INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

On the Robustness of Arabic Speech Dialect Identification

Peter Sullivan[1], *AbdelRahim Elmadany*[1], *Muhammad Abdul-Mageed*[1,2]

[1]The University of British Columbia, Canada
[2]Mohamed bin Zayed University of Artificial Intelligence, UAE
{prsull@student.,a.elmadany@,muhammad.mageed@}ubc.ca

Abstract

Arabic dialect identification (ADI) tools are an important part of the large-scale data collection pipelines necessary for training speech recognition models. As these pipelines require application of ADI tools to potentially out-of-domain data, we aim to investigate how vulnerable the tools may be to this domain shift. With self-supervised learning (SSL) models as a starting point, we evaluate transfer learning and direct classification from SSL features. We undertake our evaluation under rich conditions, with a goal to develop ADI systems from pretrained models and ultimately evaluate performance on newly collected data. In order to understand what factors contribute to model decisions, we carry out a careful human study of a subset of our data. Our analysis confirms that domain shift is a major challenge for ADI models. We also find that while self-training does alleviate this challenges, it may be insufficient for realistic conditions.

Index Terms: Arabic speech processing, language identification, domain shift, dialect identification, Arabic language processing.

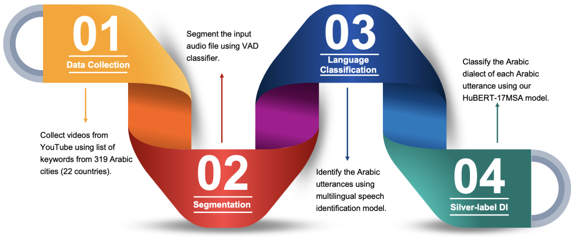
1. Introduction

Arabic faces significant challenges from a spoken language processing perspective. Mixing of dialectal Arabic (DA) and Modern Standard Arabic (MSA) in everyday speech [1], means that performance of MSA-trained ASR models in realistic settings is deemed to be limited by weak performance of these models on DA. Similarly, purely DA ASR models are hampered by availability of only limited resources and lack of dialect identification (DID) tools that can aid creation of new dialectal datasets. Another challenge that impacts all language identification (LID) and DID tasks is the problem of performance on data from domains unseen during training. For the latter challenge, unsupervised domain adaptation methods have been proposed [2, 3].

One method that could potentially quickly boost DID models in absence of limited labeled data is speech self-supervised learning (SSL) [4, 5]. It is not sufficiently clear, however, how exactly SSL can fare on realistic and out-of-domain settings. Similarly, the SSL representations potentially provide an alternative to x-vector segment representations, without overfitting to target domains. This is the case since output layers of models such as HuBERT have strong phonetic encoding [6], which could simulate earlier phonotactic approaches to language classification [7, 8].

In this work, we tackle the problem of robustness of Arabic speech DID, making the following contributions: (1) we quantify the performance of transfer learning from SSL, starting models under two settings: purely finetuned and self-trained; (2) we investigate use of SSL representations as baseline alternatives to i/x-vectors; (3) we assess performance on newly collected data to probe the limits of our transfer learning methods in a realistic data pipeline.

Figure 1: *Extracting Arabic utterances from the YouTube City data pipeline.*


01
Data Collection
Collect videos from YouTube using list of keywords from 319 Arabic cities (22 countries).
02
Segmentation
Segment the input audio file using VAD classifier.
03
Language Classification
Identify the Arabic utterances using multilingual speech identification model.
04
Silver-label DI
Classify the Arabic dialect of each Arabic utterance using our HuBERT-17MSA model.

2. Related work

Several recent works considered ADI [9, 10, 11, 12, 13]. As early as the 2015 NIST's Language Recognition Evaluation (LRE) [14, 11], the growing importance of neural architectures became apparent [15]. These include deep bottleneck features (DBN) [15, 16], DNNs for classification [16, 17], i-vector post processing [15], as well as summarization [16]. The 2017 challenge [11] further demonstrated the importance of neural embedding x-vectors for the classification task [18], marking the dominance of neural architectures over traditional, GMM-based approaches.

The ADI-5 challenge, a coarse grain dialect and MSA classification task [9], similarly illustrates this transition. The winning system in the challenge used a siamese-network of CNNs to further post-process i-vectors [19], and the second best system used a generative adversarial network in addition to a traditional Gaussian classifier for classification [20]. This trend towards more neural architectures continues with ADI-17, where neural end-to-end (E2E) architectures outperformed conventional i-vector approaches [10] with the best competition performance being an E2E ResNet architecture that employed novel pooling layers [21] Deep learning methods used on these datasets include CNNs [12, 13] as well as transformers [13].

SSL models, yet unexplored in ADI context, consist of a self-supervised pre-training objective such as a contrastive task [5], target prediction [4], or an autoregressive prediction [22]. These models learn strong representations of linguistic features [6], which allow them to be finetuned on a variety of speech processing tasks. SSL models have already shown strong performance in LID [23], which suggests a strong potential of these models for DID. To the best of our knowledge, however, no studies have explored how best to use SSL models

10.21437/Interspeech.2023-1005

on ADI especially under potential domain shift conditions.

3. Datasets and Preprocessing

3.1. Datasets

In this work, we use three public datasets (ADI-5, ADI-17, MGB-2) alongside two datasets we newly collect (YouTube City, and YouTube Dramas). We introduce these next.

ADI-5: A broadcast news dialectal identification corpus that is part of the MGB-3 Challenge [9]. It consists of $\sim$ 74 hours of audio segments labeled with broad categories of spoken dialects from the set { *MSA, Gulf, Levantine, North African, Egyptian*}.

Table 1: *The distribution of datasets in hour*

Dataset	Dialects	TRAIN	DEV	TEST
ADI-5	5	53.6	10	10.1
ADI-17	17	3,033.4	24.9	33.1
ADI-17 + MSA	18	4,233.4	26.9	35
YouTube City	17	3,539	—	—
YouTube Dramas	7	—	—	24.8

ADI-17: Created as part of the MGB challenges (MGB-5) [24], and consists of $\sim$ 3,000 hours of audio segments from YouTube programs covering a variety of different genres. The segments are split into 17 different dialectal categories, allowing for much finer grain dialectal analysis than the ADI-5 corpus (although lacking MSA as a category).

ADI-17 + MSA: to partially alleviate the lack of MSA in ADI-17, we create a new dataset by supplementing ADI-17 training data with MGB-2 training split (which is mainly MSA [25]) and the MSA part of ADI-5 training split. For the development dataset, we concatenate ADI-17's development set with the MSA portion of ADI-5's development set.

YouTube City: we randomly pick up 91,365 videos ($\sim$ 8,746 hours) from an in-house massive audio dataset collected from YouTube using a list of keywords that covers 319 Arabic cities from 22 countries. It consists of 238.2K videos ($\sim$ 62k hours). We apply a predefined preprocessing pipeline on the selected data to extract the Arabic utterances (Section 3.2 for more details).

YouTube Dramas: we manually collect Arabic dialectical drama series from YouTube that cover seven dialects: *Algerian*, *Egyptian*, *Emirati*, *Jordanian*, *Moroccan*, *Palestinian*, and *Yemeni*. Similar to the YouTube City data, we apply a predefined preprocessing pipeline to extract the Arabic utterances. The final YouTube Dramas dataset totals 24.8 hours extracted from three series for each dialect.

We ensure no overlap between ADI-17, the YouTube City, and YouTube Dramas datasets. Table 1 shows the distribution of the datasets.

3.2. Preprocessing Pipeline

As shown in Figure 1, the preprocessing pipeline for extracting the Arabic utterances from YouTube City collected data relies on the following four steps:

(1) Data Collection. Inspired by [26], we create a set of search keywords using a list of cities concatenated with country name, and use this to select videos. As stated earlier, the collected data consists of 238.2K videos (62k hours). For the self-training in this paper, we randomly select a maximum of 6,500 videos for each of the countries in ADI-17, leaving us with 91.4K videos (8.8K hours).

(2) Segmentation. We use the voice activity detection model *pyannote.audio*[1] [27, 28] to segment the audio files, keeping only utterances with length more than 3 seconds. The model give us 2M utterances from the 91.4K files.

(3) Language Identification. We use the multilingual speech identification model *lang-id-voxlingua107-ecapa*[2] [29, 30] to extract Arabic utterances only from the data. The model is able to detect 1.7M Arabic utterances from the 2M utterances.

(4) Silver-Label Dialect Identification. We create predictions for each utterance of the Arabic utterances, sorting theses into four groups for self-training: *Surrogate labelled*, labeled using the country of origin; *High*, *Medium*, and *Low* confidence, acquired by splitting the data based on the prediction confidence from HuBERT-17MSA classifier (see Section 4.1). The 442.6K utterances (26.58%) of the Arabic data are considered 'correct' (with weak labels) regarding the matching between the original country name and the model prediction label. (Section 4.3 for more details).

4. Methods

4.1. Finetuned SSL Models

To evaluate SSL models, we choose **(i)** the massively multilingual model XLS-R 300m [23] as our example of the wav2vec 2.0 model, and pick **(ii)** HuBERT-large-ll60k [4] as a contrasting monolingual English HuBERT model. Evidence from SUPERB indicates that while HuBERT may not be trained on Arabic, it has strong phonetic modeling capabilities [6].

We aim to deploy these pretrained models as dialect ID systems by adding modified TAP layers [31]. We first add a projection layer to reduce the dimensionality of our output, then mean pool the projected outputs over the temporal dimension before we pass this through a final classification layer.

Because these pretrained models are designed for use primarily in ASR, the finetuning process must be adjusted for DID. We do so by exploring the configuration space through a random search [32], which enables for an efficient search due to the likelihood of many hyperparameters not mattering significantly. Using a fixed compute budget for our two fine-tuned models of one node month of compute time (30 iterations) each, we search the following training hyperparameters: *Batch Size*, the number of samples per each gradient update; *Freeze Steps*, the number of updates with only the final prediction layer thawed; *Learning Rate*, a variable multiplying the size of the gradient update; *Max Steps*, the total number of steps in our training; *Sample Duration*, the duration of the random sample taken from the source audio; *Thaw Depth*, which layers of the model to thaw during fine-tuning. Additionally, we experiment with applying LayerNorm and Attention (LNA) fine-tuning, which is a fine-tuning technique that freezes all layers except for layer norm and multihead self-attention [33]. We summarize the ranges and distributions used for this finetuning exploration in Table 4. We adopt the original wav2vec 2.0 tri-state learning rate schedule, which consists of breaking the total training steps into a 10% ramp up to max learning rate, 50% plateu, and 40% cooldown period.

We directly fine-tune the xls-r-300m and HuBERT-large-ll60k models on our ADI-17 training set, to create fine-grained ADI models *xls-r-300m-17* and *HuBERT-17*. Since we realistically also want to differentiate between MSA and DA, we train a HuBERT model on the ADI-17MSA dataset to create *HuBERT-17MSA*.

[1] https://huggingface.co/pyannote/voice-activity-detection

[2] https://huggingface.co/speechbrain/lang-id-voxlingua107-ecapa

4.2. SSL Features

While both HuBERT and wav2vec 2.0 create outputs that can act as strong representation of the audio input, DID tasks can be prone to models overfitting to channel information or other aspects of the audio that are non-linguistically relevant [34]. This motivates us to acquire representations using the psuedo-labels from the k-means clustering of HuBERT output features.

We select representations from the Base model layer with the best Phone-Purity and PNMI, which occurs as output from layer 10 [4] [3]. Our process for generating the pseudo-labels is as follows: first, we exclude very short and very long utterances by selecting ADI-17 train files that are between 5 and 30 seconds long. We then pass these raw files through the HuBERT model, extracting the feature representations from layer 10. Because what level of phonetic information captured is related to the number of clusters, we treat this as a hyperparameter in our experiment: using a subsample of 10% of the resulting frames, we train 5 different k-means models using a choice from the set $\{200, 400, 600, 800, 1000\}$. Then, we assign cluster labels to the extracted features for our data. Because HuBERT covers 320 samples of audio per each prediction frame, for 16khz sample rate audio this leaves us with sequences of between 250 and 1500 labels. We compress these sequences into a length normalized representation essentially corresponding to the unigram count of each label divided by the total length of the sequence.

For classification, we pass the length normalized representations through a single layer neural network, training with cross-entropy loss.

Table 2: *Overview of hyperparameter search space for our pseudo-transcript classification models. Optimal results, batch:* 256*; clusters:* 1000*; learning rate:* 1×10^{-2}.

Hyperparameter	Range
Batch Size	$[64, 128, 256, 512]$
k-means Clusters	$[200, 400, 600, 800, 1000]$
Learning Rate	$[1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}]$

4.3. Self-Training

For self-training, we use silver labeled data from our YouTube City dataset. We group the newly collected data into several clusters based on (1) YouTube channel country information and (2) label confidence. We hypothesize that channels which are (in)consistently classified as being from the Gulf region are likely sources of MSA audio in contexts unfamiliar to the model. Similarly, we split up the silver labeled data into *low*, *medium*, and *high* confidence predictions. These categories are then concatenated to the ADI17+MSA dataset independently.

Table 3: *Breakdown of how the YouTube City data was split for self-training. Confidence intervals were based on 33 percentile intervals of the HuBERT-17 predictions.*

Self training settings	Description
Surrogate label	Label via country of origin
Low confidence	$< 54.24\%$ confidence
Medium confidence	$54.24\% \leq x < 87.84\%$ confidence
High confidence	$> 87.84\%$ confidence

Table 4: *The search space for our random search of finetuning hyperparameters as well as optimal configurations on ADI-17 Dev. We set the batch size such that a total of* 75 *seconds of audio can comfortably fit onto each V100 GPU, of which we train with* 4 *at a time. All values are picked from uniform distributions except for the learning rate (log uniform).*

	Range	HuBERT	xls-r
Batch Size	$4 \times \lfloor \frac{75}{Duration} \rfloor$	16	9
Freeze Steps	$[0, 1000]$	192	960
Learning Rate	$[1 \times 10^{-5}, 1 \times 10^{-2}]$	6×10^{-4}	1.2×10^{-3}
LNA	[True, False]	False	True
Max Steps	$[20k, 40k]$	29225	35956
Duration	$[4, 18]$ seconds	4.69	8.33
Thaw Depth	$[0, 23]$	3	4

5. Experiments

ADI-17. We evaluate our xls-r-300m-17, HuBERT-17, and HuBERT features models on the ADI-17 Test set to understand performance of these methods on in-domain classification.

ADI-17 Transfer to ADI-5. By pooling the country labels into the corresponding coarse region labels of ADI-5,[4] we test the out-of-domain performance of our models trained on ADI-17 on the ADI-5 Test set.

Domain Shift. We similarly, examine whether self-training on large diverse audio aids performance on a different domain distribution in comparison to models simply finetuned on ADI-17. We use our YouTube Dramas dataset as our Test set. This set consists only of drama series audio, unlike the multi-genre ADI-17 and YouTube City corpora, and may reflect a significant style of data the model is tailored for. For training, we use our existing HuBERT-17MSA checkpoint as well as the HuBERT-large-ll60k checkpoint as starting points, and use the identified finetuning hyperparameters from our HuBERT-17 training. We report results in macro-F_1 over the target classes.

Human Analysis. To understand the quality of the self-training labels, we randomly pick 25 utterances from the set of utterances where HuBERT-17MSA prediction and identified origin country do not match. We do this for four of the dialects: Emirati, Jordanian, Moroccan, and Sudanese. Then, a native speaker expert manually listens to and annotate the selected utterances. We check whether the utterance belongs to the identified country, whether it is DA or MSA, and whether or not the HuBERT-17MSA prediction is correct.

6. Results

ADI-17 Results. Compared to xls-r-300m, under our limited tuning HuBERT does surprisingly well. This is the case despite the model having no exposure to Arabic during pre-training (see Table 5). This performance can potentially be explained by a HuBERT better phonetic modelling compared to wav2vec 2.0 based training [6]. Using SSL models as feature extractors for simple classification appears to work as a reasonable baseline. Although not as competitive as even i-vector or x-vector systems, the simplicity of the approach could strike a balance between capturing the strength of SSL models and the potential for interpretability.

ADI-17 Transfer to ADI-5. Excluding MSA predictions, our HuBERT model finetuned on ADI-17 performs well. This is

[3] While the main HuBERT paper uses layer 9 for training of the HuBERT Large and XL, their graphs indicate potentially better attributes from using layer 10 with respect to PNMI and Phone-Purity [4]

[4] Gulf: KSA, UAE, OMA, IRQ, KUW, YEM, QAT; Levantine: LEB, PAL, JOR, SYR; Egypt: EGY, SUD; North Africa: MOR, MAU, LIB, ALG

Table 5: *Models' results on ADI-17's development and test datasets in term of macro F_1 score. †Our runs. *as reported in [24] and ‡as reported in [10], we do not have dev F_1 scores for these runs.* ***Bold:*** *the best results show in boldface.*

Type	Model	ADI-17 Dev	ADI-17 Test
Fixed	Majority Class	1.33	0.67
	HuBERT Features†	52.07	51.36
	i-Vector*	—	60.60
	x-Vector*	—	72.40
E2E	ResNet*	—	82.69
	DKU ResNet‡	97.4	**94.9**
	HuBERT-17†	92.23	92.12
	xls-r-300m-17†	90.77	90.20

Table 6: *Results of ADI-17 trained models ADI-5 Test (minus MSA segments). We do not compare prior submissions to the ADI-5 challenge as these predict on the full set. $^\star F_1$ 45.01 on full ADI-5 with MSA*

Model	F_1
Majority Class	10.92
HuBERT Features	67.71
HuBERT-17	**80.36**
XLS-R-300m E2E	76.20
HuBERT-17MSA*	37.17

the case despite a significant drop in performance (see Table 6), likely due to difference in channel and domain (e.g., topics covered and their distribution) of the ADI-5 dataset in comparison to the YouTube data for ADI-17. This hints at the potential problems in applying ADI models in the wild. Our SSL features method appears to be fairly robust to these same domain differences, and improves in relative performance on the ADI-5 (minus MSA) set.

Training on MSA appears to have a large detriment to performance on the DA portions of the ADI-5 dataset. Since this model has seen the MSA parts of the ADI-5 train set, it is likely that is has used channel differences between the MSA (broadcast television) and ADI-17 DA (YouTube) as a shortcut for classification. This illustrates the need for MSA that matches both domain and recording characteristics of DA, a potentially challenging issue due to sociolinguistic usage differences between MSA and DA.

Domain Shift Performance. The YouTube Dramas dataset appears to be a difficult domain for all the models (see Table 7), with models that have only been finetuned appearing to fair the worst. Self-training, in comparison, alleviates some of these issues. Considering performing self-training from scratch vs. an existing finetuned checkpoint, we observe a benefit of training from scratch. However, the finetuned models are able to achieve nearly the same results by training for only a few steps with the augmented dataset (500-1000 steps vs. 28500-29000 steps) and this short-training phenomenon likely needs further exploration.

Table 7: *Results on ADI17+MSA's development dataset and YouTube Dramas' test dataset in the term of macro F_1 score over target classes. Setting refers to the self-training starting checkpoint: HuBERT-large-ll60k (Vanilla) or our fine-tuned HuBERT-17MSA (17MSA). *Self-training with the surrogate labels from scratch failed to converge.*

Setting	Model	ADI17+MSA Dev ↑ (18 classes)	Dramas Test ↑ (7 classes)
	HuBERT-17	-	0.91
	HuBERT-17MSA	**89.39**	5.09
17MSA	HuBERT-YT$_s$	88.41	43.12
	HuBERT-YT$_h$	87.76	40.01
	HuBERT-YT$_m$	88.82	39.40
	HuBERT-YT$_l$	88.94	43.19
Vanilla	HuBERT-YT$_s$	$\star$	$\star$
	HuBERT-YT$_h$	88.89	**45.10**
	HuBERT-YT$_m$	88.51	42.89
	HuBERT-YT$_l$	88.62	43.20

Human Analysis. We find that 74.71% of the surrogate labels do not belong to the country identified through the collection process, but that of these mislabeled utterances the model is able to detect the correct dialect of 11.49%. In addition, the model is not able to recognize 85.87% of the MSA utterances, reflecting the fact that MSA on different topics or recording conditions from the training data may be difficult to detect.

7. Discussion

Quantized features from SSL models appear to be sufficient for LID and DID. These, however, could potentially be improved by either n-gram representations of features or different classification methods (e.g. CNN) that can capture a larger temporal window.

For them to be realistic and fit for real-world use, LID and DID benchmarks should consider either zero-shot or few-shot out-of-domain classification tasks. Similarly, the performance gap in detecting MSA utterances indicates the need for diverse MSA training corpora that cover both diverse recording conditions as well as a wide host of topics.

In addition, while using a small hyperparameter search space appears to be sufficient for reasonable performance for fine-tuning DID models from pretrained SSL models, there is likely room for further optimization by expanding the number of iterations as well from increasing the maximum training time.

YouTube audio may reflect vastly different recording and acoustic conditions, which we assume could be random in nature at times. The exact distribution of these characteristics could be further explored with respect to type of content and country of origin.

8. Conclusions

We presented experiments quantifying out-of-domain (both channel differences and topic shift) performance of Arabic DID models trained using self-training, finetuning, and a fixed representation approach. Our results emphasize the difficulties of domain shift for DID models and the importance of evaluating models on realistic settings.

We identify a number of directions for future work, including exploration of accented DID which would allow a better understanding of how these models may rely on phonotactic elements; evaluation of few shots settings for out-of-domain DID; and improvements to SSL representation models. Concretely, we also identify a strong need for diverse (formality, genre, channel, accent) MSA datasets.

9. References

[1] S. A. Chowdhury, Y. Samih, M. Eldesouki, and A. Ali, "Effects of dialectal code-switching on speech modules: A study using Egyptian Arabic broadcast speech." in *INTERSPEECH*, 2020, pp. 2382–2386.

[2] R. Duroselle, D. Jouvet, and I. Illina, "Unsupervised regularization of the embedding extractor for robust language identification," in *Odyssey 2020-The Speaker and Language Recognition Workshop*, 2020.

[3] S. Nercessian, P. Torres-Carrasquillo, and G. Martinez-Montes, "Approaches for language identification in mismatched environments," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 335–340.

[4] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[5] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[6] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[7] E. Singer, P. Torres-Carrasquillo, D. A. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The MITLL NIST LRE 2011 language recognition system," in *Odyssey 2012-The Speaker and Language Recognition Workshop*. Citeseer, 2012.

[8] A. Abad, "The L2F language recognition system for NIST LRE 2011," in *The NIST Language Recognition Evaluation Workshop*, 2011.

[9] A. Ali, S. Vogel, and S. Renals, "Speech recognition challenge in the wild: Arabic MGB-3," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 316–322.

[10] A. Ali, S. Shon, Y. Samih, H. Mubarak, A. Abdelali, J. Glass, S. Renals, and K. Choukri, "The MGB-5 challenge: Recognition and dialect identification of dialectal Arabic speech," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 1026–1033.

[11] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, J. Hernandez-Cordero *et al.*, "The 2017 NIST language recognition evaluation." in *Odyssey*, 2018, pp. 82–89.

[12] S. A. Chowdhury, A. M. Ali, S. Shon, and J. R. Glass, "What does an end-to-end dialect identification model learn about non-dialectal information?" in *INTERSPEECH*, 2020, pp. 462–466.

[13] W. Lin, M. Madhavi, R. K. Das, and H. Li, "Transformer-based arabic dialect identification," in *2020 International Conference on Asian Language Processing (IALP)*. IEEE, 2020, pp. 192–196.

[14] A. F. Martin, C. S. Greenberg, J. M. Howard, D. Bansé, G. R. Doddington, J. Hernández-Cordero, and L. P. Mason, "NIST language recognition evaluation—plans for 2015," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] K. A. Lee, H. Li, L. Deng, V. Hautamäki, W. Rao, X. Xiao, A. Larcher, H. Sun, T. Nguyen, G. Wang *et al.*, "The 2015 NIST language recognition evaluation: the shared view of i2r, fantastic4 and singams," in *INTERSPEECH 2016*, vol. 2016, 2016, pp. 3211–3215.

[16] O. Plchot, P. Matejka, O. Glembek, R. Fer, O. Novotnỳ, J. Pesan, L. Burget, N. Brummer, and S. Cumani, "BAT system description for NIST LRE 2015." in *Odyssey*, 2016, pp. 166–173.

[17] C. Yu, C. Zhang, S. Ranjan, Q. Zhang, A. Misra, F. Kelly, and J. H. Hansen, "UTD-CRSS system for the NIST 2015 language recognition i-vector machine learning challenge," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5835–5839.

[18] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors." in *Odyssey*, 2018, pp. 105–111.

[19] S. Shon, A. Ali, and J. Glass, "MIT-QCRI Arabic dialect identification system for the 2017 multi-genre broadcast challenge," in *2017 ieee automatic speech recognition and understanding workshop (asru)*. IEEE, 2017, pp. 374–380.

[20] A. E. Bulut, Q. Zhang, C. Zhang, F. Bahmaninezhad, and J. H. Hansen, "UTD-CRSS submission for MGB-3 Arabic dialect identification: Front-end and back-end advancements on broadcast speech," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 360–367.

[21] W. Cai, Z. Cai, W. Liu, X. Wang, and M. Li, "Insights in-to-end learning scheme for language identification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5209–5213.

[22] Y.-A. Chung and J. Glass, "Improved speech representations with multi-target autoregressive predictive coding," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2353–2358.

[23] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino *et al.*, "XLS-r: Self-supervised cross-lingual speech representation learning at scale," *arXiv preprint arXiv:2111.09296*, 2021.

[24] S. Shon, A. Ali, Y. Samih, H. Mubarak, and J. Glass, "ADI17: A fine-grained Arabic dialect identification dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8244–8248.

[25] A. Ali, P. Bell, J. Glass, Y. Messaoui, H. Mubarak, S. Renals, and Y. Zhang, "The MGB-2 challenge: Arabic multi-dialect broadcast media recognition," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 279–284.

[26] M. Abdul-Mageed, C. Zhang, A. Elmadany, and L. Ungar, "Toward micro-dialect identification in diaglossic and code-switched environments," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 5855–5876. [Online]. Available: https://aclanthology.org/2020.emnlp-main.472

[27] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, May 2020.

[28] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech 2021*, Brno, Czech Republic, August 2021.

[29] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[30] J. Valk and T. Alumäe, "VoxLingua107: a dataset for spoken language recognition," in *Proc. IEEE SLT Workshop*, 2021.

[31] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 74–81.

[32] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[33] X. Li, C. Wang, Y. Tang, C. Tran, Y. Tang, J. Pino, A. Baevski, A. Conneau, and M. Auli, "Multilingual speech translation from efficient finetuning of pretrained models," 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)*, 2021, pp. 827–838.

[34] H. Bořil, A. Sangwan, and J. H. Hansen, "Arabic dialect identification-" is the secret in the silence?" and other observations," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.