JSPEECH: A MULTI-LINGUAL CONVERSATIONAL SPEECH CORPUS

Ali Janalizadeh Choobbasti[1], Mohammad Erfan Gholamian[2], Saeid Safavi[1]

[1] Miras Technologies International
NO. 92, Movahed Danesh St., Farmanieh, Tehran, Iran
{ali, saeid}@miras-tech.com,
[2] Computer Engineering and Information Technology Department, Amirkabir University of Technology
424 Hafez Avenue, Tehran, Iran
{gholamian}@aut.ac.ir,

ABSTRACT

Speech processing, automatic speech and speaker recognition are the major area of interests in the field of computational linguistics. Research and development of computer and human interaction, forensic technologies and dialogue systems have been the motivating factor behind this interest.
In this paper, **JSpeech** is introduced, a multi-lingual corpus. This corpus contains **1332** hours of conversational speech from **47** different languages. This corpus can be used in a variety of studies, created from **106 public** chat group the effect of language variability on the performance of speaker recognition systems and automatic language detection. To this end, we include speaker verification results obtained for this corpus using a state of the art method based on 3D convolutional neural network.

Index Terms— multilingual speech corpus, speaker recognition, convolutional neural network, machine learning

1. INTRODUCTION

The study and development of multilingual speech systems can be accelerated with the availability of size-able public speech corpora. Applications of multilingual speech algorithms include automatic speaker recognition for biometric authentication, speech understanding systems, automatic conversation systems, pattern recognition, and cognitive science [1, 2]. The speaker recognition problem has been widely investigated in cases where there is a substantial amount of audio data available per speaker. There are not many speech corpora available for less mainstream languages (e.g., Farsi, French, Spanish) in comparison to more dominant languages like English. Investigating and modeling speech characteristics for other languages has, therefore, become difficult.
There are several other multilingual speech corpora available for research purposes. One of the corpora available for multilingual speech analysis is the **OGI multilingual telephone speech corpus** which consists of 1345 telephone conversations (29.5 hours of audio data) recorded in 10 different languages [1], 246 among which are in English, and the remaining nine languages with an average of 122 calls each. The **TDT text and speech corpus** presented in [3] is another multilingual corpus which includes speech audio data from the English, Arabic, and Mandarin languages. The Mandarin-English speech corpus named **SEAME** is presented in [4], which consists of 15 hours of word-level transcribed audio data for each language. Another speech corpus similar to the one presented in this study is presented in [5] in the sense that it is a multi-talker speech corpus.
In this paper, **JSpeech**, which is a multilingual conversational speech corpus is presented. This corpus contains up to **452,007** audio messages scraped from public groups, comprising a total of **1332** hours of converational audio data. The discussions in these groups are unstructured and are conducted with multiple speakers. JSpeech is a multilingual corpus with audio speech data from **47** different languages with over **12140** different speakers. The most notable feature of this audio data is the presence of different uncontrolled environments surrounding the speakers. This is useful in the development of speech technologies that are robust to different kinds of background noise.
In this study, an automatic speaker verification (ASV) system has been designed and developed to validate the corpus. The main engine of this ASV system is a 3D convolutional neural network (CNN) introduced by [6].
In the remainder of this article, first, we look at some basic information about JSpeech in section 2. In section 3, the approach taken towards collecting the data and the generation process of this corpus is explained. Statistics and meta-data about different fields have been presented in section 4. In section 5, the process for validating the corpus data is discussed and the experimental results are presented in section 6. Finally, we conclude our findings in section 7.

2. CORPUS DESCRIPTION

JSpeech is a multilingual speech corpus consisting of about 900 gigabytes of WAV files. Metadata of each file is stored in

Field Name	Description
Voice_id	Unique ID assigned to each voice message
User_id	Unique ID assigned to each speaker
Fwd_from	ID of user that this message has been forwarded from
Reply_to_msg_id	ID of the message this message was replied to
Date	Time stamp of each message
Size	Size of the voice message (byte)
Duration	Duration of the voice message (second)
Chat_name	Group name

Table 1. Corpus Description

an SQLite database.
In order to ensure the diversity and adequacy of the corpus, a set of 106 group chats from different backgrounds and languages were scraped from the public groups of the Telegram[1] messaging application. Each voice message has 6 fields which are described in Table 1.

2.1. Data Privacy and Protection

The groups scraped for creating this corpus are **public** groups that any person can join and view the content. A wrapper for the main Telegram API is used, which Telegram has made available for developers. This data is publicly available to anyone who wants to view it via joining the group. In order to insure speaker anonymity and privacy, the names of the speakers have been replaced using randomly assigned IDs, and no other information regarding the identity of the speaker is stored in the database. This data will not publicly be available, except for a small sample which will be available on GitHub. Researchers with valid requests and research goals can apply for this corpus via e-mail and given the authenticity of their request, this corpus will be securely shared with them after they signed the consent form.

3. CORPUS GENERATION

The overall architecture of the corpus generating system is elaborated in this section. A list of group links is initially fed into the crawler. These group links have been manually classified by the language spoken in the groups. These groups are then scraped for voice messages, and the message ID of the voice messages are given to the scraper. The scraper then fetches the voice messages and writes them into the database. The initial voice messages are in OGG format; therefore the preprocessing module converts the audio files to the WAV format with a sample rate of 48 kHz. This module also extracts additional metadata on the messages and also assigns each user with a user ID instead of their usernames to generate the final corpus. In the remainder of this section, the crawler and scraper module of the system are explained in detail.

[1] https://telegram.org/

3.1. Message Crawler and Scraper

As mentioned earlier, the audio files in this corpus are comprised of thousands of messages crawled from over a hundred public Telegram groups. These groups are mainly language learning groups that people can join and practice speaking in different languages. As for the crawler & scraper module, a well-known wrapper for the Telegram API named Telethone[2] has been used. Given a valid set of public group links, and using the Telegram CLI[3], the crawler starts to join these groups and filter out the voice messages sent in the groups. The scraper then starts fetching the voice messages and writes the messages into the database. The main steps of the crawling and scraping process are as follows:

1. Telegram group links are first read by the crawler. The crawler then joins these groups.
2. Contents of the telegram groups are filtered by the crawler so that only the voice messages are listed.
3. List of voice messages along with relevant metadata are fetched by the scraper and written into the database.

4. CORPUS STATISTICS

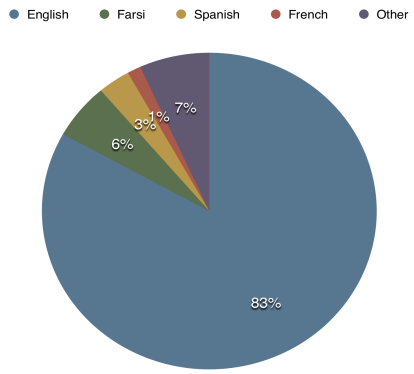

Fig. 1. Pie chart of the amount of data available for each language.

JSpeech is comprised of over **452,007** voice messages. The duration of each voice message varies from a few seconds to a few minutes. The corpus contains a total of 1332 hours of speech which makes it the largest multilingual speech corpus in the world. As shown in Figure 1, the majority of the voice messages are made up of speech spoken in English, but there is also a noticeable amount of audio data available in other languages like Farsi, Spanish, and French. Figure 2 shows the

[2] https://github.com/LonamiWebs/Telethon
[3] https://github.com/vysheng/tg

Language	Number of Speakers	Avg. Message Duration (seconds)	Total Duration (minutes)	Avg. Duration Per Speaker (seconds)	Number of Messages
Arabic	217	11.323631	272.333333	67.5207	1443
Armenian	11	5.034483	2.4333	13.2727	29
Basque	11	10.884058	25.033333	136.5455	138
Bengali	2	24.25	1.6166	48.5	4
Cantonese Chinese	51	8.32122	831.98333	978.8039	5999
Catalan	24	23.977477	88.716666	221.7917	222
Corsican	7	14.550562	21.583333	185	89
Croatian	8	7.48	3.1166666	23.375	25
Czech	7	8.065574	8.2	70.2857	61
Dansk	23	6.660714	31.083333	81.087	280
English	8692	10.694148	66429.01666	363.4304	372703
Esparanto	15	9.388889	5.633333	22.5333	36
Estonian	4	8.5	0.85	12.75	6
Farsi	2105	11.715675	4556.6166	124.7818	23336
Filipino	6	2.923077	0.6333333	6.3333	13
Finish	9	9.766667	4.883333	32.5556	30
French	395	11.66489	1057.616666	122.2678	5440
German	324	8.484562	554.183333	74.5538	3919
Greek	9	4.809524	1.683333	11.2222	21
Haitian Creole	4	9.222222	1.3833333	20.75	9
Hakka Chinese	4	5.833333	1.1666	17.5	12
Hebrew	41	5.147059	17.5	25.6098	204
Hokkien	11	5.24581	31.3	170.7273	358
Hungarian	4	6.777778	1.0166666	15.25	9
Icelandic	6	15.863636	5.816666	58.1667	22
Indian	22	5.478261	10.5	28.6364	115
Indonesian	36	6.424929	75.6	126	706
Irish	12	11.589372	39.983333	199.9167	207
Italian	180	12.691958	697	174.9791	3295
Japanese	172	6.090433	213.26666	57.6396	2101
Korean	71	13.040678	64.11666	48.6962	295
Kurdish	38	26.701987	67.2	106.1053	151
Latin	8	10.375	5.533333	41.5	32
Malay	7	9.8	7.35	63	45
Mandarin Chinese	179	7.0009	547.7	172.9579	4694
Polish	23	4.164384	5.066666	13.2174	73
Portuguese	194	8.128359	675.46666	149	4986
Russian	320	7.451277	418.016666	60.0024	3366
Scandinavian	4	11.62069	5.616666	84.25	29
Shanghainese	4	4.5	0.9	13.5	12
Spanish	479	11.069085	2507.5166	213.1034	13592
Swedish	62	8.046322	98.433333	67.8851	734
Turkish	275	11.024011	543.3	95.8765	2957
Ukrainian	17	6.204545	9.1	32.1176	88
Urdu	17	11.288889	8.46666	29.8824	45
Vietnamese	11	4.830986	5.716666	31.1818	71
Welsh	3	3	0.25	5	5
Total	12140	10.61419	79961.49965	276.094262	452007

Table 2. Statistics for the 47 languages in the generated corpus. The average message duration field is indicative of the average amount of audio available per message. The average duration per speaker field illustrates the average length of audio available per speaker.

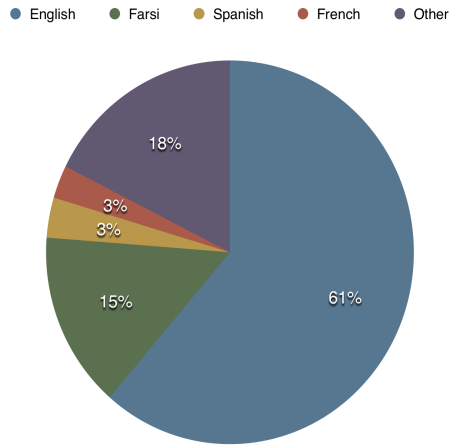

Fig. 2. Pie chart of the number of speakers in each language.

distribution of the number of speakers available for each language. Table 2 summarizes some of the important statistics about JSpeech. It can be observed that the number of total speakers is less than the sum over the number of speakers column for every language. This is due to the fact that some speakers are in multiple groups with different languages spoken in each.

JSpeech consists of scraped audio messages from 12140 dif-

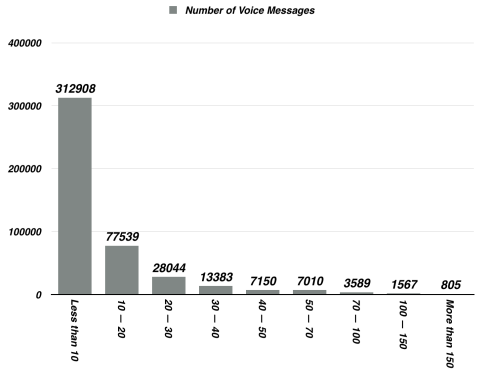

Fig. 3. Histogram of the number of voice messages available with respect to the duration of the message in seconds.

ferent speakers, each with different languages and personal habits of sending voice messages with different duration's, making the duration of the voice messages vary a lot, but the overall average message length is 10.6 seconds long. **1974** of the speakers are in multiple groups with different languages spoken in each, making this corpus applicable to studies on the effect of language mismatch on the performance of speaker verification systems. A histogram which describes the distribution of the audio samples on different lengths is displayed in figure 3. As illustrated in this figure, the majority of the voice messages have a duration of less than 10 seconds.

5. CORPUS APPLICATION

JSpeech is a multilingual speech corpus that can be used for developing text-independent speaker verification and recognition systems. One of the critical characteristics of voice messages is the presence of different environments surrounding the speakers. In addition the audios are recorded using different mobile phones and over a variety of different microphones and channels. This makes the data useful for developing ASV systems that are robust to environmental change. Being comprised of different languages can also help investigate the effects of different languages in the development of ASV systems.

5.1. Speaker Verification System

To verify a speaker's identity is to accept or refuse the speakers claim as one of the known speakers in the system by using the characteristics of their voice captured by a recording device. This crucial problem is one of the branches of biometric authentication and has received tremendous attention over the last few years.

The speaker verification (SV) problem can be divided into text-dependant and text-independent. As suggested by the name, in the text-dependant process, a predefined text phrase is read by all the speakers and is employed in the different stages. Yet in the text-independent approach, the utterances recorded from the speakers are conversational and are without any prior constraints. This makes the problem more challenging compared to the text-dependent approach.

In this study, a corpus for the text-independent scenario has been created where no prior information or constraints are provided for the speakers' utterances for the different stages of speaker verification.

There are generally three steps in the speaker verification process:

1. **Development** of the background model for the speaker representations.
2. Adapting UBM to create speaker models for new users in the **Enrollment** step. The speaker-specific models are created from the utterances provided by the speaker.
3. **Evaluation** of the claimed identity of the test utterances are confirmed or rejected by comparing the obtained score for the utterance with the calculated threshold. During this stage, each test utterance is fed into the network as input, and its representation is extracted. This representation is then compared to all speaker models and the decision is made based on a similarity score.

In this study, the 3D CNN model introduced in [6] has been employed for corpus validation. The application of CNN's

Layer	Input-size	Output-size	Kernel	Stride
Conv 1-1	$\zeta \times 80 \times 40$	$80 \times 36 \times 16$	$3 \times 1 \times 5$	$1 \times 1 \times 1$
Conv 1-2	$80 \times 36 \times 16$	$36 \times 36 \times 16$	$3 \times 9 \times 1$	$1 \times 2 \times 1$
Pool 1	$36 \times 36 \times 16$	$36 \times 18 \times 16$	$1 \times 1 \times 2$	$1 \times 1 \times 2$
Conv 2-1	$36 \times 18 \times 16$	$36 \times 15 \times 32$	$3 \times 1 \times 4$	$1 \times 1 \times 1$
Conv 2-2	$36 \times 15 \times 32$	$15 \times 15 \times 32$	$3 \times 8 \times 1$	$1 \times 2 \times 1$
Pool 2	$15 \times 15 \times 32$	$15 \times 7 \times 32$	$1 \times 1 \times 2$	$1 \times 1 \times 2$
Conv 3-1	$15 \times 7 \times 32$	$15 \times 5 \times 64$	$3 \times 1 \times 3$	$1 \times 1 \times 1$
Conv 3-2	$15 \times 5 \times 64$	$9 \times 5 \times 64$	$3 \times 7 \times 1$	$1 \times 1 \times 1$
Conv 4-1	$9 \times 5 \times 64$	$9 \times 3 \times 128$	$3 \times 1 \times 3$	$1 \times 1 \times 1$
Conv 4-2	$9 \times 3 \times 128$	$3 \times 3 \times 128$	$3 \times 7 \times 1$	$1 \times 1 \times 1$
FC5	$4 \times 3 \times 3 \times 128$	128	-	-

Table 3. The architecture of the proposed 3D-CNN [6].

for speaker recognition has also been recently used in speech processing as demonstrated in [7].

The presented 3D CNN architecture aims to capture the spatial information while trying to preserving the temporal information. Table 3 describes the proposed architecture for the 3D-CNN, and Figure 4 shows the overall block diagram for the speaker verification engine. The kernels' spatial size is presented as $D \times H \times W$ where W and H are the width (frequency) and height (temporal) dimensions, respectively. D represents the kernel dimension alongside the depth, which represents the number of utterances the information has been captured from, in order to perform the particular convolutional procedure.

A challenge faced in the data is the variety of spoken words, a variety of languages and the presence or absence of environmental noise in the background. All these factors can cause a different inference by the softmax layer, even when the same person is speaking, leading to an obstacle when the generalization of the background model is sought. To remedy this issue, the simultaneous capturing of different within-speaker utterances, will be discussed later in section 6.3, proposed by [6].

The pooling operations in the presented architecture are applied exclusively in the frequency domain so that the useful temporal features are kept within the time frames. A stride of 2 is used for the lower level convolutional layers to reduce the capturing of highly overlapped features.

6. EXPERIMENTAL RESULTS

For weight initialization in the development phase, the recently developed variance scaling initializer has been used [8]. For improved generalization and better training convergence, batch normalization [9] has been employed. The softmax layer which has been initialized with the cardinality of N =80, where N represents the number of speakers in the development phase of the ASV engine. For each of the enrollment and evaluation phases, 400 utterances have been randomly selected from the 80 speakers. PReLU activation has been applied in all layers excluding the last one.

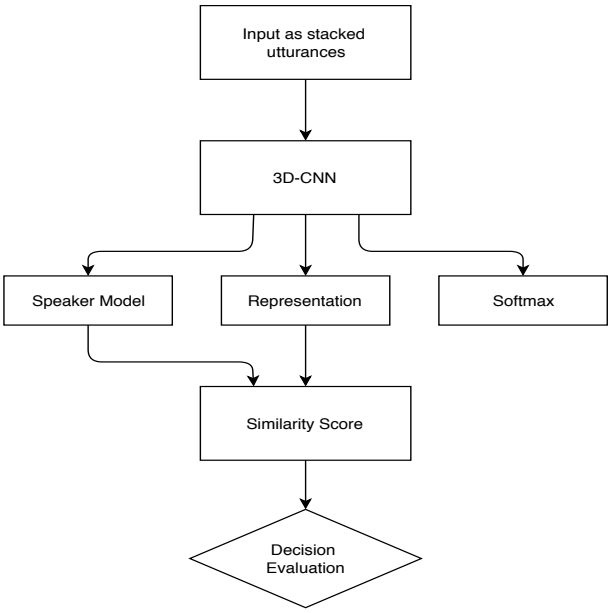

Fig. 4. Block diagram of the 3D-CNN system

6.1. Evaluation and Validation Metric

The experimental results are evaluated using the Detection Error Tradeoff (DET) [10] curve characteristics. The DET curve is made up of the False Negative Rate (FNR) and the False Positive Rate (FPR).

The evaluation metric used in this study is the Equal Error Rate (EER) that represents the point at which the *False Rejection Rate*, and *False Acceptance Rate* become equal. The Area Under the Curve (AUC) score that represents the area under the ROC curve has also been utilized as an indication of model accuracy.

6.2. Dataset

A randomly selected subset of the entire corpus has been used for the experiments. This subset consists of speakers in different environments with different background noise and also different languages. Each speaker has a different number of voice messages of different length. The audio samples available for each speaker were concatenated to create one large audio file for each speaker. Voice Activity Detection (VAD) was applied using the Google WebRTC VAD[4] to the concatenated audio file so that only the parts of speech with human speech remains. The data was split into three distinct parts for each of stages of corpus validation (i.e., development, enrollment, and evaluation). From the development data, 3200 utterances have been extracted for the development stage, 1600 utterances were extracted for the enrollment stage, and 500

[4]https://webrtc.org/

utterances were extracted for the evaluation stage. The utterances for each speaker are extracted using a 20 ms window, where each window has a 10ms overlap with the adjacent window. A feature map consists of ζ utterances, and each feature map was extracted from a separate voice message with different background environmental noise.

6.3. Input Representation

To create a frame level representation of the data, the MFCC[5] features can be used. Due to the last Discrete Cosine Transform (DCT) operation used in creating the MFCC features, a lack of local characteristics is developed, which conflicts with local characteristics of the convolution procedures. The DCT operation disturbs the locality property as in it drops some of the transformed coefficients, throwing away valuable information. To remedy this, the DCT operation is discarded, and the log-energies are used, which are called the Mel-Frequency Energy Coefficients (MFECs) [6].

Overlapping 20ms windows with a stride of 10ms were used

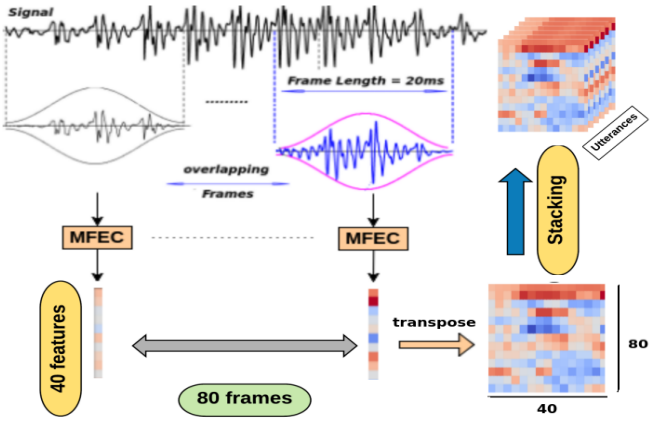

Fig. 5. Input pipeline of the ASV system [6].

for the generation of spectrum features. These spectrum features are used to represent the temporal features. 80 temporal features are used to represent each utterance. Each of these temporal features is formed from 40 MFEC features. Each input feature map has a dimensionality of $\zeta \times 80 \times 40$ which are created from 80 frames and their corresponding spectral features. In this feature map, ζ represents the number of utterances used to model the speaker during the development and enrollment stages. ζ was set to 20 as a default parameter. Figure 5 shows this input pipeline.

Since the CNN has been trained to take ζ as the number of input channels, ζ utterances are needed for each feature map representation. Therefore the test utterances which are singular utterances, are copied ζ times alongside their depth so that suitable input representations are created [6].

[5]Mel-Frequency Cepstral Coefficients

6.4. results

After computing the evaluation measures, an EER score of **38.73%** and an AUC score of **64.16%** were obtained, which indicate the effect of uncontrolled environments and usage of different languages on the performance of ASV systems. Figure 6 shows the DET curve of the employed ASV engine.

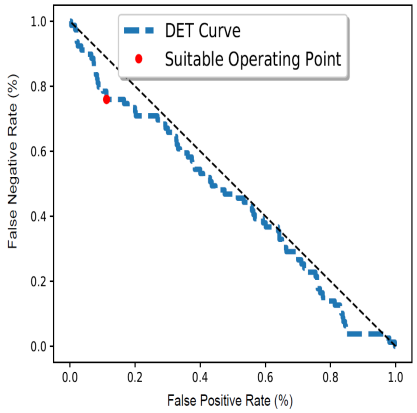

Fig. 6. Detection Error Tradeoff (DET) curve of the applied model

7. CONCLUSION

In this paper, JSpeech, which is an automatically generated multilingual speech corpus, is presented. The system developed in this study uses a list of group links as the seeds for generating the speech corpus. This system scrapes the specified groups and extracts the voice messages sent in the group and the useful meta-data for those voice messages like speaker ID, and the message duration. The generated corpus contains 1332 hours of speech data from 12140 different speakers from different languages. JSpeech is the largest conversational speech corpus available for research and development in a variety of fields like speaker recognition in noisy environments, investigating language mismatch on the performance of speaker recognition systems, and investigation of robustness of biometric engines in real life applications using mobile phones. In order to validate the consistency of JSpeech, a 3D CNN model has been trained on a subset of 80 randomly selected speakers from the entire corpus. The obtained results show how an uncontrolled environment and a mismatch between speaker languages can hurt the performance of ASV systems.

It is expected that with the availability of multilingual speech corpora with different setups of background environments will improve and boost R&D in the field of automatic speaker recognition and voice activity detection. This project is expected to show the performance of state-of-the-art systems on speech in uncontrolled environments and emphasize the need for further research in this field.

8. REFERENCES

[1] Yeshwant K Muthusamy, Ronald A Cole, and Beatrice T Oshika, "The ogi multi-language telephone speech corpus," in *Second International Conference on Spoken Language Processing*, 1992.

[2] Lawrence R Rabiner, "Applications of speech recognition in the area of telecommunications," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 501–510.

[3] Junbo Kong and David Graff, "Tdt4 multilingual broadcast news speech corpus," *Linguistic Data Consortium*, 2005.

[4] Dau-Cheng Lyu, Tien-Ping Tan, Eng Siong Chng, and Haizhou Li, "Seame: a mandarin-english code-switching speech corpus in south-east asia," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[5] Robert S Bolia, W Todd Nelson, Mark A Ericson, and Brian D Simpson, "A speech corpus for multitalker communications research," *The Journal of the Acoustical Society of America*, vol. 107, no. 2, pp. 1065–1066, 2000.

[6] Amirsina Torfi, Jeremy Dawson, and Nasser M Nasrabadi, "Text-independent speaker verification using 3d convolutional neural networks," *arXiv preprint arXiv:1705.09422*, 2017.

[7] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on*. IEEE, 2013, pp. 8614–8618.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[9] Batch Normalization, "Accelerating deep network training by reducing internal covariate shift," 2015.

[10] Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski, and Mark Przybocki, "The det curve in assessment of detection task performance," Tech. Rep., National Inst of Standards and Technology Gaithersburg MD, 1997.