switchboard *train-dev* for English; Globalphone dev for French.

5. SPANISH-ENGLISH ST

In the following, we denote an ST model by *S-T-Nh*, where *S* and *T* are source and target language codes, and *N* is the size of the training set in hours. For example, *sp-en-20h* denotes a Spanish-English ST model trained using 20 hours of data; *mb-fr-4h* denotes a Mboshi-French ST model trained on 4 hours of data.

5.1. Using English ASR to improve Spanish-English ST

Figure 2 plots BLEU (a) and unigram precision/recall (b) on the development set for baseline Spanish-English ST models and those trained after initializing with the *en-300h* model. Corresponding results on the test set (Table 2) reveal very similar patterns, so the remainder of our analysis is confined to the development set. The naive baseline, which predicts the 15 most frequent English words in the training set, achieves a precision/recall around 20, setting a performance lower bound.

Low-resource: 20-50 hours of ST training data. Without transfer learning, our baseline ST models substantially improve over previous results [8] using the same train/ test splits, primarily due to better regularization and modeling of subwords rather than words. Yet transfer learning still substantially improves these strong baselines. For *sp-en-20h*, transfer learning improves dev set BLEU from 10.8 to 19.9, precision from 41% to 51%, and recall from 38% to 49%. For *sp-en-50h*, transfer learning improves BLEU from 23.3 to 27.8, precision from 54% to 58%, and recall from 51% to 56%.

Very low-resource: 10 hours or less of ST training data. Figure 2 shows that without transfer learning, ST models trained on less than 10 hours of data struggle to learn, with precision/recall scores close to or below the lower bound provided by the naive baseline. But with transfer learning, we see gains in precision and recall of between 10 and 20 points.

We also see that with transfer learning, a model trained on only 5 hours of ST data achieves a BLEU of 9.1, nearly as good as the 10.8

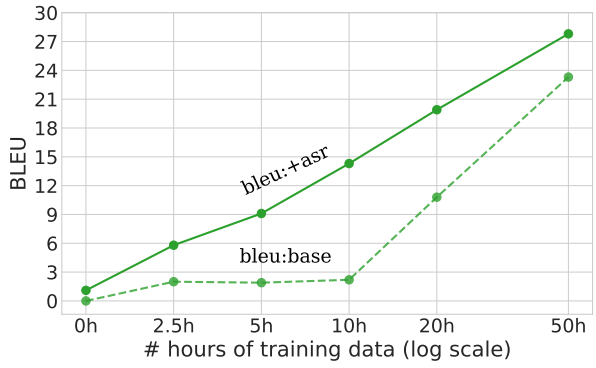
[4] `https://github.com/0xSameer/speech2text/tree/seq2seq`

[5] We compute BLEU with `multi-bleu.pl` from the Moses toolkit [36].

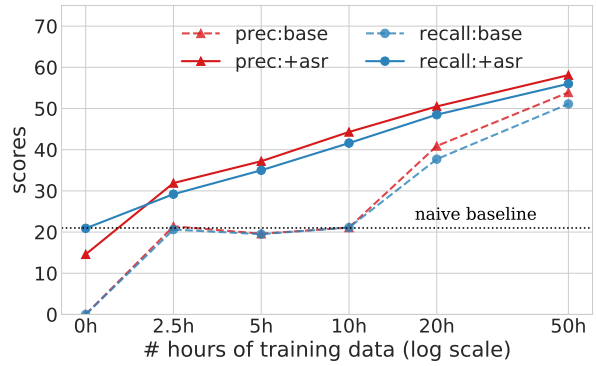
[6] `http://www.cs.cmu.edu/~alavie/METEOR/`

[7] We computed WER with the NIST `sclite` script.

[8] These WER results taken from respective Kaldi recipes on Github, and may not even represent the best results on these datasets.



(a)



(b)

Fig. 2. (a) BLEU and (b) Unigram precision/recall for Spanish-English ST models computed on Fisher dev set. **base** indicates no transfer learning; **+asr** are models trained by fine-tuning *en-300h* model parameters. *naive baseline* indicates the score when we predict the 15 most frequent English words in the training set.

	sp-en-*N*h					
N	*0h*	*2.5h*	*5h*	*10h*	*20h*	*50h*
baseline	0	2.1	1.8	2.1	10.8	22.7
+en-300h	0.5	5.7	9.1	14.5	20.2	28.2

Table 2. Fisher test set BLEU scores for Spanish-English ST. **baseline**: model without transfer learning. **+en-300h**: using model parameters from English ASR.

Spanish	buenas ah yo me llamo jenny y estoy llamando de nueva york
English	greetings ah my name is jenny and i'm calling from new york
20h	well ah ah my name is i'm calling from new york
20h+asr	good ah my name is jenny i'm calling from new york
50h	good ah my name is jenny i'm calling from new york
50h+asr	good ah my name is jenny i'm calling from new york
Spanish	super caliente pero muy bonito
English	super hot but very nice
20h	you support it but it was very nice
20h+asr	you can get alright but it's very nice
50h	super expensive but very nice
50h+asr	super hot but it's very nice
Spanish	sí y usted hace mucho tiempo que que vive aquí
English	yes and have you been living here a long time
20h	yes i've been a long time what did you come here
20h+asr	yes and you have a long time that you live here
50h	yes you are a long time that you live here
50h+asr	yes and have you been here long

Table 3. Example translations on selected sentences from the Fisher development set, with stem-level n-gram matches to the reference sentence underlined. **20h** and **50h** are Spanish-English models without pre-training; **20h+asr** and **50h+asr** are pre-trained on 300 hours of English ASR.

of a model trained on 20 hours of ST data without transfer learning. In other words, fine-tuning an English ASR model—which is relatively easy to obtain—produces similar results to training an ST model on four times as much data, which may be difficult to obtain.

We even find that in the very low-resource setting of just 2.5 hours of ST data, transfer from a pre-trained ASR model achieves a precision/recall around 30 and improves by more than 10 points over the naive baseline. In very low-resource scenarios with time constraints—such as in disaster relief—it is possible that even this level of performance may be useful, since it can be used to spot keywords in speech and can be trained in just three hours.

Sample translations. Table 3 shows some example translations for models *sp-en-20h* and *sp-en-50h* with and without transfer learning using *en-300h*, representing the four rightmost systems in each graph of Figure 2. Improvements are more noticeable between the baseline *sp-en-20h* model and the remaining models, as we might expect from the improvement of 9 BLEU.

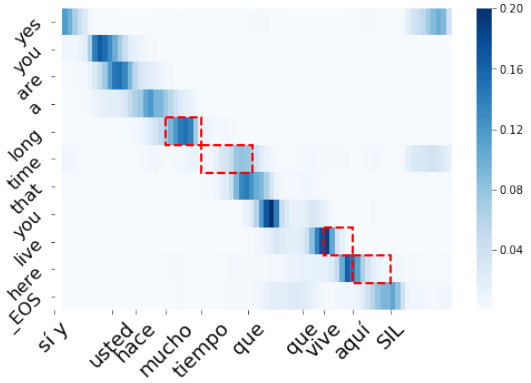
Figure 3 plots the attention weights for the last sample utterance in Table 3. For this utterance, the Spanish and English text have a different word order. *mucho tiempo* occurs in the middle of the speech utterance, and its translation, *long time*, is at the end of the English reference. Similarly, *vive aquí* occurs at the end of the speech utterance, but the translation, *living here*, is in the middle of the English reference. The baseline *sp-en-50h* model translates the words correctly but doesn't get the English word order right. With transfer learning, the model produces a shorter but still accurate translation in the correct word order.

5.2. Analysis

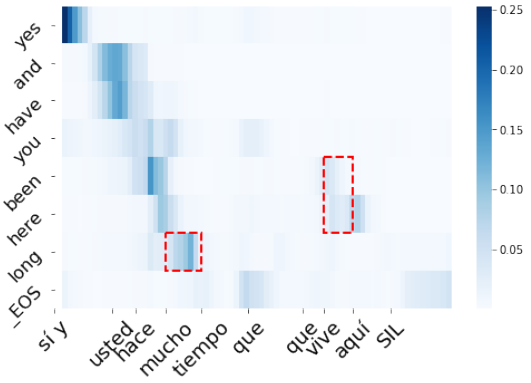
To understand the source of these improvements, we carried out a set of ablation experiments. For most of these experiments, we focus on Spanish-English ST with 20 hours of training data, with and without transfer learning.

Transfer learning with selected parameters. In our first set of experiments, we transferred all parameters of the *en-300h* model, including the speech encoder CNN and LSTM; the text decoder embedding, LSTM and output layer parameters; and attention parameters. Which of these parameters has the most impact? To answer this question, we train the *sp-en-20h* model by transferring only selected parameters from *en-300h*, and randomly initializing the rest. The resulting training curves on the development set are shown in Figure 4.[9]

[9]These scores do not use beam search and are therefore lower than the best scores reported in Figure 2.



(a) **50h:baseline**



(b) **50h:asr**

Fig. 3. Attention plots for the final example in Table 3, using 50h models with and without pre-training. The Spanish reference is *sí y usted hace mucho tiempo que que vive aquí*, and the English reference translation is *yes and have you been living here a long time*. In the reference, *mucho tiempo* is translated to *long time*, and *vive aquí* to *living here*, but their order is reversed, and this is reflected in (b). The x-axis represents the audio data marked with the approximate word occurrence positions; y-axis shows the predicted subwords.

The results show that transferring all parameters provides the best results. But they also show that the speech encoder parameters account for most of the gains. We hypothesize that the encoder learns transferable low-level acoustic features that normalize across variability such as speaker and channel differences, and that much of this learning is language-independent. This hypothesis is supported by other work showing the benefits of cross-lingual and multilingual training for speech technology in low-resource target languages [12, 38–45]. Indeed, there is evidence that speech features trained on multiple languages transfer better than those trained on the same amount of data from a single language [46].

By contrast, transferring only decoder parameters does not improve the overall BLEU score, though it produces some slight improvements early in training. Since decoder parameters help when used in tandem with encoder parameters, we suspect that the dependency in parameter training order might explain this: the transferred decoder parameters have been trained to expect particular input representations from the encoder, so transferring only the decoder parameters without the encoder might not be useful.

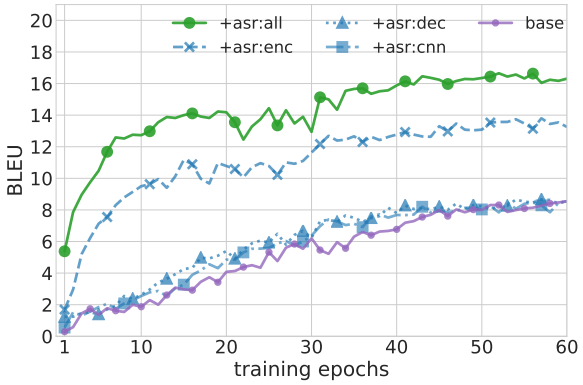


Fig. 4. Fisher development set training curves (reported using BLEU) for *sp-en-20h* using selected parameters from *en-300h*: none (**base**); encoder CNN only (**+asr:cnn**); encoder CNN and LSTM only (**+asr:enc**); decoder only (**+asr:dec**); and all: encoder, attention, and decoder (**+asr:all**).

Figure 4 also suggests that models make strong gains early on in the training when using transfer learning. The *sp-en-20h* model initialized with all model parameters (*+asr:all*) from *en-300h* reaches a higher BLEU score after just 5 epochs (2 hours) of training than the model without transfer learning reaches after 60 epochs/20 hours. This again can be useful in disaster-recovery scenarios, where the time to deploy a working system must be minimized.

Amount of ASR data required. Figure 5 shows the impact of increasing the amount of English ASR data used on Spanish-English ST performance for two models: *sp-en-20h* and *sp-en-50h*.

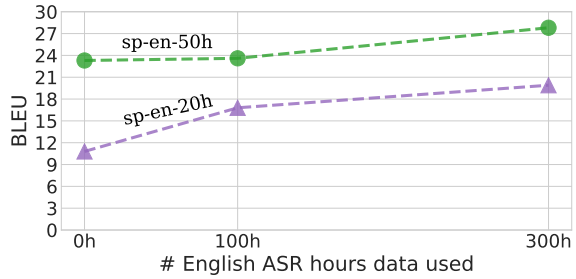


Fig. 5. Spanish-to-English **BLEU** scores on Fisher dev set, with 0h (no transfer learning), 100h and 300h of English ASR data used.

For *sp-en-20h*, we see that using *en-100h* improves performance by almost 6 BLEU points. By using more English ASR training data (*en-300h*) model, the BLEU score increases by almost 9 points.

However, for *sp-en-50h*, improvements are only observed when using *en-300h*. This implies that transfer learning is most useful in true low-resource scenarios, when only a few tens of hours of training data are available for ST. As the amount of ST training data increases, the benefits of transfer learning tail off, although it's possible that using even more monolingual data, or improving the training at the ASR step, could extend the benefits to larger ST datasets.

Impact of codeswitching. We also tried using the *en-300h* ASR model without any fine-tuning to translate Spanish audio to English text. This model achieved a BLEU score of 1.1, with a precision of 15 and recall of 21. The non-zero BLEU score indicates that the model is matching *some* 4-grams in the reference. This seems

to be due to codeswitching in the Fisher-Spanish speech data set. Looking at the dev set utterances, we find several examples where the Spanish transcriptions match the English translations, indicating that the speaker switched into English. For example, there is an utterance whose Spanish transcription and English translation are both "right yeah", and this English expression is indeed present in the source audio. The English ASR model correctly translates this utterance, which is unsurprising since the phrase "right yeah" occurs nearly 500 times in Switchboard.

Overall, we find that nearly 500 of the 4,000 development set utterances (14%) contain likely codeswitching, since the Spanish transcription and English translations share more than half of their tokens. This suggests that transfer learning from English ASR models might help more than from other languages. To isolate this effect from transfer learning of language-independent speech features, we carried out a further experiment.

5.3. Using French ASR to improve Spanish-English ST

In this experiment, we pre-train using French ASR data for a Spanish-English translation task. Here, we can only transfer the speech encoder parameters, and there should be little if any benefit due to codeswitching.

Because our French data set (20 hours) is much smaller than our English one (300 hours), for a fair comparison we used a 20 hour subset of the English data for pre-training in this experiment. For both the English and French models, we transferred only the encoder parameters.

Table 4 shows that both the English and French 20-hour pre-trained models improve performance on Spanish-English ST. The English model works slightly better, as would be predicted given our discussion of codeswitching, but the French model is also useful, improving BLEU from 10.8 to 12.5. This result strengthens the claim that ASR pre-training on a completely distinct third language can help low-resource ST. Presumably benefits would be much greater if we used a larger ASR dataset, as we did with English above.

	baseline	**+fr-20h**	**+en-20h**
sp-en-20h	10.8	12.5	13.2

Table 4. Fisher dev set BLEU scores for *sp-en-20h*. **baseline**: model without transfer learning. **+fr-20h**: using model parameters from French ASR. **+en-20h**: using model parameters from English ASR. Only encoder parameters are transferred.

6. MBOSHI-FRENCH ST

Our final set of experiments test our transfer method on ST for the low-resource language Mboshi, where we have only 4 hours of ST training data: Mboshi speech input paired with French text output.

Table 5 shows the ST model scores for Mboshi-French with and without using transfer learning. The first two rows *fr-top-8w*, *fr-top-10w*, show precision and recall scores for the *naive baselines* where we predict the top 8 or 10 most frequent French words in the Mboshi-French training set. These show that a precision/recall in the low 20s is easy to achieve, although with no n-gram matches (0 BLEU). The pre-trained ASR models by themselves (next two lines) are much worse.

The baseline model trained only on ST data actually has lower precision/recall than the naive baseline, although its non-zero BLEU score indicates that it is able to correctly predict some n-grams. We see comparable precision/recall to the naive baseline with improvements in BLEU by transferring either French ASR parameters (both encoder and decoder, *fr-20h*) or English ASR parameters (encoder only, *en-300h*).

model	**pre-training**	**BLEU**	**Precision**	**Recall**
fr-top-8w	–	0	23.5	22.2
fr-top-10w	–	0	20.6	24.5
en-300h	–	0	0.2	5.7
fr-20h	–	0	4.1	3.2
mb-fr-4h	–	3.5	18.6	19.4
	fr-20h	5.9	23.6	20.9
	en-300h	5.3	23.5	22.6
	en-300h + fr-20h	**7.1**	**26.7**	**23.1**

Table 5. Mboshi-to-French translation scores, with and without ASR pre-training. **fr-top-8w** and **fr-top-10w** are *naive baselines* that, respectively, predict the 8 or 10 most frequent training words. For **en-300h + fr-20h**, we use encoder parameters from *en-300h* and attention+decoder parameters from *fr-20h*

Finally, to achieve the benefits of both the larger training size for the encoder and the matching language of the decoder, we tried transferring the encoding parameters from the *en-300h* model and the decoding parameters from the *fr-20h* model. This configuration (*en+fr*) gives us the best evaluation scores on all metrics, and highlights the flexibility of our framework. Nevertheless, the 4-hour scenario is clearly a very challenging one.

7. CONCLUSION

This paper introduced the idea of pre-training an end-to-end speech translation system involving a low-resource language using ASR training data from a higher-resource language. We showed that large gains are possible: for example, we achieved an improvement of 9 BLEU points for a Spanish-English ST model with 20 hours of parallel data and 300 hours of English ASR data. Moreover, the pre-trained model trains more rapidly than the baseline, achieving higher BLEU than the baseline in only a couple of hours, while the baseline trains for more than a day.

We also showed that these methods can be used effectively on a real low-resource language, Mboshi, with only 4 hours of parallel data. The very small size of the data set makes the task challenging, but by combining parameters from an English encoder and French decoder, we outperformed baseline models to obtain a BLEU score of 7.1 and precision/recall of about 25%. We believe ours is the first paper to report word-level BLEU scores on this dataset.

Our analysis indicated that, other things being equal, transferring both encoder and decoder parameters works better than just transferring one or the other. But transferring the encoder parameters is where most of the benefit comes from, so if the choice is between pre-training using a large ASR corpus from a mismatched language or a smaller ASR corpus that matches the output language, using the larger ASR corpus will probably work better.

Despite impressive gains, these experiments only scratch the surface of possible transfer strategies, and our analysis suggests several avenues for further exploration. On the speech side, it might be even more effective to use multilingual training; or to replacing the MFCC input features with pre-trained multilingual features, or features that

are targeted to low-resource multispeaker settings [47, 48]. On the language modeling side, simply transferring decoder parameters from an ASR model did not work; but an alternative, and perhaps better, method would be to use pre-trained decoder parameters from a language model, as proposed by Ramachandran et al. [49], or *shallow fusion* [50], which interpolates a pre-trained language model during beam search. In these methods, the decoder parameters are independent, and can therefore be used on their own. We plan to explore this space of strategies in future work.

8. ACKNOWLEDGMENTS

This work was supported in part by a James S McDonnell Foundation Scholar Award, a Google faculty research award, and NSF grant 1816627. We thank Ida Szubert and Clara Vania for helpful comments on previous drafts of this paper and Antonios Anastasopoulos for tips on experimental setup.

9. REFERENCES

[1] Laurent Besacier, Bowen Zhou, and Yuqing Gao, "Towards speech translation of non written languages," in *Proc. SLT*, 2006.

[2] Lara J Martin, Andrew Wilkinson, Sai Sumanth Miryala, Vivian Robison, and Alan W Black, "Utterance classification in speech-to-speech translation for zero-resource languages in the hospital administration domain," in *Proc. ASRU*, 2015.

[3] Oliver Adams, Graham Neubig, Trevor Cohn, and Steven Bird, "Learning a translation model from word lattices.," in *Proc. Interspeech*, 2016.

[4] Oliver Adams, Graham Neubig, Trevor Cohn, Steven Bird, Quoc Truong Do, and Satoshi Nakamura, "Learning a lexicon and translation model from phoneme lattices," in *Proc. EMNLP*, 2016.

[5] Antonios Anastasopoulos and David Chiang, "A case study on using speech-to-translation alignments for language documentation," in *Proc. ACL*, 2017.

[6] Robert Munro, "Crowdsourced translation for emergency response in Haiti: The global collaboration of local knowledge," in *AMTA Workshop Collaborative Crowdsourcing Transl.*, 2010.

[7] Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen, "Sequence-to-sequence models can directly transcribe foreign speech," in *Proc. Interspeech*, 2017.

[8] Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater, "Low-resource speech-to-text translation," in *Proc. Interspeech*, 2018.

[9] Antonios Anastasopoulos and David Chiang, "Tied multitask learning for neural speech translation," in *Proc. NAACL HLT*, 2018.

[10] Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin, "End-to-end automatic speech translation of audiobooks," in *Proc. ICASSP*, 2018.

[11] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual Speech Recognition with A Single End-To-End Model," in *Proc. ICASSP*, 2018.

[12] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Mike Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, and Alex Acero, "Recent advances in deep learning for speech research at Microsoft.," in *Proc. ICASSP*, 2013.

[13] Sebastian Thrun, "Is learning the n-th thing any easier than learning the first?," in *Proc. NIPS*, 1995.

[14] Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight, "Transfer learning for low-resource neural machine translation," in *Proc. EMNLP*, 2016.

[15] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Trans. ACL*, vol. 5, pp. 339–351, 2017.

[16] John Godfrey and Edward Holliman, "Switchboard-1 Release 2 (LDC97S62)," `https://catalog.ldc.upenn.edu/ldc97s62`.

[17] Tanja Schultz, "Globalphone: a multilingual speech and text database developed at karlsruhe university," in *Seventh International Conference on Spoken Language Processing*, 2002.

[18] David Graff, Shudong Huang, Ingrid Cartagena, Kevin Walker, and Christopher Cieri, "Fisher Spanish Speech (LDC2010S01)," `https://catalog.ldc.upenn.edu/ldc2010s01`.

[19] Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur, "Improved speech-to-text translation with the Fisher and Callhome Spanish-English speech translation corpus," in *Proc. IWSLT*, 2013.

[20] Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur, "Fisher and CALLHOME Spanish–English Speech Translation," `https://catalog.ldc.upenn.edu/ldc2014t23`.

[21] Pierre Godard, Gilles Adda, Martine Adda-Decker, Juan Benjumea, Laurent Besacier, Jamison Cooper-Leavitt, Guy-No"el Kouarata, Lori Lamel, H'el'ene Maynard, Markus M"uller, Annie Rialland, Sebastian St"uker, François Yvon, and Marcely Zanon Boito, "A very low resource language speech corpus for computational language documentation experiments," *CoRR*, vol. abs/1710.03501, 2017.

[22] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. ASRU*, 2011.

[23] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016.

[24] Vinod Nair and Geoffrey E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, 2010.

[25] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[26] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural Comput.*, 1997.

[27] Minh-Thang Luong, Hieu Pham, and Christopher D Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015.

[28] Ronald J. Williams and David Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, 1989.

[29] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on Imagenet classification," in *Proc. ICCV*, 2015.

[31] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton, "Chainer: A next-generation open source framework for deep learning," in *Proc. LearningSys*, 2015.

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, 2014.

[33] Yarin Gal, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. NIPS*, 2016.

[34] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL*, 2002.

[36] 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 *Proc. ACL*, 2007.

[37] Alon Lavie and Abhaya Agarwal, "Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments," in *Proc. WMT*, 2007.

[38] Michael A Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky, "Rapid evaluation of speech representations for spoken term discovery.," in *Proc. Interspeech*, 2011.

[39] Aren Jansen, Kenneth Church, and Hynek Hermansky, "Towards spoken term discovery at scale with zero resources," in *Proc. Interspeech*, 2010.

[40] Ngoc Thang Vu, Wojtek Breiter, Florian Metze, and Tanja Schultz, "An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance," in *Proc. Interspeech*, 2012.

[41] Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky, "Multilingual mlp features for low-resource LVCSR systems," in *Proc. ICASSP*, 2012.

[42] Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, Abhinav Sethy, Kartik Audhkhasi, Xiaodong Cui, Ellen Kislal, Lidia Mangu, Markus Nussbaum-Thom, Michael Picheny, et al., "Multilingual representations for low resource speech recognition and keyword search," in *Proc. ASRU*, 2015.

[43] Tanel Alumäe, Stavros Tsakalidis, and Richard M Schwartz, "Improved multilingual training of stacked neural network acoustic models for low resource languages," in *Proc. Interspeech*, 2016.

[44] Yougen Yuan, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li, "Learning neural network representations using cross-lingual bottleneck features with word-pair information," in *Proc. Interspeech*, 2016.

[45] Daniel Renshaw, Herman Kamper, Aren Jansen, and Sharon Goldwater, "A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge," in *Proc. Interspeech*, 2015.

[46] Enno Hermann and Sharon Goldwater, "Multilingual bottleneck features for subword modeling in zero-resource languages," in *Proc. Interspeech*, 2018.

[47] H. Kamper, M. Elsner, A. Jansen, and S. J. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Proc. ICASSP*, 2015.

[48] Herman Kamper, Aren Jansen, and Sharon Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *ArXiv e-prints*, 2016.

[49] Prajit Ramachandran, Peter J Liu, and Quoc V Le, "Unsupervised pretraining for sequence to sequence learning," in *Proc. EMNLP*, 2017.

[50] Caglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loıc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "On using monolingual corpora in neural machine translation," *CoRR*, vol. abs/1503.03535, 2015.