*INTERSPEECH 2021*
30 August – 3 September, 2021, Brno, Czechia

# SpeakerStew: Scaling to Many Languages with a Triaged Multilingual Text-Dependent and Text-Independent Speaker Verification System

*Roza Chojnacka*, Jason Pelecanos*, Quan Wang, Ignacio Lopez Moreno*

Google LLC, USA
{roza,pelecanos,quanw,elnota}@google.com

## Abstract

In this paper, we describe *SpeakerStew* – a hybrid system to perform speaker verification on 46 languages. Two core ideas were explored in this system: (1) Pooling training data of different languages together for multilingual generalization and reducing development cycles; (2) A novel triage mechanism between text-dependent and text-independent models to reduce runtime cost and expected latency. To the best of our knowledge, this is the first study of speaker verification systems at the scale of 46 languages. The problem is framed from the perspective of using a smart speaker device with interactions consisting of a wake-up keyword (text-dependent) followed by a speech query (text-independent). Experimental evidence suggests that training on multiple languages can generalize to unseen varieties while maintaining performance on seen varieties. We also found that it can reduce computational requirements for training models by an order of magnitude. Furthermore, during model inference on English data, we observe that leveraging a triage framework can reduce the number of calls to the more computationally expensive text-independent system by 73% (and reduce latency by 59%) while maintaining an EER no worse than the text-independent setup.

**Index Terms**: SpeakerStew, speaker recognition, multilingual, cross-lingual, hybrid system, triage

## 1. Introduction

Speech applications are becoming more and more pervasive with their popularisation in cell phones, cars and smart speakers. With increased adoption there is a demand for speech applications in more languages. This introduces challenges such as creating speech resources for the unsupported languages, building new models for the added languages and maintaining them while in production. This entails significant human and computational effort. Many speech tasks naturally call for a single model to represent all languages.

We explore this problem from the perspective of a speaker verification task on smart speaker devices, although similar principles could be applied to other select scenarios. For a new language to be added, speech data including audio and speaker labels from the language of interest must be available. A language specific model is then trained on this dataset and operating thresholds are determined for the end application considered. For the dataset studied in this paper, there are 46 languages of interest. Deploying models across all 46 languages involves significant effort. This becomes more challenging when we need to maintain a lightweight text-dependent (TD) model and a larger text-independent (TI) model. In this kind of setup, a natural question comes up whether it's possible to perform well on seen languages and generalize well to unseen languages while reducing computational requirements.

Researchers have studied various speech data augmentation techniques to help speaker verification systems better generalize. Such approaches include speed perturbation [1], room simulation [2, 3, 4], SpecAugment [5, 6, 7], and speech synthesis [8]. However, these techniques are mostly known to make speaker verification more robust to noise, channel effects and related domain mismatch, instead of language mismatch.

Studies have covered language mismatched system training and cross-language speaker verification trials. NIST [9, 10], a standards organization, supported its first non-English evaluation in 2000 (AHUMADA corpus [11]) followed by its first cross-language trial tasks in 2005/2006. Lu [12] improved evaluation results on non-English trials of a model trained only on English data. This was achieved by removing language factors from a joint factor analysis model. There was also a study [13] to extract speaker features using deep learning. The CNN-TDNN model was trained on English data and evaluated on the Chinese and Uyghur languages. It outperformed the baseline i-vector model by a large margin, particularly for language mismatch between enrollment and test. Another paper [14] assessed a neural network based multilingual model trained on English and 4 Indian languages. More recently, [15] examined adversarial training techniques and [16] explored the use of language-dependent score normalization. In [17] the authors mentioned that listeners identify voices in their native language more accurately than voices in an unknown, foreign language.

There have been multiple works examining the combination of text-dependent and text-independent information. For the 2001 NIST text-independent task [10], Sturim [18] selected commonly spoken keywords to model separately and then combined the result with a text-independent system. In the commercial space, Telstra, KAZ and Nuance delivered a speaker verification solution to Centrelink, the Australian Government Social benefits arm. It involved combining the results of multiple systems based on text-dependent and text-independent components [19, 20].

Past papers have explored combining text-dependent and text-independent components generally using linear system combination to optimize for performance. In this paper we explore a novel triage mechanism for system combination such that we not only maintain close to optimal performance but can also reduce the overall computational burden and response time in returning the result. Additionally, we disseminate results studying how performance varies across high and low resource languages. Recently, [21] and [22] built a single acoustic model for multiple languages with the aim of improving automatic speech recognition (ASR) performance on low-resource languages. In our SpeakerStew system, we aim to improve speaker recognition performance such that it better generalizes across languages, and even languages without training data.

* Equal contribution.

http://dx.doi.org/10.21437/Interspeech.2021-646

The rest of this paper is organized as follows. Section 2 describes the system which includes the proposed triage mechanism, Section 3 details the experimental setup and corresponding results, and Section 4 wraps up with the conclusions.

## 2. System description

In this section we discuss the 3 main components of the system; the lightweight TD system, the larger TI system, and a triage mechanism to combine the two information sources. These systems are designed to process a wake-up keyword (such as *"Ok Google"*) followed by a query (for example, *"What is the weather tomorrow?"*).

### 2.1. TD system

The TD system has a small memory footprint (235k parameters). It is trained to perform verification for two sets of keywords: *"Ok Google"* and *"Hey Google"*. The same keywords are used across languages but pronunciation may be different. This system is based on the TD work in [23] and we discuss key details here.

Firstly, for the TD system to work, we need to extract the relevant segment of speech containing only the keyword. For this purpose we used a multilingual keyword detector trained on all available languages and more details are available here [24]. Once the segment containing the keyword is identified, this audio is parameterised into frames of 25ms width and 10ms shift. We extract 40-dimensional log Mel filter bank energy features for each frame and stack 2 subsequent frames together. To improve robustness of the model, we apply data augmentation techniques involving noise sources combined with room simulation effects [25, 3, 26]. The resulting features are globally normalized to improve neural network training.

Our neural network consists of 3 LSTM layers with projection and a linear transform layer. Each LSTM layer has 128 memory cells, and is followed by a projection layer of size 64 with a tanh activation [27]. The 64-dimensional linear transform layer is applied to the output of the last projected LSTM layer, which is followed by L2-normalization to produce the speaker embedding.

We use the cosine similarity to compare enrollment utterances with evaluation utterances. We note that each speaker can have multiple enrollment utterances. Enrollment embeddings (which are already L2 normalized) are averaged for each speaker and then L2 normalized again. We used Generalized end-to-end (GE2E) contrastive loss [23] for training.

### 2.2. TI system

In contrast to the TD system, the TI system involves a larger model with 1.3M parameters. It uses both the keyword and speech query audio to recognize the speaker. For this TI task, it is not only that the pronunciation is different but the language can also differ. The system is based on the TI model proposed in [23] and is the same as the TD system except that it is a larger model and the training criterion is different. For the TD setup we use 384 memory cells with a projection size of 128. The final output speaker embedding is also of 128 dimensions. The training criterion uses the GE2E eXtended Set (XS) softmax loss [28].

### 2.3. System triage

In this section we discuss the use of system triage as an approach for maintaining speaker recognition performance while reducing latency and computation. In this work, we use a small-footprint, text-dependent speaker recognition system and a larger text-independent system. It is proposed that when the TD system is relatively confident, only the score for the TD system is used. When the TD system is not confident, scores from both the TD and TI systems are combined. With this approach, computation can be reduced. Additionally, for confident TD decisions, this information can be passed to other components sooner without having to wait for the entire query to be spoken. For example, language models can benefit from knowing the speaker identity in advance and fetch contextual information such as contacts, playlists or search history for this speaker before processing the query [29]. Figure 1 illustrates the triage mechanism.

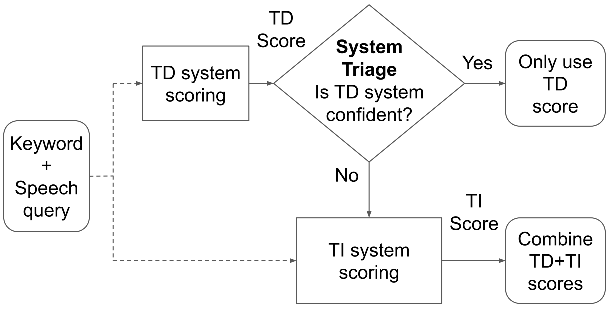

Figure 1: *Triage setup for speaker verification.*

## 3. Experimental results

In this section we introduce the experiment setup, the data used in training and evaluation, and experiments using monolingual and multilingual models followed by a system triage analysis.

### 3.1. Training and evaluation data

For training we use vendor collected speech data which contain recordings with *"OK Google"* and *"Hey Google"* keywords followed by a speech query. The data set covers 46 languages (63 dialects)[1]. The data are not evenly distributed across languages. For example, training data range from 39k utterances to 1.9M training utterances. Table 1 shows the number of utterances and speakers for selected languages in training and evaluation. The first group (de, en, es, fr, ja, ko, pt) represents *seen languages* which are used in the training of the *Multilingual-12* model used later. The second group contains *unseen languages* which are languages not used in the *Multilingual-12* model. Datasets of different languages are combined with the Multi-Reader approach during training [23]. Evidence from speech recognition also indicates combining more training data improves speech models in general [30].

### 3.2. Experimental setup

We trained 14 TD models and 14 TI models. For each of the TD/TI model sets, 12 are monolingual and trained only on data from the corresponding language, and the other two are multilingual models. The *Multilingual-46* model was trained on all

[1] A list of 46 languages: Arabic, Bulgarian, Bengali (India), Mandarin Chinese (Simplified), Mandarin Chinese (Traditional), Czech, Danish, German, Greek, English, Spanish, Finnish, Filipino, French, Galician, Gujarati (India), Hebrew, Hindi, Croatian, Hungarian, Indonesian, Italian, Japanese, Kannada (India), Korean, Lithuanian, Malayalam (India), Marathi (India), Malay, Norwegian, Dutch, Polish, Portuguese, Romanian, Russian, Slovak, Serbian, Swedish, Tamil (India), Telugu (India), Thai, Turkish, Ukrainian, Urdu (India), Vietnamese, Cantonese.

available data from 46 languages and the *Multilingual-12* model was trained on 12 languages[2]. We used the same structure and parameters for both the monolingual and multilingual models.

Table 1: *Number of speakers (Spk), utterances (Utt), same-speaker (Same) and different-speaker (Diff) trials. Data is partitioned into training and evaluation data across languages*[3]*. The first set of languages represents a sampling of 7 of the 12 languages used in training the Multilingual-12 model. The second set of languages represent a sampling of the remaining 34 languages not seen by the Multilingual-12 model. The last 2 rows show the total number of speakers and utterances used to train the Multilingual-12 and the Multilingual-46 models.*

| | **Training** | | **Evaluation** | | | |
|---|---|---|---|---|---|---|
| **Language** | **Spk** | **Utt** | **Spk** | **Utt** | **Same** | **Diff** |
| | **[k]** | **[k]** | **[k]** | **[k]** | **[k]** | **[k]** |
| German (de) | 3.4 | 908 | 1.5 | 255 | 192 | 188 |
| English (en) | 38.0 | 4531 | 0.2 | 13 | 12 | 189 |
| Spanish (es) | 18.0 | 1940 | 2.6 | 164 | 159 | 191 |
| French (fr) | 14.7 | 905 | 0.4 | 34 | 57 | 189 |
| Japanese (ja) | 14.3 | 1308 | 0.4 | 29 | 21 | 116 |
| Korean (ko) | 10.1 | 827 | 0.5 | 100 | 27 | 195 |
| Portuguese (pt) | 14.3 | 927 | 1.5 | 113 | 114 | 196 |
| Arabic (ar) | 4.6 | 463 | 1.5 | 115 | 102 | 174 |
| Mandarin (cmn) | 3.6 | 310 | 0.6 | 54 | 33 | 110 |
| Polish (pl) | 0.1 | 39 | 0.025 | 10 | 9 | 18 |
| Russian (ru) | 5.9 | 398 | 0.4 | 33 | 31 | 192 |
| Vietnamese (vi) | 2.8 | 224 | 0.3 | 41 | 39 | 185 |
| Malay (ms) | 2.1 | 168 | 0.4 | 33 | 26 | 150 |
| Multilingual-12 | 126.0 | 12619 | - | - | - | - |
| Multilingual-46 | 196.0 | 20618 | - | - | - | - |

### 3.3. Monolingual model performance

The evaluation results for different languages are shown in Table 2. They show the effectiveness of replacing one monolingual model with another. Unsurprisingly, the best results are observed for the cases where the training and evaluation languages are the same. Some models generalize better than others. For example, in the absence of the Spanish (es) model, the Portuguese (pt) model would be the next best option on Spanish language data. It is reassuring that they are related languages.

The other observation is that the TI models are significantly better than the TD models. One of the reasons is that the TI model has a larger capacity (1.3M parameters in TI vs 235k parameters in TD). Another reason is that the TI system can utilize more audio than the TD system. The TI system uses both the keyword and the query as evidence to determine similarity, whereas the TD system uses only the keyword.

### 3.4. Multilingual model performance

Table 3 compares the results across monolingual and multilingual trained systems. It is observed that the multilingual models can reach or even exceed the performance of monolingual models for both seen and unseen languages. It is observed that for languages with very few training speakers, a multilingual model can be of significant benefit. Polish is a language with only about 100 training data speakers. The performance on Polish can be greatly improved by using multilingual models over the monolingual counterparts.

Table 2: *Evaluation of monolingual models across languages. The columns represent different languages that were evaluated. The rows describe the type of model (TD/TI) and the language the model was trained on. The shaded cells represent models that have not seen training data with the same language as the evaluation data set. (Performance measured as % EER.)*

| **TD/TI** | **Performance on different languages** | | | | | | |
|---|---|---|---|---|---|---|---|
| **System** | **de** | **en** | **es** | **fr** | **ja** | **ko** | **pt** |
| TD-de | 1.27 | 5.54 | 1.86 | 3.04 | 4.32 | 3.61 | 3.09 |
| TI-de | 0.61 | 1.94 | 0.72 | 1.31 | 1.22 | 1.32 | 1.01 |
| TD-en | 2.04 | 2.64 | 1.66 | 2.05 | 3.14 | 3.05 | 1.81 |
| TI-en | 0.96 | 1.13 | 0.82 | 1.17 | 0.69 | 1.24 | 0.86 |
| TD-es | 1.80 | 3.20 | 0.64 | 1.88 | 2.83 | 2.69 | 1.50 |
| TI-es | 1.01 | 1.43 | 0.54 | 1.22 | 0.69 | 1.06 | 0.83 |
| TD-fr | 2.23 | 3.51 | 1.98 | 1.29 | 3.40 | 2.74 | 2.12 |
| TI-fr | 0.94 | 1.37 | 0.65 | 0.92 | 0.64 | 0.90 | 0.81 |
| TD-ja | 2.95 | 4.55 | 2.57 | 2.81 | 1.12 | 3.26 | 2.56 |
| TI-ja | 1.18 | 1.54 | 0.75 | 1.14 | 0.32 | 1.02 | 0.88 |
| TD-ko | 2.85 | 4.58 | 2.38 | 2.98 | 3.56 | 1.05 | 2.43 |
| TI-ko | 1.04 | 1.65 | 0.78 | 1.14 | 0.81 | 0.39 | 0.86 |
| TD-pt | 2.05 | 3.39 | 1.46 | 2.06 | 3.55 | 2.58 | 0.93 |
| TI-pt | 0.82 | 1.21 | 0.52 | 0.94 | 0.62 | 0.98 | 0.48 |

### 3.5. Linear score combination

It's beneficial to combine TD and TI models - these results are shown in the last set of rows of Table 3. The performance numbers shown here are based on finding an optimal linear combination weight (using a linear sweep) of the corresponding TD/TI systems for each combination result in the table.

An observation is that for the multilingual TD/TI linear score combination results, the *Comb-multilingual-12* (*12L*) system is comparable to the *Comb-multilingual-46* (*46L*) system for languages seen by the *12L* model. For languages not seen by the *12L* model, there is a slight performance improvement with the *46L* setup. For most practical purposes, the *12L* model is sufficient to capture the performance benefit without resorting to requiring data from 46 languages. Additionally, the *12L* model was not trained on tonal languages yet generalizes well to Mandarin (cmn) and Vietnamese (vi) which are tonal languages.

### 3.6. Triage score combination

In this section we examine how system triage can maintain speaker recognition performance while improving average decision response time (latency) and computational requirements.

In our triage implementation, we use the TD system score when the TD system is confident. When it is not confident we generate the TI system score and combine it with the TD system result. The two scores are combined based on using the same optimal combination weights for the corresponding language/model in Table 3. In determining whether the TD system is confident, we check whether the TD score is inside or outside a range defined by upper and lower score thresholds. If it is outside the selected score bounds, then the TD system is regarded as confident. Otherwise the TD system is not confident.

Figure 2 consists of two parts; (i) an EER heat map and (ii) a contour plot overlay showing the TI trigger rate (*i.e.* how often the TI system is needed). The heat map presents the EER as a function of the lower (x-axis) and upper (y-axis) cut-off thresholds of the TD system. The upper left region of the plot

[2] Multilingual-12 training data languages: Danish (da), German (de), English (en), Spanish (es), French (fr), Italian (it), Japanese (ja), Korean (ko), Norwegian (nb), Dutch (nl), Portuguese (pt), Swedish (sv).

[3] For English, training utilizes multiple varieties of English while evaluation is performed on United States (US) English only.

Table 3: *Table comparing the performance of (i) TD models (ii) TI models and (iii) the optimal linear score combination of the corresponding TD and TI systems. The columns represent the different languages that were evaluated. The left side of the table contains evaluation results for higher-resource languages that were used to train the multilingual-12 models. The right side represents lower-resource languages that the multilingual-12 models have not seen and the corresponding results are shaded to indicate the unseen language condition. The multilingual-46 models have seen all languages. Monolingual models were trained using data from one language only (the same as evaluation language). Performance measured as % EER.*

| | **Languages in *multilingual-12* model** | | | | | | | **Languages not in *multilingual-12* model** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **System** | **de** | **en** | **es** | **fr** | **ja** | **ko** | **pt** | **ar** | **cmn** | **pl** | **ru** | **vi** | **ms** |
| TD-monolingual | 1.27 | 2.64 | **0.64** | 1.29 | **1.12** | **1.05** | **0.93** | 2.26 | 1.96 | 17.89 | 2.10 | 3.15 | 3.72 |
| TD-multilingual-12 | **1.09** | **2.47** | 0.77 | 1.28 | 1.42 | 1.48 | 1.11 | 1.60 | 1.82 | **1.01** | 1.40 | 2.48 | 1.76 |
| TD-multilingual-46 | 1.16 | 2.53 | 0.80 | **1.24** | 1.64 | 1.62 | 1.14 | **1.53** | **1.27** | 1.03 | **1.34** | **1.93** | **1.59** |
| TI-monolingual | 0.61 | 1.13 | 0.54 | 0.92 | 0.32 | **0.39** | 0.48 | 1.05 | 0.60 | 11.78 | 0.87 | 1.39 | 1.58 |
| TI-multilingual-12 | **0.44** | 0.91 | 0.29 | **0.65** | **0.24** | 0.40 | **0.35** | 0.77 | 0.42 | 0.91 | **0.64** | 0.56 | **0.79** |
| TI-multilingual-46 | 0.49 | **0.84** | **0.24** | 0.66 | 0.28 | 0.48 | **0.35** | **0.65** | **0.38** | **0.50** | 0.66 | **0.51** | 0.80 |
| Comb-monolingual | 0.47 | 0.98 | 0.29 | 0.69 | 0.24 | **0.25** | 0.34 | 0.78 | 0.36 | 8.47 | 0.64 | 0.81 | 1.00 |
| Comb-multilingual-12 | **0.40** | 0.86 | **0.22** | **0.57** | **0.22** | 0.33 | **0.30** | 0.64 | 0.34 | 0.58 | **0.62** | 0.53 | **0.78** |
| Comb-multilingual-46 | 0.45 | **0.83** | **0.22** | 0.60 | 0.24 | 0.39 | 0.31 | **0.57** | **0.28** | **0.48** | 0.64 | **0.48** | **0.78** |

sustains better performance while moving away from this region increases the EER. This heat map needs to be considered along with the contour plot overlay which describes the percentage of trials that use both the TI and TD systems (*i.e.* the TI trigger rate). For the *multilingual-12* model, the English TD EER is 2.47% and the TI EER is 0.91% while the always combined result is 0.86% (Table  3). If we set lower and upper thresholds of 0.23 and 0.65 respectively, we can maintain TI EER performance while only needing the TI system for 27% of trials (Figure 2), a 73% reduction. In our application of interest, the average duration of the keyword portion of speech is 0.7s, and the query portion is 3s on average. At 27% TI trigger rate, assuming an immediate system response, the expected time to verification completion is reduced from 3.7s (the utterance duration) to 1.5s. This is a saving of 2.2s (59% reduction).

There are two factors to consider when setting the thresholds. The first is that the contour overlay and the suitable thresholds chosen are affected by the prior probabilities of the same-speaker and different-speaker classes. In Figure 2, there is a 50% same-speaker-trial probability. The second consideration is if this approach is to operate with the same thresholds across all languages. If so, it may require a larger range for the lower and upper cutoff thresholds and would reduce efficiency. Figure 3 examines this first factor by evaluating how performance can be affected by class probabilities. For clarity, four languages are shown. The dashed lines show the result for 50% same-speaker-trial probability while the shaded areas show the best/worst case scenarios by assessing performance at 0% and 100% same-speaker-trial probabilities. For this framework, the figure shows that worst case performance is not much worse than the equal class probability scenario.

## 4. Conclusions

Our experiments provide evidence toward the hypothesis that a speaker recognition model trained on data from different languages generalizes better to unseen languages than monolingual models. This applies to both text-dependent (where the keyword is pronounced differently in each language) and text-independent models. In scaling up to many languages, an opportunity is presented to improve efficiency [31]. During training, bundling languages into a single model can reduce computational requirements by an order of magnitude. At runtime, triage can be implemented to reduce the calls to a larger text-independent model without significantly degrading overall performance, thus reducing latency and computational cost.

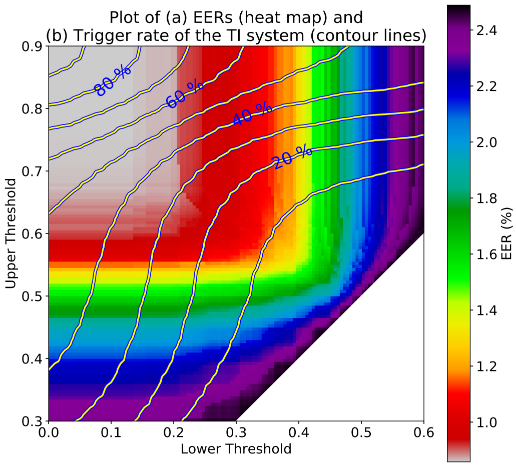

Figure 2: *Triage results showing the EER (heat map) and the TI trigger rate (contour lines) as a function of the TD upper and lower score thresholds for US English evaluation data and the TD/TI-multilingual-12 models. Same-speaker class probability is 50%.*

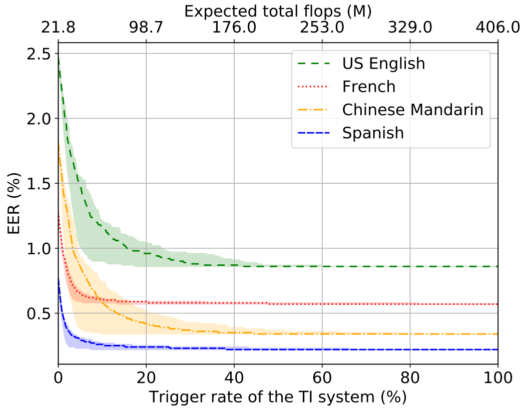

Figure 3: *Plot of EER as a function of the TI trigger rate as well as the expectation of the total number of floating point operations (flops) for the multilingual-12 system. For each language a dashed error curve is shown for the case with equal same/different speaker class prior probabilities. The corresponding shaded area reveals the EER bounds (i.e. best/worst case scenarios) determined when the same-speaker-trial class probability is either close to 0% or 100%.*

# 5. References

[1] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, “Audio augmentation for speech recognition,” in *Interspeech*, 2015.

[2] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, “Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home,” in *InterSpeech*, 2017.

[3] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, “A study on data augmentation of reverberant speech for robust speech recognition,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[5] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech*, 2019.

[6] M. Y. Faisal and S. Suyanto, “SpecAugment impact on automatic speaker verification system,” in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 2019, pp. 305–308.

[7] S. Wang, J. Rohdin, O. Plchot, L. Burget, K. Yu, and J. Černockỳ, “Investigation of SpecAugment for deep speaker embedding learning,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7139–7143.

[8] Y. Huang, Y. Chen, J. Pelecanos, and Q. Wang, “Synth2Aug: Cross-domain speaker recognition with TTS synthesized speech,” in *IEEE Spoken Language Technology Workshop (SLT)*, 2021.

[9] C. S. Greenberg, L. P. Mason, S. O. Sadjadi, and D. A. Reynolds, “Two decades of speaker recognition evaluation at the National Institute of Standards and Technology,” *Computer Speech and Language*, vol. 60, 2020.

[10] NIST, “Speaker and language recognition,” https://www.nist.gov/programs-projects/speaker-and-language-recognition, accessed: 2020-12-16.

[11] J. Ortega-Garcia, J. Gonzalez-Rodriguez, and V. Marrero-Aguiar, “AHUMADA: A large speech corpus in Spanish for speaker characterization and identification,” *Speech Communication*, vol. 31, pp. 255–264, 2000.

[12] L. Lu, Y. Dong, X. Zhao, J. Liu, and H. Wang, “The effect of language factors for robust speaker recognition,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 4217–4220.

[13] A. Misra and J. H. L. Hansen, “Spoken language mismatch in speaker verification: An investigation with NIST-SRE and CRSS Bi-Ling corpora,” in *IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 372–377.

[14] R. Kumar, R. Ranjan, S. Singh, R. Kala, and R. Tiwari, “Multilingual speaker recognition using neural network,” in *Proceedings of the Frontiers of Research on Speech and Music (FRSM)*, 2009, pp. 1–8.

[15] W. Xia, J. Huang, and J. H. Hansen, “Cross-lingual text-independent speaker verification using unsupervised adversarial discriminative domain adaptation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[16] J. Thienpondt, B. Desplanques, and K. Demuynck, “Cross-lingual speaker verification with domain-balanced hard prototype mining and language-dependent score normalization,” in *Interspeech*, 2020.

[17] T. K. Perrachione, “Speaker recognition across languages,” in *The Oxford Handbook of Voice Perception*. Oxford University Press, 2017. [Online]. Available: https://open.bu.edu/handle/2144/23877

[18] D. Sturim, D. Reynolds, R. Dunn, and T. Quatieri, “Speaker verification using text-constrained Gaussian mixture models,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 2002, pp. 677–680.

[19] D. Top, “Centrelink unveils voice authentication system,” https://opusresearch.net/wordpress/2009/05/28/centrelink-unveils-voice-authentication-system/, 2009, accessed: 2020-12-16.

[20] R. Summerfield, T. Dunstone, and C. Summerfield, “Speaker verification in a multi-vendor environment,” in *W3C workshop on speaker identification and verification (SIV)*, 2008.

[21] 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 *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[22] V. Pratap, A. Sriram, P. Tomasello, A. Hannun, V. Liptchinsky, G. Synnaeve, and R. Collobert, “Massively multilingual ASR: 50 languages, 1 model, 1 billion parameters,” in *Interspeech*, 2020.

[23] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[24] H. Park, P. Violette, and N. Subrahmanya, “Learning to detect keyword parts and whole by smoothed max pooling,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7899–7903.

[25] R. Lippmann, E. Martin, and D. Paul, “Multi-style training for robust isolated-word speech recognition,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 12, 1987, pp. 705–708.

[26] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, “Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home,” in *InterSpeech*, 2017, pp. 379–383.

[27] H. Sak, A. W. Senior, and F. Beaufays, “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,” *arXiv:1402.1128*, 2014. [Online]. Available: http://arxiv.org/abs/1402.1128

[28] J. Pelecanos, Q. Wang, and I. L. Moreno, “Dr-Vectors: Decision residual networks and an improved loss for speaker recognition,” in *Interspeech*, 2021.

[29] P. Aleksic, M. Ghodsi, A. Michaely, C. Allauzen, K. Hall, B. Roark, D. Rybach, and P. Moreno, “Bringing contextual information to Google speech recognition,” in *Interspeech*, 2015.

[30] W. Chan, D. Park, C. Lee, Y. Zhang, Q. Le, and M. Norouzi, “SpeechStew: Simply mix all available speech recognition data to train one large neural network,” *arXiv preprint arXiv:2104.02133*, 2021.

[31] E. Strubell, A. Ganesh, and A. McCallum, “Energy and policy considerations for modern deep learning research,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 09, pp. 13 693–13 696, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/7123