Automatic Spoken Language Identification using a Time-Delay Neural Network

Recognition Technologies, Inc.
Technical Report: RTI-20220519-01

DOI: 10.13140/RG.2.2.21631.89763

Benjamin Kepecs[1] *and Homayoon Beigi*[2]

[1]Columbia University
[2]Recognition Technologies, Inc. and Columbia University
[1]bek2127@columbia.edu, [2]beigi@recotechnologies.com

Abstract

Closed-set spoken language identification is the task of recognizing the language being spoken in a recorded audio clip from a set of known languages. In this study, a language identification system was built and trained to distinguish between Arabic, Spanish, French, and Turkish based on nothing more than recorded speech. A preexisting multilingual dataset was used to train a series of acoustic models based on the Tedlium TDNN model to perform automatic speech recognition. The system was provided with a custom multilingual language model and a specialized pronunciation lexicon with language names prepended to phones. The trained model was used to generate phone alignments to test data from all four languages, and languages were predicted based on a voting scheme choosing the most common language prepend in an utterance. Accuracy was measured by comparing predicted languages to known languages, and was determined to be very high in identifying Spanish and Arabic, and somewhat lower in identifying Turkish and French.

1. Introduction

Spoken language identification involves recognizing a language based on a snippet of recorded speech. There are multiple potential uses for this technology. Real-time automatic translation and transcription systems often require prior information about the input language in order to operate, and would not be suitable in an environment where multiple languages are spoken at unpredictable times, such as at a global conference. Automatic spoken language identification can be used at the beginning of a multilingual speech recognition pipeline to obviate the need for human input and make the software work more seamlessly. In particular, spoken language identification could play a role in a code-switching detection pipeline to keep track of when someone has begun to speak in a new language, and what language they are speaking in. Another potential application of this technology is in customer service; currently, to serve diverse global clients a customer service hotline may automatically list several languages and direct the caller to press a button corresponding to their language. Automatic spoken language identification could be used to detect a caller's language and automatically route their call to a native speaker or list additional options in their language.

2. Related Works

There have been many research studies exploring ways to implement spoken language identification using both neural networks and more traditional acoustic models. There are, in addition, several basic language identification utilities built into commercial products such as Google Cloud, Microsoft Azure, and AWS [1–3]. These technologies, however, are still in their early stages of application; automatic language identification was only added to AWS Transcribe in late 2020.

One of the first semi-successful approaches to language ID was to use a panel of Hidden Markov Models (HMMs), each trained on a single language [4]. Speech of some unknown language would then be decoded with each of the HMMs in turn, and the language of the model which decoded the speech most accurately was said to be the unknown language. This method was further refined by using separate stochastic models for each phoneme of the target languages and decoding speech with a distinct series of phoneme models for each language [5]. In some cases, these models took into account prosodic elements by manually adjusting features to incorporate prosodic characteristics of the languages [6, 7].

More recent works on automatic language ID use deep-learning based methods to train neural networks as acoustic models. One approach is to treat the problem of language identification as a computer vision classification problem, and thus to train a CNN on spectrograms labeled with their corresponding languages [8]. This approach, though it meets with some success in limited scenarios, must generate unnecessary intermediate images and disregards decades of progress on acoustic feature extraction and acoustic model generation.

Other state-of-the-art deep-learning approaches are more in line with traditional ASR and use convolutional neural networks (CNNs) or regular deep neural networks (DNNs) as acoustic models, accepting labeled i-vectors as features [9]. These methods achieve 3.48% and 3.55% equal error rates (EER), respectively, on language identification among 50 different languages. Surprisingly, integrating two different classifiers such as a CNN and a Support Vector Machine (SVM) by running them in parallel and then adding the resulting scores yields an even lower EER of 2.79%. Though Google, Inc. does not disclose the technology that drives it automatic language identification on Google Cloud, it is likely that a DNN-based method with i-vector features is used in their application as well [10].

3. Dataset

The training data used in this study was the MediaSpeech dataset [11], which contains approximately 10 hours of speech with matched transcriptions for Arabic, Spanish, French, and Turkish. This multilingual dataset was chosen so that the data for all four languages would have some degree of inter-language consistency in audio quality and file format. The data was primarily derived from European news channels, so it can be inferred that the French accents represented in the dataset were France-accented French as opposed to African-accented French. To test the system's performance, five additional datasets were used: a dataset of female speakers in Colombian Spanish [12], recited speech in Tunisian Modern Standard Arabic [13], a Turkish Daily Use Sentence dataset [14], the att-HACK French Expressive Speech Database with Social Attitudes [15], and finally a dataset of African accented French [16]. The first four datasets were used to test language ID performance on speech similarly accented to the speech found in the MediaSpeech dataset, and the final African accented French dataset was used to determine the robustness of the prediction on unfamiliar accents.

4. Methods

Different languages have distinct phonological compositions, vocabularies, and prosodies, and as such contain many plausible features that could be used to distinguish one from another. The approach presented in this study adapts the Kaldi Tedlium s5_r3 egs recipe for speech recognition [17, 18] to perform language identification. Traditional MFCC feature extraction [19] and acoustic modeling was followed with refined modeling via a time-delay neural network. The resulting trained network was used to decode testing data and determine its language. A flowchart displaying the high-level steps in this process is shown in Figure 1.

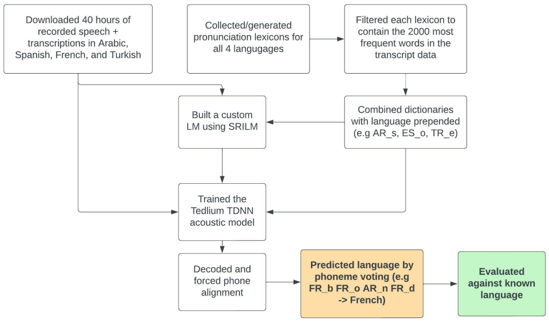

Downloaded 40 hours of recorded speech + transcriptions in Arabic, Spanish, French, and Turkish
Collected/generated pronunciation lexicons for all 4 langugages
Filtered each lexicon to contain the 2000 most frequent words in the transcript data
Built a custom LM using SRILM
Combined dictionaries with language prepended (e.g AR_s, ES_o, TR_e)
Trained the Tedlium TDNN acoustic model
Decoded and forced phone alignment
Predicted language by phoneme voting (e.g FR_b FR_o AR_n FR_d -> French)
Evaluated against known language

Figure 1: *High-level methodology for spoken language identification used in this study.*

```
a ES_a_7
a(2) FR_aa
abajo ES_a ES_b ES_a_7 ES_x ES_o
abdo ES_a_7 ES_b ES_d ES_o
abi TR_a: TR_b TR_i
abierta ES_a ES_b ES_i ES_e_7 ES_r( ES_t ES_a
aboutir FR_aa FR_bb FR_ou FR_tt FR_ii FR_rr
abril ES_a ES_b ES_r( ES_i_7 ES_l
```

Figure 2: *A small portion of the combined multilingual dictionary with language prepends and renumbering of different words with identical spellings.*

4.1. Data and Lexicon Preparation

The most recent version of Kaldi [20] was cloned from Github and compiled on a GCP VM instance. The MediaSpeech dataset was then downloaded from OpenSLR, and all audio files were converted to .wav from .flac formats. The data directory for Kaldi feature extraction was constructed according to Kaldi specification.

A pronunciation lexicon for each language in the dataset was either downloaded or manually generated: the French and Spanish pronunciation lexicons were sourced from CMUDict [21], the Turkish lexicon was directly downloaded from a private Github repository [22], and the Arabic lexicon was generated by feeding the combined corpus of Arabic utterances in the MediaSpeech dataset into a Python tool for generating Arabic pronunciation lexicons from a specified corpus [23]. The lexicon for each language was filtered to contain only (at most) the 2000 most frequently occurring words in the MediaSpeech transcriptions for that language. To adjust the lexicon for the specific task of language identification, the phones in each dictionary were prepended with a symbol indicating the language of which they were a part. For example, the 'e' phone in Spanish was prepended with 'ES', the 'aa' phone in French was prepended with 'FR', etc. The dictionaries were combined and words with duplicate spellings but different pronunciations among and within languages were appropriately renumbered, as shown in Figure 2.

Next, the transcriptions provided for each language in the MediaSpeech dataset were combined in random order into a single multilingual corpus. This corpus was fed into SRILM [24] with order 4 to generate a 4-gram multilingual language model in ARPA format. This was then converted into an OpenFST format.

4.2. Feature Extraction and Training

As in the Tedlium egs recipe, feature extraction was performed on the audio input to yield MFCC features for all recordings. 10,000 segments in the dataset were subsetted, and these were used to perform flat start monophone training. The monophones were aligned to transcriptions, and the results were used to train delta-based triphones. These too were aligned, and the results were used to bootstrap and train LDA-MLLT triphones. Finally, the results were aligned again and used to train SAT triphones.

The TDNN at the core of ASR acoustic modelling in the Tedlium recipe is trained with iVectors. To generate iVectors for the training data, the training data was speed-perturbed and volume perturbed and and FMLLR-aligned to generate low-resolution and high-resolution perturbed MFCC features. These components were used to train an iVector extractor for the perturbed training and native testing data. Alignment lattices and a decision tree were generated from the low-resolution MFCCs, and subsequently the MFCCs, decision tree, and extracted iVectors were used to train a 16-layer TDNN. The training was performed on a NVIDIA Tesla T4 GPU. A new graph was generated and the training data was decoded based on the high-resolution features computed previously.

4.3. Test Data Preparation

Rather than withhold a portion of the MediaSpeech training data as testing data, five additional datasets were used to construct a testing partition. Aside from the practical considerations governing this decision (namely, an oversight on behalf of the author), using other datasets to test the model provided a more robust evaluation of its performance in the real world because it introduced more variability into the data quality, recording apparatus, and speaker set. In addition to testing the performance of the model on identifying languages similarly accented to the training data, the robustness of the model in predicting the language of speech with an out-of-training accented was also evaluated. The Colombian Spanish, Tunisian Arabic, Expressive French, Daily Use Turkish, and African Accented French datasets were downloaded and cleaned to remove punctuation and capitalization. The data directories for each language were assembled according to Kaldi specifications as before, and then were subsequently combined into one multilingual testing data directory. High-resolution MFCC features were computed for the assembled testing data, and iVectors were extracted using the iVector extractor trained previously.

```
CLS_0001 1 0.000 0.090 SIL
CLS_0001 1 0.090 0.140 AR_h_B
CLS_0001 1 0.230 0.040 AR_l_I
CLS_0001 1 0.270 0.090 AR_l_E
CLS_0001 1 0.360 0.010 AR_s_B
CLS_0001 1 0.370 0.090 AR_r_I
CLS_0001 1 0.460 0.010 AR_f_E
CLS_0001 1 0.470 0.010 AR_m_B
CLS_0001 1 0.480 0.010 AR_l_I
CLS_0001 1 0.490 0.010 AR_aa_I
```

Figure 3: *A sample of the phone symbol predictions produce by the decoding. The first column indicates the utterance name, the third and fourth columns indicate the time range, and the fifth column indicates the predicted phone symbol. Confidence values are excluded.*

4.4. Decoding and Scoring

The combined testing data directory was decoded using the graph produced by the TDNN training stage. The decoding was used to generate phone aligned lattices, and finally a series of phone number predictions for each utterance organized by timestamp. Each phone number was converted to a phone symbol by referencing the canonical phone list used in training and decoding. A sample of the intermediate .ctm file containing these phone symbol predictions is shown in Figure 3.

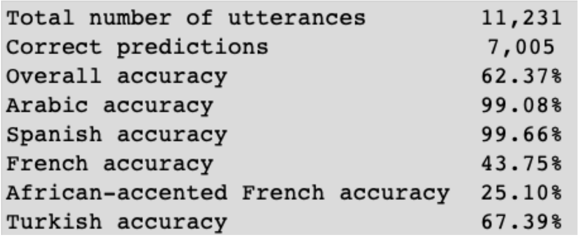
Finally, the aligned phone predictions were used to assign a hypothesized language ID to each utterance. The predicted phones for each utterance were parsed to extract their prepended language symbols, and a tally was maintained to keep track of the number of phones of a particular language were present in an utterance; e.g, the word ”basura” with phone predictions ES_b, ES_a, FR_s, FR_u, ES_r, AR_a would have a tally of ES: 3, FR: 2, AR: 1. Then, by a simple voting scheme, the utterance was predicted to be the language with the greatest tally. This prediction was compared to the known language of the utterance in each case, and the prediction accuracy rate was computed as the primary metric for assessing performance.

5. Results and Discussion

All data preparation, training, and decoding steps completed successfully. The accuracy rates for evaluation on the testing set are shown in Figure 4.

5.1. Performance Assessment

The model’s accuracy in predicting Arabic utterances to be Arabic and Spanish utterances to be Spanish was >99%, which is higher than the current state of the art. This high accuracy could be due to a number of factors, including the inherent robustness of the ASR system used in predicting the phones or because of high-quality audio

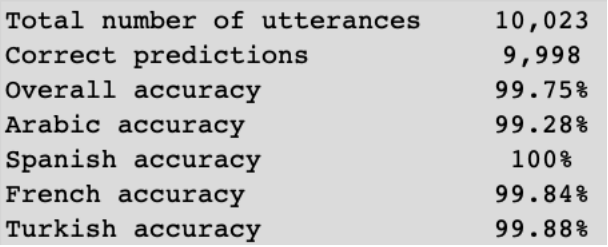


```
Total number of utterances          11,231
Correct predictions                  7,005
Overall accuracy                    62.37%
Arabic accuracy                     99.08%
Spanish accuracy                    99.66%
French accuracy                     43.75%
African-accented French accuracy    25.10%
Turkish accuracy                    67.39%
```

Figure 4: *Results of decoding and scoring on the full testing datasets for Arabic, Spanish, French, African-accented French, and Turkish. The overall accuracy was computed by simply dividing the number of correct predictions by the total number of utterances. The accuracy for each specific language was computed by dividing the number of correct predictions for that language by the total number of utterances in that language.*

and transcript data among the training and testing sets for these languages. For Arabic in particular, an additional reason for high accuracy could be that it is quite distant linguistically and acoustically from the other three languages, and therefore might be easier to identify from among these four.

The model's accuracy in identifying French and Turkish were significantly lower, though still greater than random choice for all but African-accented French. The accuracy on the Turkish testing dataset was about 70%; one source of error for this dataset in particular was that the source of the data (not OpenSLR) was quite obscure since there are very few speech corpora for Turkish, and the quality of the audio might not be optimal for testing. Alternatively, there might be an upstream issue with the pronunciation lexicon or language modeling for Turkish, leading to reduced downstream model performance. The accuracy of the model in identifying French was about 44%; this could also be due to the data source, which was part of the Expressive Speech dataset intended for emotion detection applications. It is possible that the emotional expression injected into the the audio could have interfered with identifying the language. Upon examining the .ctm file containing the phone predictions for the French utterances, it became apparent that almost all the phones in every utterance were predicted to be either French or Spanish, with Spanish in the slight majority. This is unlikely to be a coincidence; Spanish and French are closely related Romance languages, and it makes sense that the model might confuse the two with a bias toward predicting phones as Spanish. Interestingly, this examination reveals that the ASR system can pick up on linguistic similarities and relationships between different languages. One potential followup on this topic could be to build a similar ASR system that, given a large set of languages, can reconstruct the known language family tree (e.g tracing the development of various languages from the larger Indo-European family).



```
Total number of utterances     10,023
Correct predictions             9,998
Overall accuracy               99.75%
Arabic accuracy                99.28%
Spanish accuracy                100%
French accuracy                99.84%
Turkish accuracy               99.88%
```

Figure 5: *Results of decoding and scoring on the training set.*

The prediction accuracy of the model on African-accented French was about 25%, which indicates an essentially random prediction. This result is surprising; the phones in European French and African-accented French are not that substantially different [25], so a predictive model for one French should apply to the other. It could be that this predictive model is simply highly sensitive to accent because of over-training or another technical reason, or perhaps the phone differences are indeed substantial enough to confound the model.

5.2. Further Sources of Error

An additional factor that could be degrading the performance of this system is sub-optimal data preparation or network architecture. As part of the training stage of this system, the training data was decoded and scored for WER (word error rate; i.e, the error rate of the model in predicting the correct words in an utterance). At its lowest, the WER was 54.47%, which is significantly higher the the WER reported in the original Tedlium workflow. This could indicate a number of issues, including not enough training data being provided, improper processing of the data, inconsistent pronunciation lexicons, or an incorrect architecture for the neural network.

All of the enumerated potentially confounding factors in successful language identification could be the subject of future followups to this study in order to improve the system's performance. As an immediate followup, the training should be redone with only a portion of the MediaSpeech dataset, with some of the data being withheld for testing. The language identification accuracy of the model on the training data is >99% across all languages (Figure 5), which could mean that testing on different data from the same dataset would be more successful, if the model is not simply over-fitted.

One concern with the general methodology followed here to train a language ID model was that the model might learn to identify languages based on particular characteristics of the training audio rather than inherent differences in the languages. For example, if all of the speech data for French were collected with the same microphone and in the same strictly controlled environment, the model might recognize the audio as French based on subtle frequencies from that particular microphone, but fail when presented with French recorded using a different method of sound capture. Some potential solutions included artificially adding random noise to the training set for each language to mitigate distinctive background noise. However, this concern was obviated in other ways. One, a genuine multilingual dataset (MediaSpeech) was used for training, so there was increased assurance of consistent audio quality and recording apparatus across the audio recordings of each language, reducing the possibility that a particular language might have wildly distinctive background audio features. Two, the performance of the model was tested on other datasets which would not share any distinctive audio features with the training data. The success of the model in accurately predicting Arabic and Spanish utterances on external data suggests that the model has learned real features of the different languages, rather than simply audio artifacts.

5.3. Future Directions

One way to improve the model and overall performance of the system could be to drastically reduce the size of the neural network. Since the language identification task is far more trivial than a precise speech-to-text transcription task, the 16-layer TDNN used in the Tedlium recipe is likely unnecessarily large for this purpose. An overly large network could lead to poor training or over-fitting, and most certainly leads to increased training times and required resources. By reducing the number of layers in the network and otherwise simplifying the architecture, the model would be more lightweight and train more quickly.

A more lightweight model could open up additional applications for this language ID system, including in real-time code-switching detection. In an environment where one or multiple people take turns speaking in different languages, a real-time code-switching detection system could pinpoint the exact times when these switches occur, and predict the new language to which the conversation has shifted. Since real-time operation of such a system would require high decoding and scoring speed, a lightweight model would be ideal.

The methodology used in this study could readily transfer over to building a code-switching detection system. The first step in training such a system would be to generate artificial data containing multiple languages in a single utterance, along with timestamps indicating when the code-switches are known to occur. A similar ASR system to the one used in this study could be employed; indeed, only the evaluation stage would need to change significantly. Example input to such an evaluation script is shown in Figure 6. The evaluation script would read through the first three phones and be confident that it is detecting Spanish. Although it sees a single French phone next, it would not decide that the language has switched unless the number of phones of a different language that it reads is above some threshold; therefore, this part of the utterance would still be classified as Spanish. After seeing a few more Spanish phones, the system would detect a block of French phones and determine that the language had switched. The time stamps from the third column of this ctm file would be returned, along with the prediction for the new language - in this case, French.

```
F01_a4_s077_v01 1 0.380 0.020 ES_n_B
F01_a4_s077_v01 1 0.400 0.020 ES_o_I
F01_a4_s077_v01 1 0.420 0.010 ES_s_E
F01_a4_s077_v01 1 0.430 0.030 FR_aa_S
F01_a4_s077_v01 1 0.460 0.010 ES_e_7_S
F01_a4_s077_v01 1 0.470 0.020 ES_l_B
F01_a4_s077_v01 1 0.490 0.040 ES_o_E
F01_a4_s077_v01 1 0.530 0.010 ES_a_7_S
F01_a4_s077_v01 1 0.540 0.010 FR_aa_B
F01_a4_s077_v01 1 0.550 0.040 FR_tt_I
F01_a4_s077_v01 1 0.590 0.030 FR_an_I
F01_a4_s077_v01 1 0.620 0.020 FR_dd_I
F01_a4_s077_v01 1 0.640 0.010 FR_an_E
F01_a4_s077_v01 1 0.650 0.010 ES_n_B
F01_a4_s077_v01 1 0.660 0.010 ES_o_E
F01_a4_s077_v01 1 0.670 0.010 ES_e_B
F01_a4_s077_v01 1 0.680 0.010 ES_n_E
F01_a4_s077_v01 1 0.690 0.030 ES_t_B
```

Figure 6: *Example input to a code-switching evaluation.*

In addition to making the neural network smaller to improve language identification and code-switching detection performance, the architecture of the network could also be dramatically altered by replacing the TDNN model with a simple feedforward network and softmax. This version would implement language identification not by reading prepended symbols, but by direct classification. The disadvantage of this approach is that it would not be as useful for code-switching detection applications as the original approach, since it would process whole utterances at a time and would therefore lack granularity to determine if a language switch has happened in the middle of the utterance. Nevertheless, alternate schemes for language ID should be investigated to see if they are indeed more computationally efficient and accurate.

6. Conclusion

At the outset of this study, a successful language identification system was defined as one that would achieve an error rate of less than 10%. Only the most recent state-of-the-art deep learning methods have managed to achieve this level of accuracy. Although the approach used here did not achieve that benchmark for every language, it did have greater-than-random accuracy in detecting all four languages. This is encouraging, because it suggests that the approach set out here is valid and simply needs to be fine-tuned. In subsequent work the author will indeed fine-tune this method to improve language identification precision to the greatest extent possible. Using a greater volume of speech data in training, perhaps sourced from the multilingual LibriVox or TedX corpora, would likely improve performance [26,27]. Interestingly, it was anticipated that languages with close linguistic relationships might lead to difficulties in distinguishing between several candidate languages. The evidence shown in this paper regarding the mistaken prediction of French phonemes to be Spanish is precisely the expected scenario. More generally, the accuracy of the approach here will always likely be higher for some languages than others due to a variety of factors, including phonetic similarity between languages and variation in the quality of the recordings. Indeed, the mix of languages provided in the closed-set for language identification will also likely make a difference: a closet-set of entirely Romance languages would likely have poor performance compared to a closed-set of distant languages. Various aspects of the approach laid out here will continue to be adjusted in order to eliminate biases and amplify linguistic differences so as to achieve the highest accuracy possible.

7. References

[1] "Automatically detect language: cloud speech-to-text documentation, google cloud." [Online]. Available: https://cloud.google.com/speech-to-text/docs/multiple-languages

[2] Eric-Urban, "Language identification; speech service; azure cognitive services." [Online]. Available: https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/language-identification?tabs=once&pivots=programming-language-csharp

[3] S. Engdahl, "Blogs," 2008. [Online]. Available: https://aws.amazon.com/blogs/aws/amazon-transcribe-now-supports-automatic-language-identification/

[4] M. Zissman, "Automatic language identification using gaussian mixture and hidden markov models," *IEEE International Conference on Acoustics Speech and Signal Processing*, 1993.

[5] Y. Muthusamy, E. Barnard, and R. Cole, "Reviewing automatic language identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, p. 33–41, 1994.

[6] T. J. Hazen and V. W. Zue, "Segment-based automatic language identification," *The Journal of the Acoustical Society of America*, vol. 101, no. 4, p. 2323–2331, 1997.

[7] Y. Muthusamy, R. Cole, and M. Gopalakrishnan, "A segment-based approach to automatic language identification," *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*, 1991.

[8] G. Singh, S. Sharma, V. Kumar, M. Kaur, M. Baz, and M. Masud, "Spoken language identification using deep learning," *Computational Intelligence and Neuroscience*, vol. 2021, p. 1–12, 2021.

[9] P. Heracleous, K. Takai, K. Yasuda, Y. Mohammad, and A. Yoneyama, "Comparative study on spoken language identification based on deep learning," *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018.

[10] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[11] R. Kolobov, O. Okhapkina, A. P. Olga Omelchishina, R. Bedyakin, V. Moshkin, D. Menshikov, and N. Mikhaylovskiy, "Mediaspeech: Multilanguage asr benchmark and dataset," 2021.

[12] A. Guevara-Rukoz, I. Demirsahin, F. He, S.-H. C. Chu, S. Sarin, K. Pipatsrisawat, A. Gutkin, A. Butryna, and O. Kjartansson, "Crowdsourcing Latin American Spanish for Low-Resource Text-to-Speech," pp. 6504–6513, May 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.801

[13] [Online]. Available: https://www.openslr.org/46/

[14] "Automatic speech recognition datasets-magic data," 2022. [Online]. Available: https://www.magicdatatech.com/datasets/mdt-asr-e009-turkish-speech-corpus-1607306180

[15] C. Le Moine and N. Obin, "Att-HACK: An Expressive Speech Database with Social Attitudes," in *Speech Prosody*, Tokyo, Japan, May 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02508362

[16] 2022. [Online]. Available: https://www.openslr.org/57/

[17] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve, "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation," pp. 198–208, 2018. [Online]. Available: https://doi.org/10.1007%2F978-3-319-99579-3_21

[18] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," *Interspeech 2015*, 2015.

[19] H. Beigi, *Fundamentals of Speaker Recognition*. New York: Springer, 2011, ISBN: 978-0-387-77591-3, http://www.fundamentalsofspeakerrecognition.org.

[20] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The kaldi speech recognition toolkit, booktitle = In IEEE 2011 workshop," 2011.

[21] "Cmu sphinx - acoustic and language models - french," 2022. [Online]. Available: https://sourceforge.net/projects/cmusphinx/files/\Acoustic%20and%20Language%20Models/

[22] 2022. [Online]. Available: https://github.com/DuyguA/computational_linguistics

[23] 2022. [Online]. Available: https://github.com/motazsaad/ara-pronunciation-tool

[24] 2022. [Online]. Available: http://www.speech.sri.com/projects/srilm/

[25] 2022. [Online]. Available: https://www.youtube.com/watch?v=uGPWlvOd9Bs

[26] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *ArXiv*, vol. abs/2012.03411, 2020.

[27] E. Salesky, M. Wiesner, J. Bremerman, R. Cattoni, M. Negri, M. Turchi, D. W. Oard, and M. Post, "Multilingual tedx corpus for speech recognition and translation," in *Proceedings of Interspeech*, 2021.