JHU IWSLT 2022 Dialect Speech Translation System Description

Jinyi Yang[†*] **Amir Hussein**[†*] **Matthew Wiesner**[‡] **Sanjeev Khudanpur**[†‡]
[†]Johns Hopkins University
[‡] Human Language Technology Center of Excellence
{jyang126,ahussei6,wiesner,khudanpur}@jhu.edu

Abstract

This paper details the Johns Hopkins speech translation (ST) system used in the IWLST2022 dialect speech translation task. Our system uses a cascade of automatic speech recognition (ASR) and machine translation (MT). We use a Conformer model for ASR systems and a Transformer model for machine translation. Surprisingly, we found that while using additional ASR training data resulted in only a negligible change in performance as measured by BLEU or word error rate (WER), aggressive text normalization improved BLEU more significantly. We also describe an approach, similar to back-translation, for improving performance using synthetic dialect source text produced from source sentences in mismatched dialects.

1 Introduction

In this paper we describe the JHU dialect speech translation submissions and their development. Dialects are varieties of a language spoken by a group of people, often in a specific geographic location. In many languages, standard rules of pronunciation, orthography and syntax, but also available data resources are drawn from a single dominant dialect. A challenge for all language technologies, including automatic speech recognition (ASR), machine translation (MT), and speech translation (ST), is how to deal with non-standard dialects for which no formal orthography, grammar, or even data exist. Because many dialects are rarely if ever written, evaluation of ASR and MT on dialect speech is not even particularly well defined. However, there are no such problems evaluating speech translation on dialect speech, which here refers to the task of producing target language text from source language audio inputs.

A focus of both the dialect speech translation task and our system development, is how to leverage available resources from the standard dialect to improve performance on non-standard dialects. The dialect translation task focuses specifically on Tunisian Arabic.

Arabic and its dialects lie on a *dialect continuum* unified by a single standardized dialect, Modern standard Arabic (MSA) (Badawi et al., 2013). MSA is the primary language of *formal* and *written* communications (e.g. news broadcasts, parliaments and religion). However, most native Arabic speakers use local *dialects* in daily life, which generally lack a standard written form. Certain dialects, such as Algerian, Tunisian, and Moroccan Arabic also have strong Romance, and Berber substrates, and may exhibit a high degree of code-switching, especially with French.

Traditionally, speech translation systems have been built by cascading ASR and MT models to form a speech translation chain (Dixon et al., 2011). However, the more recent end-to-end approach (Berard et al., 2016; Weiss et al., 2017), which directly translates the source speech to target text, is appealing for this task since since both ASR and MT are ill-defined for unwritten spoken dialects, and there were relatively large amounts of translated speech (∼160 hrs). We found, somewhat surprisingly during initial experimentation (See rows 1,2 of Table 7), that cascaded systems outperformed their end-to-end counterparts. For this reason, we focused on building cascaded systems. We leave diagnosis of the worse performance of the end-to-end systems to future work.

Our systems incorporated three improvements over the provided baseline. 1. We aggressively normalized the Tunisian Arabic transcripts, which led to improved MT performance. 2. We use additional MSA bi-text by pretraining models on these data using a shared BPE model with a large number of BPE units for both the MSA and Tunisian data. 3. We show that training on synthetic Tunisian source sentences instead of the MSA source sentences provides small improvements.

[*]Equal contribution.

Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022), pages 319 - 326

2 The Dialect Speech Translation Task

The dialect speech translation task permitted submissions using models trained assuming different resource constraints, called: (A) basic, (B) dialect adaptation, and (C) unconstrained. We refer to these conditions as (A), (B) and (C) in the rest of the paper.

2.1 Data description

The total amount of data for the three conditions is listed in Table 1, with details of train, development and test1 sets in Table 2.

The development and test1 sets are provided by the organizers. The data are 3-way parallel: Tunisian Arabic transcripts and English translations are available for each Tunisian Arabic audio utterance. We use the development set for model comparison and hyperparameter tuning, and the test1 set for evaluating our ST systems. Finally, the task organizers provided a blind evaluation set (test2) during the evaluation period for final comparison of submissions. We used the test2 set to generate English translations, which were scored by the organizers.

For condition (C), we explored using pretrained audio representations trained only on additional unlabeled audio. However, we applied the exact same MT models as used in conditions (A) and (B).

3 Methods

We model the speech translation problem as a two step process. First, input audio is converted to source language text via an ASR model. Next, an MT model, which may have been trained on entirely different data from the ASR model, is used to translate the ASR output transcript into target language sentences. This model is known as a cascade model.

While cascade models suffer from a few well known problems, such as compounding error and inability to make direct use of the acoustic signal to improve translation quality, their modularity facilitates training on and incorporation of additional resources such as transcribed speech, bi-text, monolingual text, and unlabeled source language audio. We describe how we used these available resources to train the ASR and MT models in our ST cascade in each data condition.

3.1 ASR

Condition (A). We train our ASR model using the Tunisian Arabic audio and transcripts from the training set.

Condition (B). The MGB-2 data from condition (B) is used to train a large scale MSA conformer. The parameters of our conformer model are adopted from (Hussein et al., 2022). Then the pretrained model is fine-tuned on the Tunisian training data from condition (A). There are several sources of domain mismatch since the Tunisian data is sampled at 8KHz from telephone channel and the MGB-2 is sampled at 16KHz from broadcast news. As a result in this work we compare between two domain matching strategies for pretraining and fine-tuning: 1) Pretrain on 16KHz microphone data and fine-tune on up-sampled 16KHz telephone data, 2) Pretrain on down-sampled 8KHz microphone data and fine-tune on 8KHz telephone data.

Condition (C). We use the pretrained Wav2Vec2 multilingual model, XLSR-53 (Conneau et al., 2021) and fine-tune with the training data from condition (A). This model was trained on unlabeled speech in 53 languages, but notably, 1,000+hr of telephone conversations in 17 languages. There are some read prompts in Arabic, as well as a significant amount of French, which we suspect makes this model a better suited starting point for a Tunisian dialect ASR system.

3.2 MT

We use a transformer architecture for our MT models in condition (A) and (B). The model sizes are adjusted according to the amount of training data. We did not train MT models with extra data from condition (C).

Condition (A). We use the training data from condition (A). Two Byte-pair encoding (BPE) models were separately trained for Tunisian and English and applied to train, development and test1 sets. The trained model is referred as "*Ta2En-basic*".

Condition (B). We used two adaptation approaches. The first one is fine-tuning. We combine the Tunisian and MSA text to train a universal Arabic BPE model and use it to encode all the Arabic text. We also combine the English text from condition (A) and (B) to train an English BPE model and encode all the English text; an MT model, which

Condition	ASR	MT
(A) Basic	166 hours of manually transcribed Tunisian speech	∼212 k lines of manually translated English from Tunisian
(B) Dialect adaptation	1200 hours of Modern Standard Arabic (MSA) broadcast news speech with transcripts from MGB-2 (Ali et al., 2016)	∼42,000k lines of bitext in MSA-English for MT from the organizers (downloaded from OPUS (Tiedemann, 2012)
(C) Unconstrained	any English, Arabic dialects, or multilingual models beyond English and Arabic	any English, Arabic dialects, or multilingual models beyond English and Arabic

Table 1: Data for different conditions, provided by the organizers.

	ASR (hours)	MT (lines)
train (condition A)	160	∼202k
train (condition B)	1200+160	∼42M
dev	3.0	3833
test1	3.3	4204
test2	3.6	4288

Table 2: Details for train, dev and test1 sets for condition (A) and (B).

we call "*Msa2En*", is trained with MSA-English data from condition (B). The *Msa2En* model is then fine-tuned with the Tunisian-English data from condition (A), and called "*Msa2En-tune*".

The second method additionally tries to reduce the domain mismatch between conditions (B) and (A). Let $p_\theta(y_t \mid y_s)$, be an MT model with parameters, θ, trained on MSA-English bi-text, that generates English target sentences, y_t, conditioned on source sentences, y_s. Let $p(y_s)$ denote the marginal density over MSA source sentences. Let $q(y_s)$ denote the marginal density over Tunisian Arabic source sentences, and let us assume that the conditional density, $p(y_t \mid y_s)$, between English and MSA sentences, is the same as between English and Tunisian sentences. A good model should ideally then minimize

$$\mathbb{E}_{q(y_s)}\left[D(p(y_t \mid y_s) \parallel p_\theta(y_t \mid y_s))\right], \quad (1)$$

the expected value of the KL-divergence between the model posteriors and ground-truth Tunisian data over the Tunisian data. However, when training on the MSA data, the model is instead trained using

$$\mathbb{E}_{p(y_s)}\left[D(p(y_t \mid y_s) \parallel p_\theta(y_t \mid y_s))\right], \quad (2)$$

i.e, with the empirical MSA data marginal density, $p(y_s)$, instead of the Tunisian marginal, $q(y_s)$. We can reduce this dialect mismatch in training by using an extra back-translation model (Sennrich et al., 2016) to convert MSA text to Tunisian. Formally, we use this back-translation model, $q_\phi(y_s|y'_s)$, with parameters, ϕ, to generate samples that approximate draws from $q(y_s)$. We therefore propose to train our model to minimize

$$\mathbb{E}_{q_\phi(y_s|y'_s)}\left[D(p(y_t \mid y_s) \parallel p_\theta(y_t \mid y_s))\right]. \quad (3)$$

Because we have extra bi-text instead of simply monolingual text, we can choose to either back-translate the MSA source text to Tunisian, using English as a pivot language (i.e., y'_s is an MSA sentence), or we can back-translate directly from the English target text (i.e, $y'_s = y_t$). We trained both back-translation models, but ultimately trained using the MSA to Tunisian model following the steps below:

- Train an English to MSA MT model using the data from Table 2 condition (B). This model is referred to as "*En2Msa*",
- Translate the English from condition (A) to MSA, using the "*En2Msa*" model from the previous step. Thus, we obtain the paired Tunisian-MSA translation data, while the Tunisian are manually transcribed and the MSA are machine-translated.
- Train an MSA to Tunisian MT model, which we call "*Msa2Ta*", i.e., $q_\phi(y|y')$, with training data from the previous step.
- Translate the MSA from condition (B) to Tunisian, using the "*Msa2Ta*" model from the previous step from which we obtain around 42,000k pairs of Tunisian-English MT data.
- Train a Tunisian to English model with the data obtained from the previous step, referred as "*Ta2En-bt*".

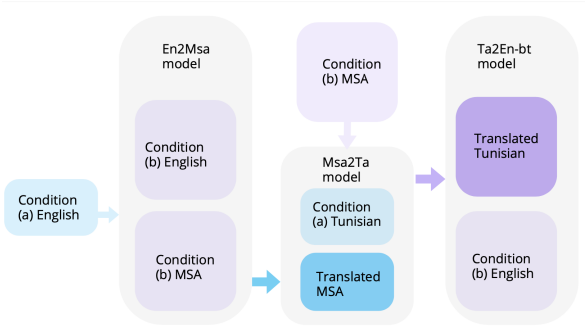

Figure 1: Generation of the back-translation model, $q_\phi(y_s \mid y'_s)$, used in our MT system. The En2Msa model is trained using the Condition (b) bi-text. The target English data from Condition (a) is passed through the En2Msa model to generate Condition (a) MSA source sentences (Translated MSA). We train an Msa2Ta model, i.e., $q_\phi(y_s \mid y'_s)$, using the Condition (a) Tunisian and Translated MSA. All Condition (b) MSA data is converted to Tunisian (Translated Tunisian). The final Ta2En-bt model is trained using the Translated Tunisian data as source sentences instead of the original Condition (b) MSA data.

- Fine-tune the above model, with data from condition (A), this model is referred to as "*Ta2En-bt-tune*".

The steps are illustrated in Figure 1, except the last step for fine-tuning.

We attempted to benchmark the different back-translation approaches by comparing the *En2Msa* + *Msa2Ta* cascade on the dev and test1 sets against the simpler, direct *En2Ta* approach using a single "*En2Ta*" model trained using the transcripts and translations from condition (A). However, the comparison is not completely fair. We also report performance of the *En2Msa* model on the condition (B) development and test sets, which each contains 40,000 randomly selected sentences from the six subsets from OPUS. Results are shown in table 3.

First, we see that the *En2Msa* model performs fairly well, with a BLEU score above 30, which is significantly higher than translation from English to Tunisian (row $En2Ta$). Next, comparing the rows $En2Ta$ and $Msa2Ta$, it appears that direct translation from English to Tunisian performs better. However, the *Msa2Ta* model may appear to perform artificially worse due to domain mismatch between the condition (B) and (A) English targets, as well as due to compounding errors from the sequential use of the 2 translation models, *En2Msa*, and *Msa2Ta*. We will conduct a "real" evaluation of our "Msa2Ta" model using ground-truth MSA-TA data (rather than synthetic MSA) in future work.

Model	dev	test1
En2Msa	31.7	31.4
En2Ta	14.2	12.1
Msa2Ta	10.6	10.6

Table 3: BLEU scores evaluating the back-translation quality of the En2Msa, En2Ta and Msa2Ta models.

4 Experiments

To test our approach, we conducted experiments on the ASR, MT, and ST tasks. In all experiments, unless otherwise stated we performed additional text normalization in order to reduce some of the orthographic variation in the Tunisian transcripts. In all experiments and for all languages / dialects, we remove punctuation, using the scripts provided by the organizer.[1]

For both Tunisian and MSA, we convert eastern Arabic digits to western Arabic digits, and remove diacritics and single character words. We also perform Alif/Ya/Ta-Marbuta normalization, which removes distinctions within three sets of characters that are often written inconsistently in dialect Arabic and even sometimes in modern standard Arabic: Alif forms (A = ا ,أ , إ, آ), Ya forms (y = ي , Y = ى, and Ta-Marbuta forms (p = ة , h = ه). For English, we keep all the text in lowercase, as the evaluation is performed on lowercased English text, and we use MOSES (Koehn et al., 2007) for text tokenization. It is difficult to assess the normalization affect on the quality of the ASR. However, we can measure its effect on the downstream task of translation, described in section 4.2.

4.1 ASR experiments

We tested to what extent additional MSA resources might benefit the ASR performance on the Tunisian dialect data. All models for conditions (A) and (B) are trained using Espnet (Watanabe et al., 2018) using the hybrid attention / CTC architecture (Watanabe et al., 2017) and decoding (Hori et al., 2017).

Baseline-small. We improve the Baseline end-to-end conformer model provided by the organizer[2] by reducing its number of parameters: BPE units 1000 -> 500, CNN sub-sampling kernel 31 -> 15. This

[1] https://github.com/kevinduh/iwslt22-dialect
[2] https://github.com/espnet/espnet/blob/master/egs2/iwslt22_dialect/asr1

model is trained with only the Tunisian data from condition (A). The details of the Baseline-small architecture are provided in Table 4.

MGB-tune. The provided MGB-2 data from condition (B) was used to pretrain a large conformer model, with parameters parameters adopted from (Hussein et al., 2022) as shown in Table 4. Then the pretrained model is fine-tuned on Tunisian data from condition (A) by updating all model parameters with $1/10$ of the learning rate that was used during the training similar to (Hussein et al., 2021). The original MGB-2 dataset comes with very long segments >100 seconds. We noticed that training on these segments was preventing the model from converging. As a result we used a better MGB-2 segmentation from (Mubarak et al., 2021) which has segments of maximum length of 15 seconds.

Table 4: Values of Baseline-small hyperparameters CNN: refers to CNN module kernel, Att: attention, Enc: encoder, Dec: decoder, and FF: fully connected layer

Model	BPE	Att heads	CNN	Enc layers	Dec layers	d^k	FF units
Baseline-small	500	4	15	8	4	512	2048
MGB-tune	5000	8	31	12	6	512	2048

MGB2-tune-trans is a pretrained transformer (Hussein et al., 2022) on 16KHz MGB-2 and then fine-tuned. This is the state-of-the-art ASR transformer model on MGB-2 test set.

MGB2-tune-conf is a conformer trained on MGB-2 16KHz. The training hyperparameters are similar to the *MGB2-tune-trans* model.

MGB2-tune-best is the same model structure as *MGB2-tune-conf*, except that the MGB-2 speech recordings are down sampled from 16KHz to 8KHz.

Wav2Vec2. For the unconstrained submissions we fine-tuned the self-supervised, Wav2Vec2 model XLSR-53. We fine-tune these models, generally following the method described in (Baevski et al., 2020): we added a single additional linear layer at the output of the XLSR-53 model corresponding to the number of BPE units, and fine-tuned using the CTC loss on the the normalized target transcripts. Baevski et al. (2020), only use character outputs, but since many vowels are not written in Arabic, we opted to instead use a small number of BPE units (400, which is roughly the number of digraphs in Arabic) so that hidden vowels might be modeled by surrounding context. As in (Baevski et al., 2020), we froze only the feature-extractor, i.e., the convolutional layers in the model during fine-tuning. We trained with the Adam optimizer, using a learning rate of 1e-05, with 8000 warmup steps, after which the learning rate was decayed exponentially with a decay rate of 1e-05. We used a gradient threshold of 5.0, and a weight decay of 1e-06.

We decode using a WFST decoder for CTC models (Miao et al., 2015) implemented in k2.[3] We trained a 3-gram language model on the Tunisian transcripts, and used a "pronunciation" lexicon mapping words to BPE units. We augmented the fixed vocabulary with the BPE units themselves, which enables the decoder to decode OOVs (about 5% of the tokens), by taking back-off transitions in the language model.

Looking at rows "(A) Baseline" and "(C) Wav2Vec2-tune" in Table 5, we see that fine-tuning the XLSR-53 model provided very marginal gains over the baseline model.

	MGB-2		TA	
Model	dev	test	dev	test1
(A) Baseline	-	-	40.8	45.2
(A) Baseline-small	-	-	40.8	44.8
(B) MGB2-tune-trans	14.6	14.2	40.5	44.1
(B) MGB2-tune-conf	**13.0**	**13.2**	40.1	44.9
(B) MGB2-tune-best	**13.0**	13.3	**38.8**	**43.8**
(C) Wav2Vec2-tune	-	-	40.6	44.5

Table 5: WER (%) of ASR models.

The best ASR peformance on the TA test1 set is achieved by *MGB2-tune-best*. This model is a large conformer model pre-trained on down-sampled 8KHz MGB-2 data and fine-tuned on the Tunisian training data. The *MGB2-tune-conf* model achieves (to our knowledge) a new state-of-the-art on the MGB-2 dataset, with relative improvements of 10% on dev and 7% on the test MGB-2, comparing to *MGB2-tune-trans*.

4.2 MT experiments

We train the MT models as described in Section 3.2, with Fairseq (Ott et al., 2019). We use Sacrebleu (Post, 2018) to compute the case-insensitive (all text in lowercase) BLEU (Papineni et al., 2002) scores for the dev and test1 sets. We test models using either the manual, source language transcript ("Gold Source"), or the ASR output ("ASR Source"), as shown in Table 7. The "ASR Source"

[3]https://github.com/k2-fsa/k2

for all the MT models in Table 7 was generated by ASR model "*(A) Baseline*" for fair comparison among MT models.

Condition	**A**	**B**
Encoder layers	6	6
Encoder embed dim	512	512
Encoder ffn embed dim	1024	2048
Encoder attn heads	4	8
Decoder layers	6	6
Decoder embed dim	512	512
Decoder ffn embed dim	1024	2048
Decoder attn heads	4	8

Table 6: MT model parameters. (* "ffn": feed-forward; "attn": attention)

	Gold Source		ASR Source	
Model	dev	test1	dev	test1
(A*) Ta2En-e2e, raw	-	-	16.7	13.7
(A*) Ta2En-basic, raw	24.7	20.9	18.1	15.3
(A) Ta2En-basic	25.3	21.2	18.7	16.1
(B) Msa2En	3.5	2.8	-	-
(B) Msa2En-tune	27.4	**24.2**	19.8	17.0
(B) Ta2En-bt	12.1	11.2	-	-
(B) Ta2En-bt-tune	**27.6**	**24.2**	**19.9**	**17.2**
(B) Ta2En-bt-tune, best	**29.0**	**25.0**	**20.5**	**17.8**

Table 7: BLEU scores of various MT models using either the gold reference transcripts or ASR hypotheses. **Bold** values indicate the best among comparable results. **Bold and underlined** values are the best overall results using different hyperparameters.

Ta2En-basic. The model parameters can be found in Table 6 Condition (A). We use 4000 BPE units for Tunisian Arabic, and 4000 BPE units for English. We train with the Adam optimizer (Kingma and Ba, 2015); each batch contains maximum 4096 tokens; the maxiumum learning rate is 5e-04, attained after 4000 warm-up steps, and then decayed according to an inverse square root scheduler; we use dropout probability of 0.3; the model is trained for 50 epochs.

We first evaluate the effects of Arabic text normalization. Without text normalization, as shown in Table 7 *(A∗) Ta2En-basic,raw*, the BLEU scores are consistently worse on both dev and test1 sets regardless of the input source (gold vs. ASR). Therefore, we use normalized Arabic text for all the other MT experiments. This simple pre-processing was the greatest source of improvement that did not involve training on additional bi-text, or hyper-parameter tuning.

Msa2En and Msa2En-tune. The model parameters can be found in Table 6 Condition (B). We use 2000 BPE units for the combined MSA and Tunisian Arabic, and 2000 BPE units for the combined English from conditions (A) and B. The hyper-parameters are identical to those used when training "Ta2En-basic", except that we increase the batch size to maximum 20000 tokens. When fine-tuning, we reduce the maximum learning rate to 4e-05, and the batch size to 2048 tokens.

Comparing rows *(B) Msa2En* and *(B) Msa2En-tune* in Table 7, we see a large improvement in BLEU scores from this fine-tuning procedure, which is reasonable, since direct application of the *(B) Msa2En* without fine-tuning results in significant dialect and domain mismatch. However, comparing rows *(B) Msa2En-tune* and *(A) Ta2En-basic*, we see that pre-training on unrelated data and fine tuning with in domain data improves the MT performance on both dev and test1 sets.

Ta2En-bt and Ta2En-bt-tune. We then examine to what extent back-translation of MSA source sentences to synthetic Tunisian Arabic text improves adaptation of the MSA MT system. We use the same BPE models as the one used for *Msa2En*, as well as the model parameters and training hyper-parameters. The tuning hyper-parameters are the same as used for the *Msa2En-tune*.

An interesting finding, comparing the *Msa2En* and *Ta2En-bt* models, neither of which is fine-tuned on any Tunisian-English data, is that the *Ta2En-bt* performs, on average, ∼8 BLEU better on the dev and test1 set, which indicates that our method to reduce dialect mismatch between MSA and Tunisian is helpful. After fine tuning, the *Ta2En-bt-tune* still shows some marginal improvement over the *Msa2En-tune* model.

Ta2En-bt-tune, best The training and tuning data are exactly the same as the one used for the *Ta2En-bt-tune*, except that we increased the BPE units from 2000 to 32,000, for both Tunisian and English. We also increased the model size, using the model parameters according to the original implementation (Vaswani et al., 2017). This model gave the best MT performance on both dev and test1 sets.

	MT Model							
	(A) Ta2En-basic			(B) Msa2En-tune		(B) Ta2En-bt-tune, best		
ASR Model	dev	test1	test2	dev	test1	dev	test1	test2
(A) Baseline	18.7	16.1	**17.1**	19.8	17.0	20.7	17.8	**18.9**
(B) MGB2-tune-conf	18.7	15.8	-	19.7	16.9	20.5	17.6	-
(B) MGB2-tune-best	19.1	16.3	-	20.0	17.4	20.7	18.0	-
(C) Wav2Vec2-tune	18.3	15.6	-	19	16.9	20.3	17.5	**18.7**

Table 8: BLEU scores on the dev, test1 and test2. For the submission, for the basic condition, we use ASR model "(A) Baseline" and MT model "(A) Ta2En-basic"; for the dialect adaptation condition, we use ASR model "(A) Baseline" and MT model "(B) Ta2En-bt-tune,best"; for the unconstrained condition, we use ASR model "(C) Wav2Vec2-tune" and MT model "(B) Ta2En-bt-tune,best". The BLEU scores for the evaluation set are in bold text.

4.3 ST experiments

For our cascaded ST system, we chose the ASR and MT models that gave the best BLEU scores on the dev set in each condition. During the evaluation period, we ran our ST system and generated translations of the blind evaluation set (test2); the BLEU scores on this set were calculated by the organizers and provided to our team. The results are listed in Table 8.

For the "Basic condition" submission, we used ASR model: "*(A) Baseline*" and MT model: "*(A) Ta2En-basic*". For the "Dialect adaptation condition" submission, we used ASR model: "*(A) Baseline*" and MT model: "*(B) Ta2En-bt-tune, best*". For the "Unconstrained condition" submission, we used ASR model: "*(C) Wav2Vec2-tune*" and MT model: "*(B) Ta2En-bt-tune, best*".

Note that we actually have better ST performance with ASR model "*(B) MGB2-tune-best*", consistently with all MT model combinations. However, the training of this ASR model was only completed after the evaluation period, therefore we did not use it for our final submission.

5 Conclusion

We have detailed the our submission for the IWSLT 2022 dialect speech translation task. We briefly compared end-to-end to cascaded systems and found that cascaded models were slightly outperforming their end-to-end counterparts despite, a relative abundance of training data.

We demonstrated that increased text normalization, and back-translation to reduce dialect mismatch improved speech translation performance. Finally, we described two ways of using extra mismatched dialect resources and found surprisingly that using additional unlabeled data through the use of the XLSR-53 model resulted in only small improvements. Using additional large labeled MSA resources resulted in slight improvements to the ASR, and modest improvements in MT.

Future work should expand upon the back-translation results to determine the optimal method for minimizing the dialect mismatch when augmenting training with additional bi-text.

6 Acknowledgments

We would like to thank Dr. Ahmed Ali, and Dr. Shammur Chowdhury for their support and guidance as well as the Qatar Computing Research Institute (QCRI) more broadly for providing some of the computational resources that made this work possible.

References

Ahmed M. Ali, Peter Bell, James R. Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2016. The mgb-2 challenge: Arabic multi-dialect broadcast media recognition. *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 279–284.

El Said Badawi, Michael Carter, and Adrian Gully. 2013. *Modern written Arabic: A comprehensive grammar*. Routledge.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Alexandre Berard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *ArXiv*, abs/1612.01744.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdel rahman Mohamed, and Michael Auli. 2021. Unsupervised cross-lingual representation learning for speech recognition. In *Interspeech*.

Paul R. Dixon, Andrew Finch, Chiori Hori, and Hideki Kashioka. 2011. Investigation of the effects of ASR tuning on speech translation performance. In *Proceedings of the 8th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 167–174, San Francisco, California.

Takaaki Hori, Shinji Watanabe, and John Hershey. 2017. Joint CTC/attention decoding for end-to-end speech recognition. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 518–529, Vancouver, Canada. Association for Computational Linguistics.

Amir Hussein, Shammur Chowdhury, and Ahmed Ali. 2021. Kari: Kanari/qcri's end-to-end systems for the interspeech 2021 indian languages code-switching challenge. *arXiv preprint arXiv:2106.05885*.

Amir Hussein, Shinji Watanabe, and Ahmed Ali. 2022. Arabic speech recognition by end-to-end, modular systems and human. *Computer Speech & Language*, 71:101272.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Yajie Miao, Mohammad Abdelaziz Gowayyed, and Florian Metze. 2015. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 167–174.

Hamdy Mubarak, Amir Hussein, Shammur Absar Chowdhury, and Ahmed Ali. 2021. QASR: QCRI aljazeera speech resource a large scale annotated Arabic speech corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2274–2285, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Yalta, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. Espnet: End-to-end speech processing toolkit. *ArXiv*, abs/1804.00015.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey, and Tomoki Hayashi. 2017. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11:1240–1253.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Z. Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *INTERSPEECH*.